

# **Dynaamisesti kokoaan muuttavat kursorit kohdevalintamenetelmänä**

Joonas Laukkanen

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Tietojenkäsittelyoppi  
Pro gradu -tutkielma  
Ohjaaja: Kari-Jouko Räihä  
Heinäkuu 2008

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojenkäsittelyoppi

Joonna Laukkanen: Dynaamisesti kokoaan muuttavat kursorit kohdevalintamenetelmänä

Pro gradu -tutkielma, 53 sivua, 2 liitesivua

Heinäkuu 2008

---

Näytöllä näkyvien kohteiden valinta perinteistä pistekursoria käyttäen vaatii tarkkuutta ja on hidasta. Kohdevalinnan nopeuttamiseksi onkin pyritty kehittämään pistekursoria nopeampia kohdevalintamenetelmiä, vaikka tällaisia ei vielä käytännön sovelluksista usein löydykään.

Mielenkiintoinen esimerkki pistekursoria nopeammasta kohdevalintamenetelmästä on Tovi Grossmanin ja Ravin Balakrishnanin kehittämä ympyränmuotoinen, dynaamisesti kokoaan lähimpien kohteiden etäisyyksien mukaan muuttava *kuplakursori*. Kuplakursori on nopea, koska alue, jota käyttäen kursorilla kohteita valitaan (*kupla*), laajenee ja kutistuu siten, että kursori ulottuu aina lähinnä olevaan kohteeseen ja lähin kohde on näin aina valittavissa klikkaamalla. Tässä tutkielmassa esitellään kaksi kuplakursorin käyttämien ideoiden ja siinä havaittujen kehityskohtien pohjalta kehitettyä vaihtoehtoista kohdevalintamenetelmää. Keskeinen ajatus oli rajoittaa kuplan kokoa kahden lähinnä olevan kohteen etäisyyksien mukaan niin, että kupla on aina lähellä korkeintaan yhtä kohdetta. Näin kursori on pienempi kuin kuplakursori ja on aina selvää, mikä kohde on kursoria lähinnä oleva kohde. Tämän toivottiin johtavan siihen, että kursori on miellyttävämpi käyttää ja katsoa sekä vähemmän virhealtis. Pienemmän koon saattoi kuitenkin myös odottaa hidastavan kursoria. Tästä syystä toisessa kohdevalintamenetelmässä pyrittiin lisäksi kompensoimaan pientä kokoa säilyttämällä valinta kohteeseen eräänlaisen hännän avulla vielä senkin jälkeen, kun kupla ei enää ulottunut kohteeseen.

Uusille kursoreille suoritettut kokeet kuitenkin osoittivat, että valinnan säilyttämisestä ei juurikaan ollut hyötyä. Kursori, joka ei säilyttänyt valintaa kohteeseen hännän avulla oli kahdesta uudesta kursorista nopeampi ja pidetympi.

Tulokset antoivat myös aiheita kyseenalaistaa aiempi oletus siitä, että kuplakursorin ajoittain suuri koko on ongelmallinen nopeuden kannalta sen vuoksi, että suuri koko saattaa häiritä käyttäjää. Näyttäisi siltä, että kuplakursorin suuren koon ongelmallisuudesta tehdyt havainnot ovat vakuuttavammin selitettävissä sillä, että kuplakursorin kasvaessa suureksi sen perusteella voi olla vaikea arvioida kohde-etäisyyttä, jos kursori kutistuu matkalla kohteeseen.

Avainsanat ja -sanonnat: kuplakursori, dynaaminen kursori, kohdevalinta.

## Sisällys

1. Johdanto.....	1
2. Fittsin laki .....	5
3. Mielenkiintoisia kohdevalintamenetelmiä.....	7
3.1. Kohdeosoittaminen .....	7
3.2. Delphian Desktop.....	8
3.3. Aluekursori .....	9
3.4. Laajenevat kohteet .....	10
3.5. Drag and pop.....	11
4. Kuplakursori.....	12
4.1. Teoreettinen tausta .....	12
4.2. Kuplakursorilla suoritettut kokeet.....	13
4.3. Kuplakursorin ongelmat.....	15
5. Laiska kuplakursori ja tötterökursori .....	17
5.1. Laiska kuplakursori.....	17
5.2. Tötterökursori .....	18
5.3. Kursorien toteutus .....	20
6. Uusilla kursoreilla suoritettut kokeet.....	24
6.1. Kokeiden tavoitteet .....	24
6.2. Koeasetelma .....	24
6.2.1. Koeympäristö .....	25
6.2.2. Koehenkilöt .....	25
6.2.3. Proseduuri .....	26
6.2.4. Riippumattomat muuttujat .....	26
6.2.5. Kokeiden suorittaminen .....	27
6.2.6. Käyttömukavuuden arviointi kyselylomakkeella.....	28
6.3. Tulokset.....	29
6.3.1. Valinta-ajat .....	29
6.3.2. Virheet ja klikkausten sijainti kohteisiin nähden .....	35
6.3.3. Käyttömukavuus .....	41
7. Pohdinta.....	44
7.1. Koeasetelmasta .....	44
7.2. Uusien kursorien optimoinnista .....	45
7.3. Kohde-etäisyyden arvioimisesta .....	46
7.4. Liikkeen suunnan suosimisesta.....	47
8. Yhteenvedo.....	49
Viiteluettelo .....	52
Liite 1: Kyselylomake kursorin arvioimiseksi.....	54
Liite 2: Kyselylomake paremmuusjärjestyksen arvioimiseksi .....	55

**Kiitossanat**

Tahdon kaikella kunnioituksella kiittää Kari-Jouko Rähää ja Poika Isokoskea, jotka kerta toisensa jälkeen jaksoivat kärsivällisesti uhrata aikaa työni ohjaamiseen lukemattomien muiden kiireidensä lomassa. On ollut inspiroivaa työskennellä asiantuntevassa ohjauksessanne. Rohkaisevat kommenttinne pitivät myös huolen siitä, että työhuumori ei päässyt laskemaan missään vaiheessa, vaan tutkielman kirjoittaminen oli varsin antoisa kokemus alusta loppuun.

Erinomaisten ohjaajieni lisäksi tahdon kiittää erityisesti myös vanhempiani, joiden ehdottomaan tukeen ja kannustukseen olen aina voinut luottaa kaikilla elämän aloilla. Kiitos teille kaikille.

## 1. Johdanto

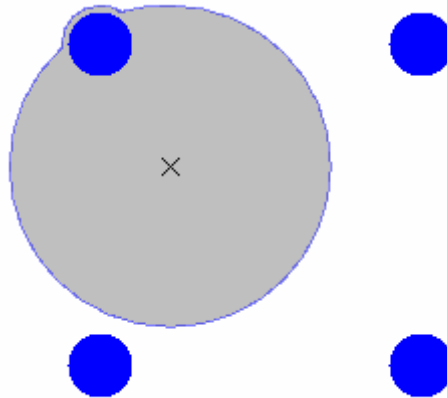
Douglas Engelbartin kehittämä hiiri on ollut näppäimistön ohella keskeinen työkalu tietokoneen sovellusten käytössä käytännössä siitä saakka, kun tietokoneet tulivat yleisesti saataville henkilökohtaisiksi työkaluiksi 1980-luvun puolivälissä. Hiiri on myöskin jonkin verran kehittynyt vuosikymmenten aikana, yhtenä esimerkkinä hiiristä nykyään löytyvä rulla, jonka avulla voidaan selata aktiivista näkymää ylös tai alas. Näytöllä näkyvä kursori (tai vaihtoehtoisesti 'kohdistin') toimii kuitenkin edelleen *pistekursorina* (engl. *point cursor*), jolloin hiirellä kontrolloidaan yhden pikselin kokoista, esimerkiksi nuolena näytöllä visualisoitua kursoria, jolla voidaan klikata näytöllä kursorin kohdalla sijaitsevaa kohdetta. Graafisen käyttöliittymän käyttäjälle näytöllä näkyvien kohteiden, erilaisten painikkeiden ja muiden käyttöliittymäkomponenttien valitseminen perinteistä pistekursoria käyttäen on arkinen ja näennäisen lyhytkestoinen tehtävä. Lähemmin tarkasteltaessa havaitaan kuitenkin, että pistekursori ei ole optimaalinen tapa valita kohteita hiirellä. Kohdistin täytyy siirtää lähtöpisteestä a tarkasti kohteen b päälle, ennen kuin kohdetta voi klikata. Erityisesti kun kohteet ovat pieniä ja kaukana toisistaan, tämä vaatii huomattavaa tarkkuutta ja myös kestää tarpeettoman pitkään. Kohdevalintojen suorittamiseksi on olemassa myös pistekursoria nopeampia kohdevalintamenetelmiä (viitataan tässä tutkimuksessa aina nimenomaan hiirellä suoritettaviin kohdevalintoihin ja hiirikäyttöisiin kohdevalintamenetelmiin, ellei toisin ole mainittu).

Fittsin laki [Fitts, 1954] on informaatioteoriaan perustuva yhtälö, jota voidaan käyttää sen mallintamiseksi, kuinka kauan tietyllä etäisyydellä sijaitsevan tietyn kokoisen kohteen valitsemiseen kuluu aikaa. Yhtälön mukaan valinta-aika kasvaa kun valittavan kohteen etäisyys kasvaa tai kohteen leveys pienenee. Kohdevalinnan nopeuttaminen perustuu tähän nimenomaiseen havaintoon kohteen etäisyyden ja leveyden vaikutuksesta valinta-aikaan. Kohteiden valintaa nopeuttavat kohdevalintamenetelmät tavalla tai toisella joko pienentävät yhtälössä esiintyvää kohteen etäisyyttä tai kasvattavat kohteen kokoa. Fittsin laki esitellään vielä hieman tarkemmin luvussa 2, ennen kuin luvuissa 3 ja 4 esitellään kohdevalinnan nopeuttamiseksi kehitettyjä kohdevalintamenetelmiä.

Kohdevalintojen suorittamista helpottavien ja nopeuttavien menetelmien kehittämistä on perusteltu niin tuottavuuden kasvulla, vanhempien ikäryhmien mahdollisuuksien parantumisella kuin terveydellisilläkin syillä. Worden et al. [1997] havaitsivat vanhempien ihmisten suoriutuvan kohdevalintatehtävistä huomattavasti hitaammin ja jopa yli viisi kertaa suuremmalla virhemäärällä (kun kohteet olivat pieniä) kuin nuorempien koehenkilöiden. He toteuttivat *aluekursorinsa* (kohta 3.3) ensisijaisesti vanhempien ihmisten tarpeista ja rajoituksista huolestuneina helpottamaan ja nopeuttamaan heidän tietokoneenkäyttöään. Hiiren käytön haitalliset terveys-

vaikutuksetkin, vaikkakin suhteellisen vähän tutkittuja verrattuna näppäimistön käyttöön, alkavat olla kiistattomia. Jensen et al. [2002] esimerkiksi osoittivat päivittäisen hiirenkäytön määrän olleen miehillä selvästi yhteydessä käsi- ja rannevaivojen esiintymiseen, kun henkilö käytti hiirtä yli puolet työajastaan. Nopeammat kohdevalintamenetelmät voivat mahdollisesti vaikuttaa osittain näihin hiiren käytön terveyshaittoihin, jos ajatellaan, että menetelmät vähentävät kohteiden valintaan tarvittavien liikkeiden määrää sekä pienentävät tarvittavien liikkeiden suuruutta ja kestoja. Kaikki nämä perustelut ovat yhä pätevämpiä kun otetaan huomioon, että näyttölaitteiden resoluutiot kasvavat jatkuvasti. Kohdevalinnan helpottamiseksi ja nopeuttamiseksi onkin kehitetty sittemmin lukuisia eri menetelmiä. Esittelen niitä luvussa 3.

Luvussa 4 esittelen tämän tutkimuksen tekemiseen inspiroineen, Grossmanin ja Balakrishnanin kehittämän, *kuplakursoriksi* (engl. *bubble cursor*) nimetyn kohdevalintamenetelmän [Grossman & Balakrishnan, 2005a]. Lähtökohtana kupla-kursorin kehittämisessä Grossman ja Balakrishnan käyttivät eräiden aikaisempien kohdevalintaa nopeuttavien menetelmien soveltamaa ideaa suuremmasta *aluekursorista* (engl. *area cursor*), sekä aluekursoreissa havaittuja, erityisesti kursorin koon ja muodon määrittämiseen liittyviä ongelmia. Kuplakursori (kuva 1) toimii siten, että kursori on ympyrän muotoinen alue, jonka koko määräytyy dynaamisesti niin, että kursori ulottuu aina lähimpään kohteeseen, joka on tällöin aktivoitavissa klikkaamalla.




---

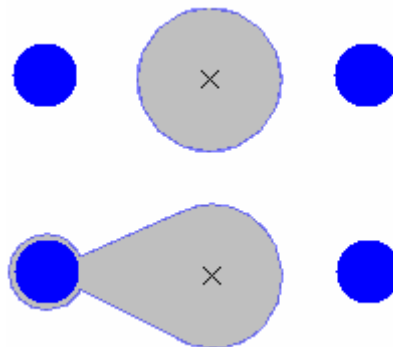
Kuva 1. Kuplakursori.

Kuplakursorin osoitettiin nopeuttaneen kohteiden valintaa huomattavasti, mutta siitä löytyi myös joitakin ongelmia. Keskeisin niistä vaikuttaisi olevan kuplakursorin suuret koonmuutokset. Grossmanin ja Balakrishnanin tutkimuksessa havaittiin esimerkiksi, päinvastoin kuin oltiin odotettu, kohdevalinnan hidastuneen hieman kun kohdetiheys oli pieni, jolloin kursori kasvaa usein varsin suureksi. Lisäksi kuplakursori laajenee ja kutistuu usein varsin aggressiivisesti, kuten Grossman ja Balakrishnan itsekin

tiedostivat. Toisinaan koon muutokset saattavat olla käyttäjälle myös jokseenkin yllättäviä. Ympyrän muotoisena kuplakursori ei liikuttaessa myöskään laajenee ainoastaan liikkeen suuntaan vaan myös liikesuuntaan nähden sivusuunnassa ja kupla osuu usein myös kohteisiin, jotka eivät ole liikkeen suunnassa. Nämä tekijät saattavat häiritä käyttäjää ja toisinaan johtaa myös virheisiin.

Pyrin tässä tutkimuksessa selvittämään, voisiko esitettyjä ongelmia ratkaista kohdevalintamenetelmällä, joka eroaa kuplakursorista siten, että kursori ulottuu tiettyyn kohteeseen vain kun kohde on *selvästi* muita kohteita lähempänä. Tällainen kursori olisi kuplakursoriin nähden kooltaan ja koon muutoksiltaan pienempi ja koon määräytymisperusteen mukaisesti olisi aina selvää, mikä kohde on lähinnä oleva kohde. Kursori ei myöskään osuisi enää niin moniin kohteisiin. Nämä seikat voisivat johtaa suurempaan käyttömukavuuteen ja pienentyneeseen virhealttiuteen.

Koska näin toimiva kursori on pienempi, on selvää, että tällä on myös suora negatiivinen vaikutus kohdevalinnan nopeuteen. Toteutinkin tästä kursorista myös muunnelman, joka pyrkii kompensoimaan kursorin pienemmän koon aiheuttamaa tehokkuuden menetystä säilyttämällä valinnan viimeksi valittuun kohteeseen aina siihen saakka, kunnes kursori osuu seuraavaan kohteeseen. Muunnelmalla oli tätä varten eräänlainen *häntä*, joka yhdisti kuplan viimeksi valittuun kohteeseen, kun kupla ei enää ulottunut kohteeseen. Kutsun ensimmäistä kursoria laajenemisstrategiansa mukaan *laiskaksi kuplakursoriksi*. Muunnelma tästä kursorista näyttää hännän kanssa hieman jäätelötötteröltä, joten se nimettiin *tötterökursoriksi*. Molemmat uudet kursorit näkyvät kuvassa 2.



Kuva 2. Laiskaa laajenemisstrategiaa soveltavat uudet kursorit, laiska kuplakursori ja tötterökursori.

Valinnan säilyttämisen ei kuitenkaan ollut syytä olettaa täysin korvaavan pienemmän kuplan myötä syntyvää tehokkuushaittaa. Oletinkin tötterökursorin olevan kuplakursoria suositumpi vaihtoehto erityisesti sen vuoksi, että se on miellyttävämpi käyttää ja mahdollisesti myös vähemmän virhealtti. Toisaalta, vaikka tötterökursori olisikin oletetulla tavalla miellyttävämpi käyttää, sen käyttöä ei voitane suositella, jos se

häviää liikaa tehokkuudessa. Molempien uusien kursorien tarkempi kuvaus ja ajattelu, jolla suunnittelu perustellaan sekä kursorien toteutus esitellään luvussa 5.

Vaikka en siis odottanut uusien kursorien täysin pärjäävän kuplakursorille tehokkuudessa, uskoin tötterökursorin olevan suositeltavissa käytön miellyttävyyden perusteella, mikäli se osoittautuisi vain vähän hitaammaksi kuin kuplakursori. Kursorien välisen tehokkuuseron suuruutta ja käyttäjien tuntemuksia kursoreista selvitettiin kokeellisesti. Kokeissa koehenkilöiden tehtävänä oli valita toistuvasti eri kursoreilla kaksiulotteisessa ympäristössä tasaväliseen ruudukkoon vaihtelevalla tiheydellä asetelluista kohteista punaisena erottuva kohde. Lisäksi käyttäjiä pyydettiin vastaamaan kursoreiden käytöstä saatuja tuntemuksia tiedusteleviin kysymyksiin. Kokeiden tulokset osoittivat, että tötterökursorista ei kuitenkaan ollut haastajaksi kuplakursorille. Se ei hävinnyt dramaattisesti tehokkuudessa kuplakursorille, mutta koehenkilöt eivät yksinkertaisesti pitäneet kursorista. Valinnan säilyttämisestä ei myöskään saatu toivottua hyötyä.

Laiska kuplakursori toimi kuitenkin hyvin. Sen lisäksi, että laiska kuplakursori miellytti koehenkilöitä, se oli itse asiassa kahdesta uudesta kursorista hieman nopeampi ja se hävisi kuplakursorillekin nopeudessa vain vähän. Tämä havainto johti lopulta tarkastelemaan uudelleen Grossmanin ja Balakrishnanin oletusta siitä, miksi kuplakursorin koko on ongelmallinen valinta-aikojen kannalta. Sen sijaan, että suuri koko häiritsisi käyttäjiä, kuten tutkijat arvelivat, ilmiö selitettäneen vakuuttavammin kohde-etäisyyden arvioimisen vaikeudella, kun kursorin koko muuttuu dynaamisesti. Suoritetut kokeet, niiden suunnittelu, toteutus sekä tulokset esitellään yksityiskohtaisesti luvussa 6.

Luvussa 7 pohdin koeasetelman vaikutusta tuloksiin sekä muita tekijöitä, joilla on saattanut olla vaikutusta uusien kursorien toimivuudesta saatuihin tuloksiin. Lisäksi pohdin kohde-etäisyyden arvioimisen merkitystä kohdevalinnan nopeuden suhteen ja sitä, kuinka kuplakursoria voisi yrittää nopeuttaa suosimalla liikesuuntaa.

Luvussa 8 esitän yhteenvedon tutkimuksesta.



## 2. Fittsin laki

Fittsin laki [Fitts, 1954] on ihmisen psykomotorisen toiminnan malli [MacKenzie, 1992, p. 92]. Alun perin kokeellisen psykologian tutkimusalan tutkimustuloksena julkaistu laki ennustaa, kuinka kauan tietyllä etäisyydellä sijaitsevan tietyn levyisen kohteen valitsemiseen kuluu aikaa. Fittsin hypoteesin mukaan ihmisen suorittaessa muuttumattomassa ympäristössä nopeita, samanlaisia, opittuja liikkeitä, suorituskykyä rajoittaa pääasiallisesti vain henkilön motoriikka. Suorituskyky säilyy verrattain vakiona riippumatta tehtävän vaikeudesta, kohteen etäisyydestä suhteessa valinnan vaatimaan tarkkuuteen (kohteen leveys) [Fitts, 1954]. Näin ollen kohteen valitsemiseen kuluva aikaa on mahdollista mallintaa jokseenkin suoraviivaisesti kun tiedetään tehtävän vaikeus ja suorituskyky. Fittsin laki kohteen valitsemiseen kuluvan ajan  $MT$  mallintamiseksi esitetään tyypillisesti seuraavassa muodossa.

$$MT = a + b \log_2 \left( \frac{A}{W} + 1 \right) \quad (1)$$

Tässä  $A$  on etäisyys kohteeseen,  $W$  kohteen leveys, logaritminen termi  $\log_2(A/W+1)$  tehtävän vaikeus (ID, index of difficulty) esitettynä biteissä, sekä  $a$  ja  $b$  empiirisesti tehtävän mukaan määriteltäviä vakioita. Ensin mainitun vakion voi tulkita mallintavan sitä, kuinka kauan valintaan vähintään kuluu aikaa, valinnan vaikeudesta riippumatta. Jälkimmäinen vakio  $b$  kuvaa valinta-ajan muutosta valinnan vaikeuden kasvaessa, sekunteja per bitti. Yhtälö eroaa tässä hieman Fittsin alun perin esittämästä muodosta. Tehtävän vaikeus  $ID$  esitetään tyypillisesti Shannonin teoreeman 17 [Shannon & Weaver, 1949] mukaisessa, jo edellä esitetyssä muodossa  $\log_2(A/W+1)$ . Erityisesti on havaittu, että  $ID$  ei tässä muodossa voi saada negatiivista arvoa, kuten Fittsin alun perin esittämässä, hieman edellisestä poikkeavassa muodossa [MacKenzie, 1989].

Itse laki on vuosikymmenien aikana sen esittämisen jälkeen “osoittautunut yhdeksi kestävimmistä, eniten lainatuista ja laajimmin sovelletuiksi kokeellisesta psykologiasta syntyneeksi malliksi”, kuten MacKenzie lakiin kunnioittavasti viittaa [1992, p. 93]. Lakia onkin sovellettu monilla tutkimusaloilla ja sen toimivuutta tukevia kokeita on suoritettu lukuisin, hyvinkin erilaisin asetelmin muun muassa veden alla, eri lihasryhmillä, vanhoilla ihmisillä, mieleltään sairailta sekä apinoilla [MacKenzie, 1992]. Lisäksi Fittsin laki on osoittautunut käyttökelpoiseksi toistuvasti myös vuorovaikutteisen teknologian alalla erilaisten interaktiivisten järjestelmien tehokkuuden mallintamisessa.

Kohdevalinnan nopeuttamisen kannalta merkittävää on ollut ennen kaikkea havaita, että lain perusteella kohdevalintaa on mahdollista nopeuttaa joko pienentämällä

välimatkaa kohteeseen tai kasvattamalla kohteen kokoa. Päättelyä voidaan käyttää hyväksi suoraan käyttöliittymäsuunnittelussa. Esimerkiksi hiiren toisella painikkeella avautuva valikko (engl. *context menu*) aukeaa välittömästi kursorin vierestä ja sen alkiot ovat näin nopeammin valittavissa kuin esimerkiksi ikkunan yläreunassa sijaitsevaan valikkoriviin sijoitetun valikon alkiot. Toinen esimerkki on piirakkavalikko (engl. *pie menu*), jossa valikon alkiot ovat lisäksi piirakan muotoon aseteltuina kaikki yhtä nopeasti valittavissa. Yleisemmin päättelyä voisi soveltaa käyttöliittymäsuunnittelussa yksinkertaisesti kasvattamalla kohteiden kokoa ja sijoittamalla kohteet lähekkäin. Tämä lähestymistapa on kuitenkin ymmärrettävästi ongelmallinen lukuisista syistä jo käyttöliittymän käytettävyyden kannalta. Kohteiden kasvattaminen kasvattaa lopulta lisäksi myös kohteiden välimatkaa, mikä vaikeuttaa edelleen komponenttien optimaalista sijoittelua. Kohdevalinnan nopeuttamiseksi löytyy myös huomattavasti elegantimpia tapoja. Joitakin tällaisia kohdevalintamenetelmiä esitellään luvussa 3.

Yhtälöstä on olennaista huomata myös, että W:n tulkinta kohteen leveytenä johtaa siihen, että laki on luonteeltaan lähtökohtaisesti yksiulotteinen. Sopivin muunnoksin yhtälöä on kuitenkin mahdollista soveltaa myös kaksiulotteisiin tehtäviin, joita esimerkiksi tietokoneen näytöllä tapahtuvat kohdevalinnat tyypillisesti ovat. Tällöin on tarpeen ennen kaikkea käyttää mielekästä tulkintaa kohteen leveydestä (esimerkiksi kohteen leveys lähestymissuunnasta nähden) ja ottaa huomioon eri suuntiin tehtävien liikkeiden suorituksessa esiintyvät motoriset erot (esimerkiksi se, että eri suuntiin suoritettavat liikkeet suoritetaan hieman eri nopeuksin). Fittsin lakia ei kuitenkaan käsitellä tämän tutkimuksen puitteissa tätä syvällisemmin. Fittsin lain laajentamista kaksiulotteisiin tehtäviin käsittelevät seikkaperäisesti esimerkiksi MacKenzie ja Buxton [1992], Accot ja Zhai [2003] sekä Grossman ja Balakrishnan [2005b]. Yleisemmin, joskin edelleen erityisesti vuorovaikutteisen teknologian näkökulmasta Fittsin lakia käsittelee MacKenzie [1992].

### 3. Mielenkiintoisia kohdevalintamenetelmiä

Perinteinen pistekursori on Fittsin lakiin suhteutettaessa varsin yksinkertainen ja hidas tapa suorittaa kohteen valinta. Pistekursoria käyttäen kohdistin on aina siirrettävä tarkasti (mahdollisesti pienen, kaukana sijaitsevan) kohteen päälle, jotta se voidaan onnistuneesti valita. Näin on toimittava, vaikka käyttäjä haluaisi valita ainoan kohteen ruudun tietyssä reunassa, lähtöpisteen ja kohteen välissä on vain tyhjää tilaa ja kohdistin lähestyy juuri tuota yksinäistä kohdetta, ja on käytännössä selvää, minkä kohteen käyttäjä aikoo valita. Pistekursori ei selvästikään hyödynnä tehokkaasti informaatiota kohteiden sijainnista tai hiiren liikkeistä. Kun tätä informaatiota käytetään tehokkaammin, käyttöliittymän komponentteja tai kursoria on mahdollista mukauttaa dynaamisesti niin, että kohteen valinnan ei tarvitse välttämättä edellyttää täsmällistä liikettä lähtöpisteestä tarkasti alkuperäiselle kohdealueelle. Näin onkin kyetty kehittämään useita pistekursoria nopeampia kohdevalintamenetelmiä.

Seuraavissa kohdissa esitellään esimerkkejä vuorovaikutustekniikoista, joilla on yritetty nopeuttaa kohdevalintaa. Tekniikat pyrkivät nopeuttamaan kohdevalintaa joko pienentämällä A:ta (kohteen etäisyyttä), kasvattamalla W:tä (kohteen leveyttä / kokoa), tai sekä pienentämällä A:ta että kasvattamalla W:tä. Tämän tutkielman inspiraationa toiminut kuplakursori esitellään yksityiskohtaisemmin erikseen luvussa 4.

#### 3.1. Kohdeosoittaminen

*Kohdeosoittaminen* (engl. *object pointing*) [Guiard et al., 2004] on hyvä esimerkki kohdevalintamenetelmästä, joka käyttää hyväkseen kohdistimen tuottamaa informaatiota huomattavasti tehokkaammin kuin perinteinen pistekursori. Nimensä mukaisesti kohdeosoittamisessa pyritään poimimaan hiiren liikkeistä tieto siitä, mitä kohdetta hiiren liikkeellä yritetään osoittaa, ei siitä, mitä pikseliä osoitetaan. Tavoitteena on päästä eroon tyhjän tilan osoittamisesta, mihin suuri osa ajasta kohdevalintoja pistekursorilla suoritettaessa usein kuluu (esimerkiksi tyhjä tausta työpöydällä lähtöpisteen ja kohteen välissä). Pistekursorilla tapahtuva osoittaminen, *bittikartta-osoittaminen*, hyödyntää vain informaatiota kohdistimen sijainnista bittikartalla. Tämä ei selvästikään ole optimaalinen lähestymistapa, kun järjestelmän kannalta on usein tarpeellista vain kyetä spesifioimaan jokin tietty kohde. Guiard kollegoineen havainnollistaa tätä esimerkillä muuten tyhjästä graafisesta 1600x1200 resoluutioisesta työpöydästä, jolla on kuitenkin 40 ikonia, kukin 20x30 pikseliä kooltaan. Biteissä ilmaistuna informaatio, joka on välitettävä järjestelmälle yhden 600 pikselin laajuisen kohteen identifioimiseksi pistekursorilla on 11.6, vaikka yhden kohteen ilmaiseminen 40 kohteen joukosta vaatii vain 5.3 bittiä [Guiard et al., 2004].

Kohdeosoittaminen toimii niin, että hiiren liikkeet siirtävät kursorin suoraan seuraavaan kohteeseen liikesuunnassa yli tyhjän alueen. Ensin, kursorin siirtyessä

kohteen ulkopuolelle, tarkistetaan, että kursori liikkuu tarpeeksi nopeasti ollakseen selvästi uuden kohteen valitsemiseksi suoritettu liike. Sen jälkeen tarkistetaan, ettei kursorin vauhti laske samalla hetkellä liian jyrkästi. Tällä pyritään varmistamaan, ettei kohdealueen ylitys johdu vain kohteen 'yli-ampumisesta' kun kursori on jo siirretty seuraavaan kohteeseen. Jos ehdot eivät täyty, kursori jää alkuperäiseen kohteeseen. Jos ehdot täyttyvät, pyritään etsimään lähin kohde liikesuunnasta ja siirretään kursori tuon kohteen päälle, sille sivulle kohdetta, joka sijaitsee lähinnä lähtöpistettä. Jos liikesuunnassa, tietyn kokoisella sektorilla, ei ole ainuttakaan kohdetta, kursori jää alkuperäiseen kohteeseen.

Kohdeosoittamisessa jätetään siis huomioimatta kaikki tyhjä alue. Kursori on koko ajan jonkin kohteen päällä ja kohdistimen liikkeet johtavat joko toiseen kohteeseen siirtymiseen tai samassa kohteessa pysymiseen. Fittsin lakiin suhteuttaen kohdeosoittaminen pienentää A:ta, etäisyyttä kohteeseen. Kohdeosoittamisen raportoitiinkin nopeuttavan kohdevalintaa kaksiulotteisessa ympäristössä perinteiseen pistekursoriin nähden. Näin tapahtui kuitenkin vain kun valinnan vaikeus ylitti tietyn rajan, muulloin kohdeosoittaminen hidasti valintaa. Mitä vaikeampi valinta oli, sitä suurempi myös hyöty [Guiard et al., 2004]. Grossman ja Balakrishnan havaitsivat kohdeosoittamisen kuitenkin kärsineen kovasti suuresta kohdетиheydestä [Grossman & Balakrishnan, 2005a]. Kohdeosoittamisen ongelma on siinä, että liikesuunnasta valitaan aina lähin kohde ja oikeaan kohteeseen täytyy toisinaan liikkua usean eri kohteen kautta.

Informaatiota kohdistimen liikkeistä voisi ehkä olla mahdollista käyttää hyväksi vieläkin enemmän. Kohdeosoittaminen hyödyntää vain tietoa kursorin suunnasta ja valitsee aina lähimmän kohteen liikesuunnasta.

### **3.2. Delphian Desktop**

Asano et al. [2005] kehittivät kohdeosoittamista muistuttavan kohdevalintamenetelmän, joka niin ikään pyrki vähentämään tyhjän tilan yli siirtymiseen kuluvaa aikaa.

Aikaisempien tutkimustulosten perusteella [Asano et al., 2005, p. 134] oli mahdollista olettaa, että kohdistimella suoritetun liikkeen huippunopeus esiintyy liikkeen *suunnitteluvaiheessa*, liikkeen alussa, ja että huippunopeus korreloi kohteen etäisyyden kanssa niin, että nopeus kasvaa etäisyyden kasvaessa ja niin tapahtuu kohteen koosta riippumatta. He vahvistivat vielä alustavasti, että huippunopeus (a) kasvaa jokseenkin tasaisesti kohteen etäisyyden kasvaessa ja (b) on riippuvainen jokseenkin symmetrisesti liikkeen suunnasta, kasvaen liikesuunnan lähestyessä horisontaalista tasoa. Näiden havaintojen perusteella he pyrkivät toteuttamaan kohdevalintamenetelmän, joka kohdistimella suoritetun liikkeen saavutettua huippunopeutensa siirtää kohdistimen suoritetun liikkeen suunnan ja huippunopeuden perusteella suoraan sinne minne kursori

olisi muutenkin siirretty. Siirto ennustettuun päätepiisteeseen visualisoitiin nopealla viiva-animaatiolla.

Kullekin käyttäjälle kalibrointia vaativaa kohdevalintamenetelmää testattiin yksinkertaisella koeasetelmalla, jossa käyttäjien oli toistuvasti valittava samanlainen, vaihtelevalla etäisyydellä, vaihtuvassa suunnassa (8 pääilmansuuntaa) sijaitseva kohde. Jos kohdistin päätyi kohteen ulkopuolelle oli suoritettava myös tarvittavat korjaavat liikkeet, kunnes kohde valittiin onnistuneesti.

Delphian Desktop osoittautui pistekursoria nopeammaksi, kun valittava kohde oli riittävän kaukana. Etäisyyden arvioiminen huippunopeuden perusteella ei kuitenkaan tuonut hyötyä ennen kuin välimatkaa kohteeseen oli vähintään 800 pikseliä. Siitä huolimatta Delphian Desktop on mielenkiintoinen esimerkki siitä, kuinka suurempaa määrää informaatiota hiiren liikkeistä voi olla mahdollista käyttää hyväksi kohdevalinnan nopeuttamiseksi. Koska tarkkoja siirtoja on mahdoton ennustaa pelkästään alkuperäisen nopean liikkeen perusteella, voisi myös spekuloida toimivuuden paranevan jos ennustukset voisi yhdistää jonkin toisen menetelmän, esimerkiksi kohdeosoittamisen kanssa. Asano et al. [2005, p. 140] näkivät, ettei Delphian Desktop olisi ristiriidassa ainakaan laajenevien kohteiden (kohta 3.4) tai hiiren ja näytöllä näkyvän kursorin liikkeiden suhdetta (engl. *control-display ratio, gain*) kohteiden päällä säätelevän kohdevalintamenetelmän [Blanch et al., 2004] kanssa.

Delphian Desktopia ei testattu ympäristössä, jossa olisi ollut häiritseviä kohteita lähtöpisteen ja kohteen välillä, joten sen toimivuutta todellisessa ympäristössä tai soveltuvuutta yhdistettynä johonkin toiseen kohdevalintamenetelmään on vaikea arvioida. Asano et al. raportoivat kuitenkin etäisyyden kohteeseen siirron jälkeen olleen suuremmilla kohde-etäisyyksillä keskimäärin 17.9% alkuperäisestä etäisyydestä kohteeseen, siirron jäätyä yleensä vajaaksi kohteesta. Näin suuren virhemarginaalin voitaneen olettaa vaikeuttavan huomattavasti menetelmän soveltamista esimerkiksi kohdeosoittamisen kanssa varsinkin kun kohdettiheys kasvaa.

### 3.3. Aluekursori

Kabbash ja Buxton [1995] pohtivat, että aivan kuten karpäseenkin on helpompi osua karpäslätkällä, pitäisi myös pieneen kohteeseen olla helpompi osua kursorilla, joka on suuremman kokoinen alue kuin vain yksi pikseli. He olettivat, että suuremman kokoisen kursorin toimintaa voisi itse asiassa mallintaa suoraan Fittsin lailla jos tulkitaan kursorin leveys yhtälössä  $W$ :nä. He myös demonstroivat tätä menestyksekkäästi ylisuuria tennismailoja valmistaneen Prince-mailamerkin mukaan nimetyllä Princemenetelmällään. Tällä aluekursorilla valinta oli suoritettavissa tavallisen pistekursorin yhden pikselin sijaan suuremmalla suorakulmion muotoisella alueella. Kabbash ja Buxton uskoivat aluekursorinsa helpottavan erityisesti pienten, kenties vain yhden

pikselin kokoisten kohteiden valintaa, joiden valitseminen oli hankalaa tavallisella kursorilla.

Kursori toimikin käytännössä hyvin. Sen havaittiin nopeuttavan pienen kohteen valintaa lähes yhtä paljon kuin valintaa olisi ollut mahdollista nopeuttaa kasvattamalla kohteiden kokoa vastaavasti. Käytännön ongelma, joka aluekursorissa kuitenkin havaittiin, oli mahdolliset konfliktit kun kursorin alueella on useampi kuin yksi kohde. Ongelma voitaisiin kuitenkin ratkaista esimerkiksi sijoittamalla aluekursorin sisälle erillinen kohdistin, kun aluekursorin alueella olisi useampia kohteita tai jos hiirellä suoritettava tehtävä muuten vaatisi suurempaa tarkkuutta.

Konseptia erillisestä konfliktit ratkaisevasta kohdistimesta aluekursorin sisällä sovelsivat sittemmin motorisilta kyvyiltään heikompien, vanhempien ihmisten tietokoneiden käytön helpottamiseen pyrkineet Worden ja muut [1997]. He toteuttivat (muun muassa) kohdistamista helpottavan neliön muotoisen aluekursorin, jonka sisällä oli perinteisen pistekursorin tapaan toimiva *ristikursori* (engl. *crosshair*), jolla valinta suoritettiin kun neliön muotoisen alueen sisällä oli useampia kohteita. Muulloin kohde oli valittavissa millä tahansa osalla kursorin aluetta. Worden et al. olettivat, että kursori ei hidastaisi tiheästi sijoiteltujen kohteiden valintaa pistekursoriin nähden ja nopeuttaisi valintaa kun kohteet olivat kauempana toisistaan. Kursori toimikin juuri näin. Erityisen suuresti kursorista hyötyivät juuri huolenaiheena olleet vanhemmat tietokoneen käyttäjät. Sittemmin aluekursori on toiminut ainakin suurimpana innoittajana Grossmannin ja Balakrishnanin kuplakursorille [Grossman & Balakrishnan, 2005a].

Aluekursorin ongelmana voidaan pitää sitä, että sille on vaikeaa määrittää optimaalinen koko. Kursorin kasvava koko tuo tehokkuutta erityisesti kun kohteet ovat huomattavasti kursoria pienempiä ja vain siihen saakka, kun kohdetiheys ei kasva niin suureksi, että valinnan joutuu suorittamaan alueen sisällä olevalla kursorilla. Aluekursoriinkin liittyviin ongelmiin palataan vielä lyhyesti kuplakursorille omistetun luvun (luku 4) yhteydessä.

### **3.4. Laajenevat kohteet**

Dynaamisesti laajenevat kohteet ovat harvinainen esimerkki kohdevalintaa nopeuttavasta kohdevalintamenetelmästä, joka on levinnyt lukuisiin olemassa oleviin käytännön sovelluksiin (esimerkiksi Mac OS X-työpöydän työkalupalkki). Idea on varsin intuitiivinen. Hyödyntäen informaatiota kohdistimen etäisyydestä kohteisiin laajennetaan kohdistimen lähellä olevia kohteita, jolloin niistä tulee helpommin valittavia. Voidaanko kohteen merkitsevänä leveytenä kuitenkin pitää sen lopullista kokoa, kun kohde laajenee vasta siinä vaiheessa kun kohdistin on kohteen lähellä? Ja toisaalta, missä vaiheessa kohteen on laajennuttava, jotta suuremmasta koosta on hyötyä? McGuffin ja Balakrishnan [2002] etsivät vastauksia näihin kysymyksiin selvittääkseen laajenevien kohteiden vaikutusta kohdevalinnan nopeuteen ja sitä, kuinka

kohdevalintamenetelmä pitäisi suhteuttaa Fittsin lakiin. He testasivat laajenevia kohteita vaihtelevin laajenemisparametrein ja löysivätkin mielenkiintoisia, sekä laajenevien kohteiden soveltamiseen kannustavia että niiden soveltamista ohjaavia tuloksia.

McGuffin ja Balakrishnan havaitsivat, että laajenevan kohteen valitsemiseen kuluvaa aikaa voidaan mallintaa Fittsin lailla, että näin voidaan toimia käyttäen  $W$ :nä kohteen laajennettua kokoa, ja että näin voidaan toimia, vaikka laajeneminen alkaisi vasta kun kohdistinta on siirretty jo 90% alkuperäisestä etäisyydestä kohteeseen. Laajenevat kohteet eivät siis kärsi tehokkuudessa siihen nähden, että kohteet olisivat koko ajan laajennettuina.

### **3.5. Drag and pop**

Drag and pop on vuorovaikutustekniikka, jossa jotakin käyttöliittymän ikonia raahattaessa liikkeen suunnassa sijaitsevat potentiaaliset kohdeikonit tuodaan tilapäisesti lähemmäs kohdistinta. Baudischin ja muiden [2003] kosketus- ja kynänyttöille kehittämässä vuorovaikutustekniikassa järjestelmä reagoi tiettytyyppisen ikonin raahaamiseen siirtämällä eräänlaisten *kuminauhujen* avulla raahattavan ikonin kanssa yhteensopivat todennäköiset kohdeikonit tilapäisesti lähemmäs raahattavaa ikonia. Kohteen valinta on tämän jälkeen suoritettavissa lyhyemmällä kohdistimen siirrolla.

## 4. Kuplakursori

Innoitus tähän työhön löytyi Grossmanin ja Balakrishnanin kehittämästä kuplakursorista [Grossman & Balakrishnan, 2005a], joka on kiistattomasti eräs toistaiseksi onnistuneimmista kehityksistä kohdevalinnan nopeuttamiseksi. Tässä luvussa esitellään ensin kohdevalintamenetelmän teoreettista taustaa (kohta 4.1), minkä jälkeen kuvaillaan itse kursori (4.2) sekä sille suoritettut kokeet (4.3) ja lopulta kohdassa 4.4 siinä havaitut ongelmat.

### 4.1. Teoreettinen tausta

Kuplakursorin suunnittelussa Grossman ja Balakrishnan paneutuivat erityisesti kahden aluekursori-aiheen ongelman ratkaisemiseen. [Grossman & Balakrishnan, 2005a, p. 283]

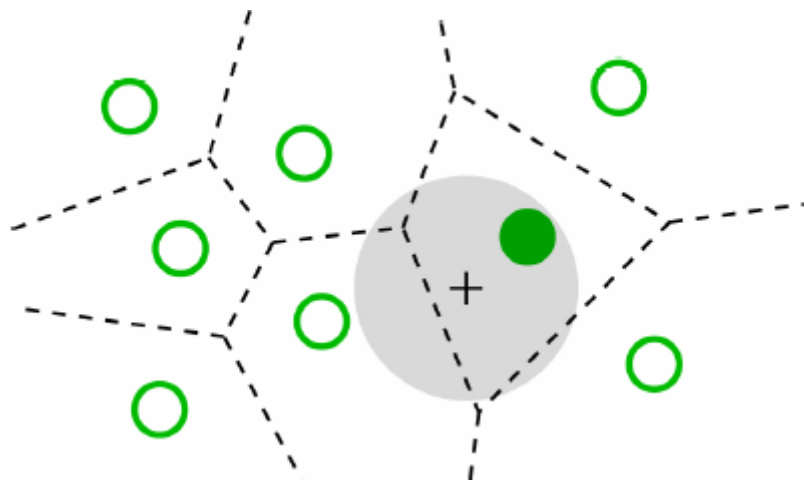
1. Koska aluekursorit olivat tyypillisesti neliöitä tai suorakulmioita, aluekursorille oli mahdollista, että kohde, joka kursorilla valittiin, ei ollut kursoria lähinnä ollut kohde (kun kohteen valitsi kursorin jokin nurkka) ja kohteen valitseminen saattoi edellyttää ei-optimaalista hiiren liikerataa.
2. Aluekursorin valinta-alueen sisällä saattoi sijaita useampi kohde samaan aikaan (katso kohta 3.3). Vaikka ongelma olikin ratkaistavissa erillisellä, toisella aktivointipisteellä varsinaisen aluekursorin sisällä, aluekursorista ei tällöin enää saatu tehokkuushyötyä kun kursori toimi perinteisen pistekursorin tapaan. Aluekursorin tehokkuus oli siis riippuvainen kursorin koosta suhteessa kohteiden sijoitteluun näytöllä.

Ongelmien ratkaisemiseksi tutkijat esittivät kuplakursoriksi nimeämänsä vaihtoehdon edellisille aluekursorille. Ensinnäkin, kuplakursori on ympyrän muotoinen, mikä ratkaisee ensimmäisen ongelman, kun ympyrän muoto takaa, että lähin kohde tulee aina valituksi ensimmäisenä. Toisekseen, kursorin koko muuttuu dynaamisesti niin, että kursori ulottuu aina lähimpään kohteeseen tai (korkeintaan) sulkee sen sisälleen. Lähinnä oleva kohde on näin aina valittavissa klikkaamalla ilman, että kuplan keskellä sijaitseva kohdistin olisi siirrettävä kohteen päälle. Kursorin koon määrittäminen niin, että kupla ulottuu aina lähimpään kohteeseen ja vain lähimpään kohteeseen poistaa ongelman mahdollisista useammista kohteista kursorin valinta-alueen sisällä.

Kursorin dynaaminen koon määrittäminen on myös syy siihen, että kuplakursori on niin nopea. Dynaaminen koon määrittäminen johtaa kohteiden valinta-alueiden kasvamiseen niin, että kohteiden välillä ei itse asiassa ole lainkaan tyhjää tilaa, vaan kaikki tila on käytetty



kohteiden valinta-alueisiin. Kuvan 3 Voronoin diagrammi havainnollistaa tilan jakamista kohteiden valinta-alueiksi [Grossman & Balakrishnan, 2005a, p. 284].



Kuva 3. Näytön alueen jakaminen kohteiden valinta-alueiksi.

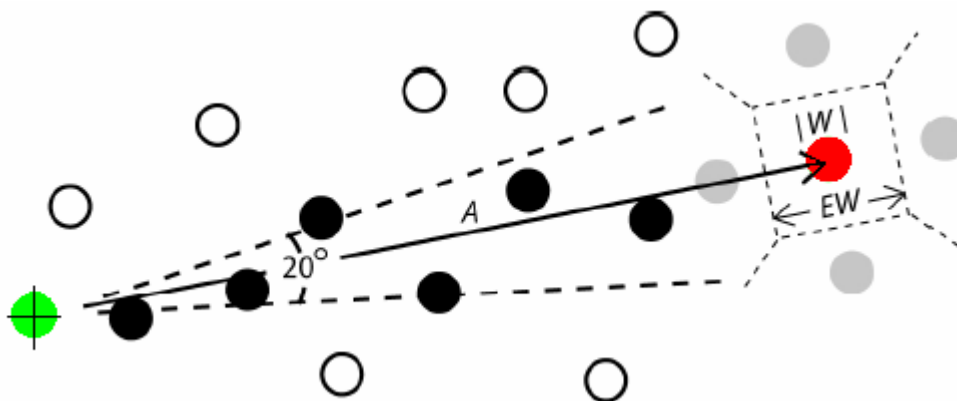
#### 4.2. Kuplakursorilla suoritettut kokeet

Laajenevien kohteiden parissa saatujen tutkimustulosten perusteella (katso kohta 3.4) oli mahdollista olettaa, että kuplakursorin nopeutta voitaisiin mallintaa soveltaen Fittsin lakia niin, että kohteen koko valinta-alue tulkitaan kohteen merkitsevänä leveytenä. Ottaen huomioon kuplan jatkuva kasvaminen ja pientyminen tämä ei ollut kuitenkaan itsestäänselvä oletus. Tämän oletuksen pätevyyyden lisäksi Grossman ja Balakrishnan [2005a] halusivat selvittää kokeellisesti kuinka kursori toimii kohdeosoittamiseen verrattuna.

Kuplakursorin toimivuutta testattiin kahdessa kokeessa yksi- ja kaksiulotteisissa ympäristöissä vaihtelevan kokoisilla kohteilla ja tiheyksillä. Ensimmäisessä, yksiulotteisessa ympäristössä suoritettussa kokeessa Grossman ja Balakrishnan pyrkivät ensisijaisesti verifioimaan, että Fittsin laki todellakin soveltuu kursorin nopeuden mallintamiseen. Tavoitteena oli myös selvittää, kuinka pitkälle tulkinta kohteen tosiasiallisesta valinta-alueesta kohteen merkitsevänä leveytenä on oikeutettu. Yksinkertaisessa kokeessa koehenkilöt klikkasivat vuorotellen ruudun vasemmalla ja oikealla puolella yhtä kaukana keskustasta sijainneita kahta kohdetta, joiden valinta-alueiden leveyksiä kontrolloitiin ylimääräisten ruudulle sopiviin kohtiin sijoitettujen kohteiden avulla. Kuplakursorin nopeus osoittautui kokeessa mallinnettavaksi Fittsin lailla ja kohteen todellinen valinta-alue osoittautui oikeutetuksi valinnaksi esittämään kohteen valinta-ajan kannalta merkitsevää leveyttä.

Käytännön sovellukset huomioon ottaen oli kuitenkin tarpeen selvittää lisäksi, kuinka kursori menestyy realistisemmassa kaksiulotteisessa ympäristössä erilaisin kohdeasetteluin. Tätä tutkijat selvittivät toisen kokeen avulla, jossa kuplakursoria verrattiin pistekursorin lisäksi kohdeosoittamiseen (katso kohta 3.1). Tässäkin kokeessa

koehenkilöt valitsivat vaihtelevilla etäisyyksillä toisistaan sijaitsevia erikokoisia ympyrän muotoisia kohteita. Kohteen merkitsevää leveyttä kontrolloitiin jälleen sopivasti sijoitelluin ylimääräisin kohtein, nyt niin, että kohteen valinta-alue oli aina optimaalisen lähestymiskulman kanssa linjassa oleva neliön muotoinen alue. Tämä helpottaa Fittsin lain soveltuvuuden selvittämistä kaksiulotteisessa ympäristössä, kun kohteen leveydeksi voidaan suoraviivaisesti tulkita neliön leveys. Muun kuin neliömuotoisten alueiden kohdalla leveyden määrittäminen ei ole yhtä ongelmattonta, vaikka aiheeseen liittyvää tutkimusta onkin olemassa (esimerkiksi [MacKenzie and Buxton, 1992], [Accot and Zhai, 2003] sekä [Grossman and Balakrishnan, 2005b]). Valinta-alueen lisäksi kokeessa kontrolloitiin kohdetiheyttä (erityisen tarkasti juuri 20° sektorilla lähtöpisteestä kohteen suuntaan) kuplakursorin laajenemisen ja pienenemisen mahdollisten vaikutusten huomioon ottamiseksi. Koeasetelmaa havainnollistetaan kuvassa 4 [Grossman & Balakrishnan, 2005a, p. 287].



Kuva 4. Kokeessa kontrolloitiin kohde-etäisyyttä (A), kohdetiheyttä (tarkasti 20° sektorilla lähtöpisteestä kohteeseen) ja lisäksi neljän maalin ympärille sijoitetun kohteen avulla sekä maalin valinta-alueen leveyttä (W) että maalin valinta-alueen merkitsevän leveyden ja leveyden suhdetta (EW/W).

Kohdetiheyden kontrolloimiseksi reitillä kohteeseen näytön kohteet aseteltiin ja piirrettiin aina kokonaan uudestaan onnistuneen valinnan jälkeen.

Myös tämä jälkimmäinen koe osoitti, että kuplakursorilla valinta-alueen leveys oli valinta-aikojen kannalta merkitsevä leveys. Maksimaalista hyötyä valinta-alueesta ei tosin saatu vielä kun leveys oli 33% kohteen leveyttä suurempi, vaikka tällöinkin kuplakursori toimi nopeammin kuin pistekursori. Erityinen havainto oli kuitenkin nimenomaan se, että kuplakursori toimi huomattavan hyvin myös tihein kohdeasetteluin, toisin kuin kohdeosoittaminen, joka kärsi dramaattisesti tiheistä kohdeasetteluista. Kursorityypillä oli tilastollisesti merkitsevä ( $F_{2,22} = 7947$ ,  $p < .0001$ ) vaikutus valinta-aikaan kuplakursorin ollessa huomattavasti muita kursoreita nopeampi.

Kursorien keskimääräiset valinta-ajat olivat 1.41 s. (kohdeosoittaminen), 1.08 s. (piste-kursori) ja 0.93 s. (kuplakursori).

### **4.3. Kuplakursorin ongelmat**

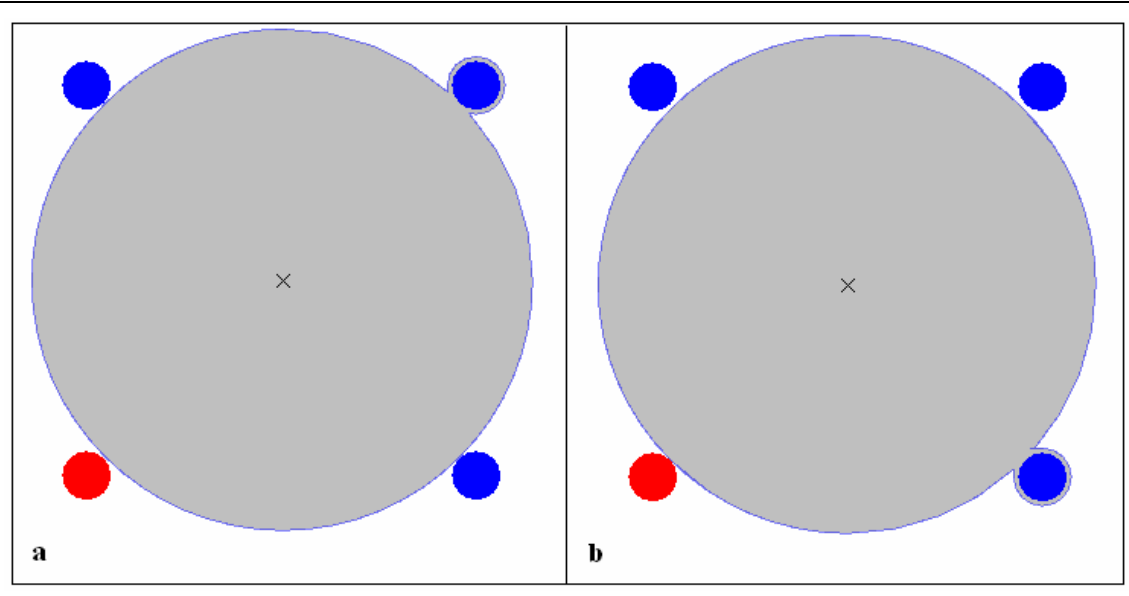
Grossman ja Balakrishnan havaitsivat kuplakursorissa myös joitakin kehityskohtia. Erityisesti kursorin suuri koko vaikuttaisi olleen ongelmallinen useastakin syystä.

Eräs mielenkiintoinen havainto oli, että kuplakursori hidastui hieman kun kohdettiheys oli pieni. Tutkijoiden perustellulta kuulostava analyysi oli, että kun kohteet ovat kaukana toisistaan kursori kasvaa häiritsevällä tavalla hyvin suureksi, mikä hidastaa kursorin liikuttamista. Mahdollisina ratkaisuuina tutkijat pitivät tietyn maksimikoon asettamista kursorin koolle tai vaihtoehtoisesti kursorin muodon mukauttamista kursorin kasvaessa suuren koon vaikutusten vähentämiseksi [Grossman & Balakrishnan, 2005a]. Voisi myös väittää, että erityisen suureksi kasvaessaan kursori ei ole esteettisesti erityisen miellyttävän näköinen.

Toinen tutkijoidenkin mielenkiintoiseksi havaitsema mahdollisuus kuplakursorin kehittämiseksi oli kokeileminen eri kursorin muodoilla ympyrän sijaan kursorin nopeuttamiseksi. Kiinteästi ympyrän muotoisena kursori osuu matkalla kohteeseen herkemmin myös suureen joukkoon muita kohteita. Jos kuplakursori laajenisi enemmän nimenomaan hiiren liikkeiden suuntaan ja vähemmän hiiren liikkeiden suuntaan nähden sivusuunnassa kuten ympyrän tapauksessa, kohteiden merkitsevät leveydet kasvaisivat edelleen.

Turhien valintojen vähentyminen matkan varrella voisi nopeuttaa kuplakursoria myös sen vuoksi, että ylimääräisten osumien myötä esiintyvä suurpiirteinen edestakainen laajeneminen ja kutistuminen saattaa häiritä käyttäjää, kuten Grossman ja Balakrishnan tiedostivat. Paitsi että ylimääräiset kohdevalinnat ovat turhia, ne voivat olla käyttäjälle toisinaan myös varsin odottamattomia, jos valinta tapahtuu liikesuuntaan nähden kaukana sivulla. Tämänkin voisi kuvitella toisinaan häiritsevän käyttäjää. Olennaisempaa on kuitenkin se, että ennakoimattomien valintojen voisi kuvitella myös lisäävän virhevalintoja.

Virhevalintoja voisi kuvitella syntyvän erityisesti tilanteissa, joissa kursori on lähellä useaa kohdetta. Jos juuri ennen halutun kohteen valintaa, käyttäjän odottaessa palautetta kohteen valinnan vaihtumisesta kursori osuu odottamattomasti johonkin muuhun kohteeseen, käyttäjä voi hiiren painiketta painamalla tulla epähuomiossa valinneeksi väärän kohteen (kuva 5).



Kuva 5. Ennen alavasemmalla sijaitsevaan kohteeseen osumista (a) pienikin liike sivulle johtaa siihen, että kursori osuu väärään kohteeseen (b).

Riippumatta siitä, kuinka paljon tällaisia virheitä tapahtuu, on jokseenkin ongelmallista, että kuplakursori voi olla lähellä useaa kohdetta samanaikaisesti, eikä kursorin perusteella ole aina selvää, mikä kohde on lähinnä tulla valituksi seuraavaksi. Koska kuplakursori ulottuu aina johonkin kohteeseen ja kohteet ovat kaikki virtuaalisesti vierekkäin, jokainen epäonnistunut valinta ei vain tarkoita sitä, että käyttäjä klikkaa ohi kohteen, vaan että käyttäjä klikkaa väärää kohdetta. Väärän kohteen klikkaamisella voi ymmärrettävästi olla huomattavasti suurempi haittavaikutus käytännön sovelluksessa kuin tyhjään tilaan osuneella harhaosumalla.

Vaikka kuplakursorin virhealttiudessa voi olla parantamisen varaa, on todettava, että kuplakursoria käyttäessään koehenkilöt tekivät suhteellisen vähän virheitä. Kuplakursorin virheprosentti oli 2.35% yksiulotteisessa ympäristössä suoritetuissa kokeissa ja 1.58% kaksiulotteisessa ympäristössä suoritetuissa kokeissa (molemmat alle tyypillisesti Fittsin lakia soveltavissa tutkimuksissa käytetyn 4% virherajan [MacKenzie, 1992]). Tämä huomioon ottaen voi olla, että huomattavaa parannusta virhealttiudessa voi olla vaikea saavuttaa.

## 5. Laiska kuplakursori ja tötterökursori

Tämän tutkimuksen tavoitteena on ollut kehittää kuplakursorin käyttämistä ideoista vaihtoehtoinen kohdevalintamenetelmä, joka poistaisi osan kuplakursorin ongelmista. Uuden cursorin kehittämisen kannalta keskeisiksi ongelmiksi kuplakursorissa havaittiin erityisesti kuplan suuri koko sekä kuplakursorin laajenemisstrategia, joka sallii suuren määrän turhia, mahdollisesti odottamattomia osumia muihin kohteisiin matkalla kohteeseen. Mainittuja ominaisuuksia lienee mahdollista kehittää varsin suoraviivaisesti muuttamalla kursoria *laiskemmaksi* niin, että kupla pysyy pienempänä, eikä ulotu mihinkään kohteeseen, ennen kuin lähinnä kursoria oleva kohde on selvästi lähin kohde. Laiskan kuplakursorin pienempi koko pienentää kuitenkin myös kohteiden valinta-alueiden leveyksiä, millä tiedetään olevan kohdevalintaa hidastava vaikutus. Tämän ongelman huomioon ottamiseksi kehitin myös muunnelman laiskasta kuplakursorista, joka on muuten täsmälleen samanlainen kuin laiska kuplakursori, mutta säilyttää eräänlaisen hännän avulla valinnan viimeksi valittuun kohteeseen, kunnes kursori osuu johonkin toiseen kohteeseen.

Laiskan kuplakursorin ja tötterökursorin suunnittelussa käytettyihin lähtökohtiin sekä tötterökursorin tehokkuuden mallintamiseen paneudutaan tarkemmin kohdissa 5.1 ja 5.2. Luvun päätteeksi, kohdassa 5.3, esitellään kursorien toteutus.

### 5.1. Laiska kuplakursori

Keskeinen lähtökohta suunnittelussa oli rajoittaa kursorin kokoa. Vaikka suuri koko johtaa nopeampiin valintoihin, se aiheuttaa myös erinäisiä ongelmia, joita esiteltiin jo edellä kohdassa 4.3. Kuplakursori esimerkiksi hidastui kun kohteita ei ollut tiheässä, mikä Balakrishnanin ja Grossmanin käsityksen mukaan johtui siitä, että kursori kasvaa häiritsevällä tavalla hyvin suureksi kohteiden ollessa kaukana toisistaan. Intuitiivisesti oletus vaikuttaa perustellulta ja ylisuuri kursori joka tapauksessa hidasti valintaa. Kabbash ja Buxton vertasivat aluekursoria välineenä osua pieniin kohteisiin osuvasti karpäslätkään. Samanlaiseen kielikuvaan perustuen karpästä ei tunnetusti kuitenkaan kannata ampua tykillä.

Toinen keskeinen lähtökohta oli vähentää turhien kohteiden valintaa. Tämä on mahdollista jos kursori ei laajene tasapuolisesti joka suuntaan vaan suosii liikkeen suuntaa, tai jos jotenkin muuten asetetaan kohteen valinnalle tiukempia ehtoja kuin se, että kohde on yksinkertaisesti lähinnä kursoria. Kolmas asia oli virheiden määrän vähentäminen. Vaikka kuplakursori ei ollut erityisen virhealtis, sen suunnittelussa on tekijöitä, jotka tukevat virheiden tapahtumista.

Asetettuihin tavoitteisiin, kursorin koon pienentämiseen sekä turhien kohdevalintojen ja virheiden vähentämiseen pyrittiin ensisijaisesti yhdellä muutoksella kuplakursoriin. Kursorin kokoa rajoitettiin niin, että kursoria estettiin ulottumasta

mihinkään kohteeseen, ellei ole olemassa yhtä kohdetta, joka on selvästi muita kohteita lähempänä. Se, että kursori on yksinkertaisesti lähinnä jotain kohdetta ei vielä ole vahva viite siitä, että kohdetta oltaisiin valitsemassa, ja tästä syystä kursorin koko pysyy maltillisesti kuplakursoria pienempänä. Vasta kun on olemassa kohde, joka on selvästi muita lähempänä, kuplan koko kasvaa niin, että se ulottuu lähimpään kohteeseen. Kuten edellä todettiin, kursorin pienemmän koon saattoi toivoa johtavan siihen, että kursoria on vaivattomampi liikuttaa. Se, että on aina olemassa korkeintaan yksi kohde, joka on lähellä tulla valituksi saattaa puolestaan pienentää kursorin virhealttiutta ja puoltaa kursorin ennakkoluulotonta liikuttamista myös silloin, kun kursori on suuri.

Vaikka laiska kuplakursori vähentää tarpeettomia osumia kohteisiin ja pienempi koko saattaa johtaa siihen, että kursoria liikutetaan nopeammin ja menetelmä on vähemmän virhealtis, pienemmällä koolla on väistämättä myös se negatiivinen vaikutus, että osa kohteiden valinta-alueista menetetään. Kursorin ei oletettukaan kilpailevan tasaväkisesti kuplakursorin kanssa sellaisenaan.

## **5.2. Tötterökursori**

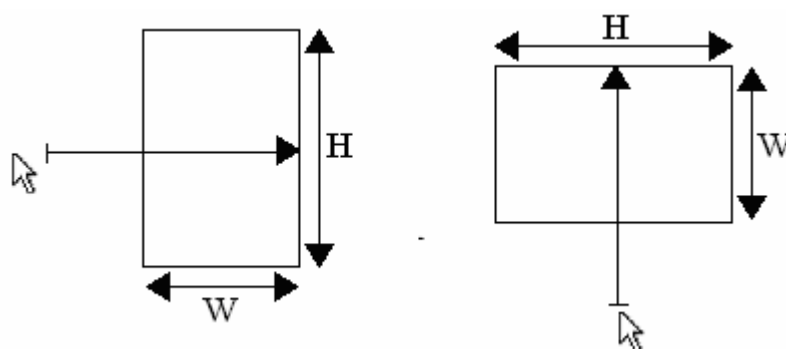
Valinta-alueita, jonka laiska kuplakursori menettää pienen kokonsa vuoksi, voisi kuitenkin olla mahdollista kompensoida säilyttämällä valinta kohteeseen, johon kursori on viimeksi osunut, vielä senkin jälkeen kun kupla ei enää ulotu kohteeseen. Valinnan säilyttämisen myötä valinta-alue kasvaa laiskaan kuplakursoriin nähden, joka kuplan menettäessä kosketuksen kohteeseen menettää myös mahdollisuuden kohteen klikkaamiseen. Lähtötilanteessa valittavan kohteen valinta-alue on tötterökursorilla täsmälleen saman kokoinen kuin laiskalla kuplakursorilla, mutta kun kohteeseen osuu, sen valinta-alue kasvaa. Valinnan säilyttäminen voitaisiin visualisoida esimerkiksi venyttämällä kursoria tavalla tai toisella. Tötterökursorin tapauksessa kupla yhdistetään kohteeseen eräänlaisen hännän avulla.

Valinnan säilyttäminen kuvatulla tavalla voi kuitenkin olla myös ongelmallista. Vaikka kupla on tavoitteen mukaisesti pienempi, tötterökursori voi olla todella pitkä ja pinta-alaltaan varsin suuri. Valinnan säilyttämisen myötä tötterökursori on aina myös kosketuksessa johonkin kohteeseen ja klikkaukset kohteen ohi johtavat jälleen automaattisesti väärän kohteen klikkaamiseen. Lisäksi saattaa olla häiritsevää, että kupla yhdistetään uuteen kohteeseen myös jos häntä osuu johonkin kohteeseen. Tämä on oletettavasti kuitenkin suhteellisen pieni ongelma. Hännän osuminen uuteen kohteeseen ei vaikuta mitenkään siihen, kuinka kuplan koko määrittyy ja ylimääräiset osumat kursorin hännällä näin ollen tuskin ovat ainakaan yhtä häiritseviä kuin uuteen kohteeseen osuminen kuplalla.

Entä kuinka paljon tötterökursorin voi odottaa kompensoivan kuplakursoriin nähden pienempää valinta-alueita? On epäselvää, kuinka paljon valinta-alueen laajenemisesta todellisuudessa on hyötyä, kun laajeneminen tapahtuu vasta kun

kohteeseen on osunut ensimmäisen kerran. Kuinka kasvava alue tulisi suhteuttaa Fittsin lakiin? Joitakin oletuksia on mahdollista tehdä aiemman tutkimuksen perusteella.

Accot ja Zhai [2003] esittivät, että kun leveytenä tulkitaan kohteen leveys liikesuunnassa ja korkeutena vastaavasti kohteen leveys liikkeen kanssa poikittaisessa suunnassa (kuva 6), alueen leveydellä on vaikutus valintanopeuteen senkin jälkeen, kun leveys on suurempi kuin korkeus, mutta ei toisin päin.



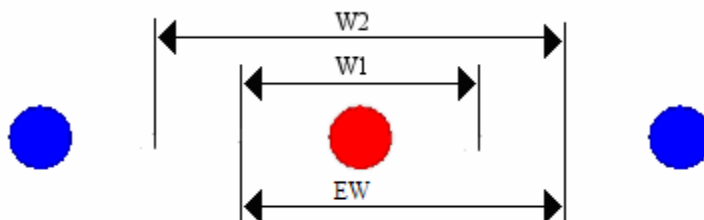
Kuva 6. Kohteen leveyden (W) ja korkeuden (H) tulkinta liikesuunnassa.

Samoin Grossman ja Balakrishnan havaitsivat, että kohteen leveys liikesuunnassa on merkitevin tekijä valinta-ajan suhteen, mutta Accotin ja Zhain kanssa ristiriitaisesti, että korkeudella voi olla vaikutus valinta-aikaan senkin jälkeen kun korkeus kasvaa suuremmaksi kuin leveys [Grossman & Balakrishnan, 2005b]. Näiden havaintojen perusteella voisi olettaa, että tötterökursori hyötyisi kasvavasta alueesta eniten tilanteissa, joissa valinta-alue kasvaa erityisesti liikesuuntaan (tulosuunnasta nähden valittavan kohteen takana on paljon tyhjää tilaa). Toisaalta havainnot antavat myös syytä olettaa, että jos kohteen takana on vain vähän tilaa eikä valinta-alue kasva paljoa liikesuunnassa, tyhjästä tilasta kohteen sivuilla on korkeintaan pientä hyötyä.

Grossman ja Balakrishnan [2005b] huomasivat myös, että ballistiset nopeat liikkeet, joilla yritetään suoraan osua kohteeseen päätyvät noin kaksi kertaa pidemmälle alueelle liikesuunnassa kuin liikkeen suuntaan nähden poikittaisessa suunnassa. Tämäkin tukee päättelyä valinta-alueen kasvamisen hyödyistä kohteen takana. Jos kohteen valitsemiseksi riittää karkeasti ottaen, että kursori päätyy kohteen taakse ennen seuraavaa kohdetta, käyttäjät voisivat oppia käyttämään kursoria optimistisin liikkein niin, että kursori yritetään vain siirtää alkuperäiselle valinta-alueelle tai sen taakse.

Kohteen valinta-alueeksi voisi parhaassa tapauksessa kenties arvioida suurinpiirtein sen alueen, jolle lähtöpisteestä katsoen suoraan kohdetta kohti liikuttaessa kohdistin olisi siirrettävä, jotta kohde olisi valittuna hiiren liikkeen pysähtyessä. Kuva 7 havainnollistaa päättelyä. Selkeyden vuoksi kuvassa on esitetty kohdevalintatehtävä yksiulotteisessa ympäristössä, jossa kohteet sijaitsevat tasaisin välein ja kohteet sekä kursorin liikkeet on rajoitettu x-akselille. Alkutilanteessa kursori on vasemman puoleisessa kohteessa. Keskimmäisen kohteen valitsemiseksi kursori on ensin siirrettä-

vä alkuperäiselle valinta-alueelle (kuvan W1), jolloin valinta-alue, jolla klikkauksen on tapahduttava, laajenee (W2). Liikkeen suunta huomioon ottaen valinta-ajan kannalta merkitsevä valinta-alue voisi oletuksen mukaan olla suurinpiirtein kuvan alue EW.



Kuva 7. Keskimmäisen kohteen alkuperäinen valinta-alue (W1), kohteen laajentunut valinta-alue (W2) ja oletettu valinta-ajan kannalta merkitsevä valinta-alue (EW).

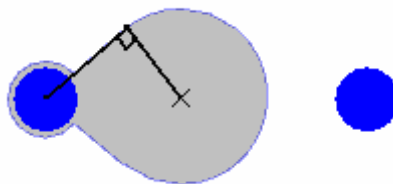
Vaikka näin määritellystä aktivointialueesta saataisiin täysi tehokkuushyöty ja kuvassa 7 esitetyllä tavalla valinta-alueen koko säilyisi itse asiassa yhtä suurena kuin kupla-kursorin tapauksessa, matka kohteeseen kasvaa. Näin on, koska kohteen valinta-alue on kauempana kuin kuplakursorin tapauksessa (kuvassa 7 keskimmäisen kohteen valinta-alue alkaisi kuplakursoria käyttäen vasemmanpuoleisen kohteen ja keskimmäisen kohteen puolesta välistä). Vaikka itse valinta-alue voisi edellä kuvatussa tilanteessa siis lopulta säilyä kooltaan lähes ennallaan, on otettava huomioon, että valinta-alue ei kaikilla kohdeasetteluilla olisi samankokoinen kuin kuplakursorilla. Jos esimerkiksi lähellä kohdetta, sen takana, on toinen kohde, valinta-alue ei kasvaisi juuri lainkaan. Tällaisessa tilanteessa kohteen valinta-alue alkaa edelleen myöhemmin kuin kuplakursorin tapauksessa mutta ei missään vaiheessa kasva paljoa, ja alue jää huomattavasti kuplakursorin valinta-aluetta pienemmäksi. Toisaalta sama pätee myös toisinpäin. Jos kohde on lähellä, mutta sen jälkeen liikesuunnassa on paljon tyhjää tilaa ennen seuraavaa kohdetta, kohteen valinta-alue voi kasvaa kuplakursorinkin valinta-aluetta suuremmaksi. Välimatka kohteeseen kuitenkin kasvaa aina. Ainoa vahva oletus onkin se, ettei tötterökursori voi toimia ainakaan nopeammin kuin kuplakursori. Tehokkuuseron voisi kuitenkin kuvitella jäävän suhteellisen pieneksi, jolloin tötterökursori voi olla suositeltavissa muin perustein.

### 5.3. Kursorien toteutus

Uusien kursorien toteutuksessa olen seurannut pitkälti alkuperäistä kuplakursorin toteutusta. Kupla on ympyränmuotoinen, puoliksi läpinäkyvä, tässä tapauksessa harmaa alue, jonka reunat eivät kuitenkaan ole läpinäkyviä ja ovat väriltään siniset. Samoin kuin kuplakursorissa, kuplan keskustassa on lisäksi pieni rasti, joka toimii visuaalisena vihjeenä pisteestä, jota hiirellä itse asiassa liikutetaan. Kursorin osuessa johonkin kohteeseen kursori yhdistetään siihen visuaalisesti piirtämällä kohteen ympärille puoliksi läpinäkyvä harmaa kehys sinisin reunoin. Tötterökursorin toteutus on muuten



identtinen laiskan kuplakursorin kanssa, mutta valinta kohteeseen säilytetään hänen avulla, joka piirretään kuplasta kohteeseen (kuva 8).



Kuva 8. Tötterökursori piirretään yhdistämällä kupla ja kuplan sivuilta kohteen keskustaan piirrettyjen janojen ja kuplan välinen alue, kuplan häntä.

Kursorien toteutuksessa poikettiin myös muutamassa kohdassa alkuperäisestä kuplakursorin toteutuksesta. Kuplakursorikin toteutettiin vain sillä suoritettuja kokeita varten, eivätkä valitut toteutusratkaisut värien ja muiden visuaalisten yksityiskohtien kohdalla välttämättä olleet parhaita mahdollisia. Kursoria reunustava sininen reunus sekä kuplakursorin alkuperäisessä toteutuksessa uuden kohteen valintaan liittyneiden animaatioiden (kohdetta ympäröivä kehys piirrettiin kohteen ympärille animoidusti) pois jättäminen ovat esimerkkejä ratkaisuista, joissa ei seurattu tarkasti kuplakursorin toteutusta.

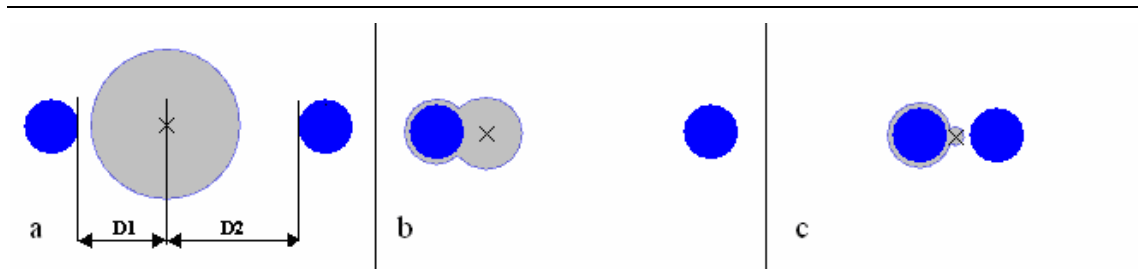
Olellainen ero kuplakursoriin löytyy algoritmista, joka laskee kuplan säteen kursorin jokaisen liikkeen jälkeen. Algoritmi on identtinen laiskalle kuplakursorille ja tötterökursorille; se laskee kuplan säteen lähimpien kohteiden etäisyyksien mukaan oletusarvoisesti niin, että kupla ulottuu lähimpään kohteeseen jos lähimmän ja toiseksi lähimmän kohteen etäisyyksien suhde alittaa tietyn rajan. Säteen määrittämiselle ei ole muita suoria vaatimuksia. Algoritmi toteutettiin muodossa, jossa kursori pienenee kun kursori lähestyy kohdetta, johon kuplan säde jo ulottuu, ja joka on entistä selvemmin lähin kohde. Tällä ratkaisulla kursorin koko pyrittiin pitämään mahdollisimman pienenä. Lisäksi kuplan kokoa rajoitetaan niin, ettei kursori koskaan pienene liikaa.

Jos etäisyys lähimpään kohteeseen on  $D1$  (etäisyys kursorin keskustasta lähimmän kohteen lähimpään pisteeseen), etäisyys toiseksi lähimpään kohteeseen  $D2$  (etäisyys kursorin keskustasta toiseksi lähimmän kohteen lähimpään pisteeseen), oletusarvoinen kuplan säde voidaan laskea esimerkiksi kaavalla

$$\text{defaultRadius} = \left( \frac{1}{2} + k - \frac{D1}{D2} \right) D1,$$

kuten tässä tutkimuksessa tehtiin. Tässä  $k$  on vakio, joka määritellään siten, että yhtälö tuottaa suurempia arvoja kuin  $D1$  kun lähimmän ja toiseksi lähimmän kohteen suhde

riittää valintaan, muulloin tätä pienempiä arvoja. Kun lähimmän ja toiseksi lähimmän kohteen suhde saa arvoja väliltä  $[0.0, 1.0]$  ja yhtälössä sulkujen sisällä olevan termin on oltava suurempi tai yhtäsuuri kuin 1, jotta kupla ulottuu lähimpään kohteeseen,  $k:n$  on oltava arvo, joka on suurempi kuin  $\frac{1}{2}$  ja pienempi kuin  $1\frac{1}{2}$ . Esimerkiksi  $k:n$  arvolla 1, jota tässä tutkimuksessa käytettiin, lähimmän kohteen on oltava puolet lähempänä kuin toiseksi lähimmän kohteen, jotta kupla ulottuu kohteeseen. Kuvassa 9 (a) kursori ei vielä ole riittävän lähellä vasemmanpuoleista kohdetta, jotta kupla ulottuisi kohteeseen.



Kuva 9. Kupla ei ulotu lähimpään kohteeseen, ellei se ole riittävän selvästi lähinnä oleva kohde (a). Kupla on saavuttanut minimikoon (b). Turvallinen minimikoko (c).

Kun kupla lähestyy kohdetta, johon se jo ulottuu, oletusarvoisen säteen laskeva yhtälö tuottaa säteelle jokseenkin homogeenisia arvoja niin, että kursori, hieman karrikoiden, aina vain juuri ja juuri ulottuu kohteeseen. Koska ei kuitenkaan ole mielekäästä, että kursori pienenee liikaa kohdetta lähestyttäessä (tai kursorin ollessa kohteen päällä katoaa kokonaan), on säteelle määriteltävä sopiva alaraja. Kun alarajana on lähimmän kohteen sädettä hieman suurempi arvo, kursori erottuu aina hieman kohdetta suurempana alueena (kuva 9 (b)). Tällöin on kuitenkin riski, että kursori ulottuu myös toiseksi lähimpään kohteeseen. Siksi minimi on  $\frac{1}{2} D2$  edellisen sijaan aina, kun kursoria lähinnä olevan kohteen säde ( $R1$ ) on suurempi tai yhtä suuri kuin ensimmäinen minimi (9 (c)). Turvallinen minimisäde mahdollistaa kuitenkin kursorin häviämisen kohteeseen. Toteutuksessani kohde säilyy tällöinkin kehystettynä ja kursorin sijaintia esittävä rasti on edelleen näkyvissä.

Oletussäteen laskevaa yhtälöä sovelletaan siis yhdessä mielekkäitä minimiarvoja tuottavan yhtälön kanssa. Tämä yhtälö esitetään algoritmina `minRadius`.

**Algorithm** `minRadius()`

```

if ( $D2 / 2 > R1$ )
    return  $R1$ 
return  $D2 / 2$ 

```

Säteen laskeva yhtälö on lopulta esitettävissä seuraavasti.

$$radius = \max(defaultRadius, minRadius())$$

Tässä *defaultRadius* siis tuottaa kohteeseen ulottuvia säteen arvoja kun lähimmän ja toiseksi lähimmän kohteen etäisyyksien suhde saavuttaa jonkin raja-arvon ja *minRadius()* sopivia minimiarvoja, jotka varmistavat, ettei kursorin koko pienene liikaa.

Oletussäde *defaultRadius* olisi kuitenkin voitu laskea myös hieman toisin. Kuplan säde olisi myös voinut kasvaa, kun lähin kohde on yhä selkeämmin lähin kohde. Säde lasketaan kuitenkin tässä tutkimuksessa tarkoituksella juuri niin, että säde pienenee kohdetta lähestyttäessä, jotta kuplan koko pysyisi pienenä, minkä uskotaan vaikuttavan positiivisesti valinta-aikaan. Ei kuitenkaan ole varmaa, että tämä on paras ratkaisu. Jos kursorin koko kasvaisi kun lähin kohde on entistä selvemmin lähin kohde, kursorin käyttö voisi olla intuitiivisempaa. Valittu strategia saattaa johtaa myös vähemmän loogiselta vaikuttaviin tiheisiin, joskaan ei niin suuriin, edestakaisiin koon muutoksiin.

Tötterökursori säilyttää siis lisäksi myös valinnan kohteeseen, johon se on viimeksi osunut, häntänsä avulla.

## 6. Uusilla kursoreilla suoritettut kokeet

Uusien kursorien toimivuutta testattiin kokeellisesti. Tässä luvussa kuvataan kysymykset joihin kokeissa etsittiin vastauksia, kokeiden suunnittelu ja toteutus sekä tulokset, joita kokeissa saatiin.

### 6.1. Kokeiden tavoitteet

Uusien kursorien toimivuuden selvittämiseksi oli tarpeen tutkia ennen kaikkea kahta asiaa.

- Kuinka suuri tehokkuusero kuplakursorin ja uusien kursorien välillä on?
- Ovatko uudet kursorit muuten suositeltavampia kuin kuplakursori?

Edes tötterökursorin ei, edellä erityisesti kohdassa 5.2 esitellyistä syistä (kuplakursoriin nähden pienempi alkuperäinen valinta-alue, suurempi etäisyys kohteeseen) johtuen, oletettu olevan ainakaan nopeampi kuin kuplakursori. Tötterökursorin ei kuitenkaan välttämättä myöskään tarvinnut olla dramaattisesti hitaampi. Jos ero jäisi pieneksi, tötterökursori voisi olla suositeltavissa muin perustein. Selvitettiin siis kuinka suuri ero kursorien välillä on.

Oletuksen mukaan uudet kursorit saattoivat kuitenkin olla suositeltavia vain, jos ne olisivat muilta osin kuin puhtaasti tehokkuuden kannalta parempia kuin kuplakursori. Oli siis tarpeen selvittää myös eroja virhealttiuksissa ja käyttäjien mieltymyksissä.

Lisäksi kiinnostavaa oli erityisesti se, kuinka paljon valinnan säilyttämisestä lopulta on hyötyä tötterökursorille. Tötterökursorilla kohteen valinta-alue laajenee sille saavuttaessa, minkä oletettiin kompensoivan kuplakursoriin nähden alun perin pienempää valinta-aluetta. Hyötyäkseen suuremmasta valinta-alueesta, todennäköisimmin ampumalla kohteen alkuperäisen valinta-alueen yli, kursoria olisi mahdollisesti kuitenkin käytettävä tavallista optimistisemmin. Pyrittiin siis myös selvittämään, onko laajenevasta valinta-alueesta todella hyötyä tötterökursorille ja kuinka paljon.

### 6.2. Koeasetelma

Koska oletus oli, ettei edes tötterökursori voisi toimia nopeammin kuin kuplakursori, koeasetelmaa ei nähty tarpeelliseksi laatia niin, että kokeiden avulla voitaisiin vahvistaa kursorien toiminta suhteessa Fittsin lakiin. Koeasetelman suunnittelussa tavoiteltiin ensisijaisesti vain selkeästi kontrolloitua, eri kohdetiheydet sallivaa kaksikulotteista kohdeympäristöä.

Koehenkilöitä pyydettiin kokeissa valitsemaan toistuvasti tasaväliseen ruudukkoon valituista kohteista punaisena erottuva kohde. Koehenkilöt suorittivat kohdevalintoja käyttäen pistekursoria, laiskaa kuplakursoria, tötterökursoria sekä kuplakursoria.

Laiskan kuplakursorin ei oletettu menestyvän kokeissa tötterökursoria paremmin, mutta se oli syytä ottaa osaksi koetta sen vuoksi, että sen avulla oli mahdollista selvittää hyötyä, jonka tötterökursori saa valinnan säilyttämisestä.

Kursorien käytöstä automatisoidusti kerätyn datan lisäksi kokeissa selvitettiin koehenkilöiden tuntemuksia kursoreista kyselylomakkeiden avulla. Koeasetelma kuvataan kokonaisuudessaan yksityiskohtaisemmin alakohdissa.

### **6.2.1. Koeympäristö**

Kokeet suoritettiin 3.2 Ghz Pentium 4 PC:llä Windows (XP) ympäristössä. Näyttö oli 1600x1200 resoluutioinen LCD-näyttö. Kursoria kontrolloitiin langallisella, optisella, normaalin kokoisella Logitech USB-hiirellä. Testisovellus oli toteutettu Java™-kielellä. Koehenkilöt liikuttivat hiirtä pöytään teipatun A3-kokoisen valkoisen paperiarkin päällä.

Kokeita varten oli myös tarpeen valita sopiva raja sille, kuinka selvästi lähimmän kohteen on oltava toiseksi lähintä kohdetta lähempänä, jotta kuplan säde uusilla kursoreilla ulottuu kohteeseen ja kohde tulee valituksi. Informaalit kokeilut muutamilla eri arvoilla eivät tuottaneet selviä eroja nopeuden suhteen, joten raja-arvoksi valittiin sopivan selkeä, ei erityisen pieni taikka suuri, silti selvästi kuplakursorista eroava 1:2. Näin lähimmän kohteen oli oltava puolet lähempänä kuplan keskustaa kuin toiseksi lähimmän kohteen, jotta kupla ulottui kohteeseen.

Huomioimisen arvoista koeympäristössä on vielä se, että hiiren ja näytöllä näkyvän kursorin liikkeiden suhdetta ei määritelty eksplisiittisesti. Näin ollen Windows XP:n automaattisesti käyttämät hiiren ja näytöllä näkyvän kursorin liikkeiden suhdetta optimoivat algoritmit ovat väistämättä vaikuttaneet kursorin käyttäytymiseen, kun kokeissa epähuomiossa huolehdittiin vain siitä, että Windowsin hiiridialogin nopeusasetus oli aina haluttu. Mainitut algoritmit säätävät Windows XP:ssä hiiren ja näytöllä näkyvän kursorin liikkeiden suhdetta matalilla hiiren liikenopeuksilla. Sillä, että tämä suhde ei ollut vakio vaan käyttäjärjestelmä sääti sitä dynaamisesti, voi olla pieni merkitys tulosten kannalta. Koska tilanne oli sama kaikille kursoreille ja kursorit, pistekursoria lukuun ottamatta, toimivat jokseenkin samalla tavalla, ei liene syytä olettaa, että merkitys olisi ollut suuri. Erityisesti muista kursoreista poikkeava pistekursori, jolle algoritmit on suunniteltu, saattoi kuitenkin saada tästä ylimääräistä hyötyä.

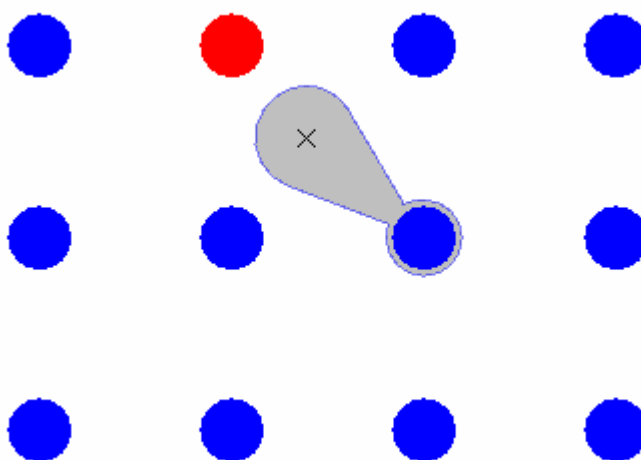
### **6.2.2. Koehenkilöt**

Kokeeseen osallistui 18 koehenkilöä, 9 miestä ja 9 naista. Iältään koehenkilöt olivat 18–26-vuotiaita. Kaikki koehenkilöt olivat oikeakätisiä eikä kukaan heistä ollut värisokea. Koehenkilöt oli velvoitettu osallistumaan testiin suoritusmerkinnän saamiseksi eräällä vuorovaikutteisen teknologian kurssilla, eivätkä he saaneet muuta kompensatiota

osallistumisesta. Poikkeuksena tästä, koehenkilöiden loputtua kesken, kaksi alkuperäisten koehenkilöiden ulkopuolelta hankittua koehenkilöä saivat minimaalisen rahallisen kompensaation osallistumisesta.

### 6.2.3. Proseduuri

Koehenkilöiden tehtävänä oli valita kaksiulotteisessa ympäristössä tasaväliseen, koko ikkunan täyttävään ruudukkoon sijoitetuista ympyrän muotoisista kohteista punaisena erottuva kohde (kuva 10). Muut kohteet olivat sinisiä ja onnistuneen valinnan jälkeen uusi kohde näkyi punaisena ja juuri valittu kohde muuttui jälleen siniseksi.



Kuva 10. Kohteet oli kokeissa sijoitettu koko ruudun kokoiseen tasaväliseen ruudukkoon. Valittava kohde erottui muista kohteista punaisena.

Käyttäjälle annettiin palautetta tehtävän aikana visuaalisesti näyttämällä kohde mustana kun kursori oli sen päällä ja hiiren painike oli painettuna alas, sekä kaiuttimista järjestelmän *beep*-äänellä kun hiiren painikkeen vapauttaminen johti virhevalintaan (valinnan onnistumiseksi hiiren painike piti sekä painaa alas että vapauttaa oikean kohteen päällä).

### 6.2.4. Riippumattomat muuttujat

Kursorikondition lisäksi kokeessa kontrolloitiin vain yhtä riippumatonta muuttujaa, kohdetiheyttä. Tiheydet määriteltiin niin, että kahden vierekkäisen kohteen (vaaka- tai pystysuorassa) keskipisteiden etäisyys ruudukossa oli käytetystä tiheydestä riippuen aina joko 64, 96, 160 tai 256 pikseliä. Kohteen koko säilyi vakiona, kohteiden ollessa aina 32 pikseliä halkaisijaltaan. Tyhjää tilaa kohteiden välillä oli näin ollen tiheydestä riippuen aina joko 1, 2, 4 tai 7 kertaa kohteen koon verran.

### 6.2.5. Kokeiden suorittaminen

Kohdevalintoja suoritettiin kokeissa sarjoittain neljällä eri kursorilla. Sarjojen sisältämät valinnat oli arvottu etukäteen ja ne toistuivat jokaiselle koehenkilölle ja kursorikonditiolle samoina. Koska valintojen etäisyydet ja liikkeiden suunnat olivat satunnaisia, sarjat koostuivat jokseenkin suuresta määrästä (20 kappaletta) valintoja. Sarja alkoi kuitenkin aina vasta ensimmäisestä onnistuneesta valinnasta, eli yksi sarja sisälsi teknisesti ottaen 21 onnistunutta kohdevalintaa.

Jokaisella kursorilla suoritettiin neljä eri sarjaa jokaisella tiheydellä. Koe sisälsi siis 4 kursorikonditiota {pistekursori = PC, laiska kuplakursori = LB, tötterökursori = CC, kuplakursori = BC} x 4 tiheyttä {kohdeväli 1, 2, 4, 7 x kohdekoko} x 4 toistoa eli kokonaisuudessaan 64 sarjaa á 20 valintaa. Sarjat suoritettiin kursori kerrallaan vaihdellen tasaisesti kolmen viimeisen kursorin järjestystä koehenkilöittäin. Ensimmäinen kursori oli kuitenkin aina pistekursori. Näin sen vuoksi, että informaalit kokeilut ennen kokeita antoivat syytä olettaa, että kokeita suoritettaessa saattaa esiintyä asymmetristä oppimista. Jos muita kursoreita käytettiin ennen pistekursoria, siirtyminen pistekursorin käyttöön näytti johtavan kasvaneeseen virheiden määrään. Pistekursorin oletettiin myös olevan huomattavasti muita kursoreita hitaampi, jolloin mahdollinen pieni vääristymä kursorin nopeudessa muihin kursoreihin nähden ei olisi tulosten kannalta olennainen. Konditioittain varsinaiset koesarjat tulivat suoritettavaksi satunnaisessa järjestyksessä, kuitenkin niin, että 1) kaikilla tiheyksillä suoritettiin aina ensin yksi sarja ennen kuin mitään tiheyttä käyttäen suoritettiin toinen sarja ja 2) mitään tiheyttä käyttäen ei suoritettu kahta sarjaa peräkkäin.

Jokaisen kursorikondition aluksi koehenkilöt suorittivat aina ensin yhden harjoitussarjan 96 pikselin kohdevälillä (tiheys = 2) ennen varsinaisten koesarjojen alkua. Kursorikondition vaihtuessa koehenkilö joutui viettämään pakollisen lyhyen lepotauon, jonka aikana hän sai myös täytettäväkseen kyselylomakkeen juuri käyttämästään kursorista (pistekursoria ei tarvinnut arvioida, mutta minuutin lepotauko oli silti pakollinen). Koska todellista lepoa ei lomakkeen täyttämistä johtuen saanut, koehenkilölle annettiin mahdollisuus levätä vielä hetki ennen seuraavaa kursorikonditiota. Kokeen päätteeksi koehenkilöt täyttivät vielä yhden lomakkeen, jossa pyydettiin asettamaan kursorit paremmuusjärjestykseen.

Jokaisen sarjan jälkeen järjestelmä kysyi käyttäjältä, oliko tämä valmis jatkamaan. Ennen jatkamista koehenkilö sai halutessaan levähtää hetkeksi ja seuraavan sarjan aloitusnäkyvä tuli esiin kun käyttäjä koki olevansa valmis jatkamaan ja klikkasi ok-painiketta. Kursorikonditio vaihtui kun koehenkilö oli suorittanut kaikki 16 sarjaa yhdellä kursorilla.

Grossmannia ja Balakrishnania seuraten koehenkilöt ohjeistettiin suorittamaan testit ”mahdollisimman nopeasti ja tarkasti”. Kokeiden suorittamiseen kului kokonaisuudessaan keskimäärin noin 40 minuuttia henkilöä kohden.

Koeasetelma erosi lopulta huomattavasti kuplakursorin testauksessa käytetystä asetelmasta (katso kohta 4.2). Ympäristö ei täysin vastannut realistista ympäristöä, koska erityisen helpot ja vaikeat valinnat eivät olleet edustettuina, mutta kaksiulotteiseen tasaväliseen ruudukkoon sijoitetut kohteet mahdollistivat kursorien yleisluonteisen vertailun ilman, että koeasetelmasta olisi tullut kohtuuttoman vaikeasti kontrolloitava. Kohdetiheyttä vaihtelemalla saatiin myös näkyviin kursorien koonmuutokset, joiden oletettiin mahdollisesti vaikuttavan kursorien käyttöön, erityisesti kursorin tehokkuuden tai käytön miellyttävyyden kannalta, mutta mahdollisesti myös virheiden suhteen.

Jokaisella tiheyden ja kursorikondition yhdistelmällä suoritettiin siis 4 toistoa á 20 valintaa, eli yhteensä 80 eri valintaa. Valintojen määrää olisi voinut olla perusteltua kasvattaa 240 valintaan per kursorikonditio-tiheys yhdistelmä. Tämä olisi ollut mahdollista ilman, että kokeiden olisi tarvinnut kestää pidempään, jos koehenkilöt, jotka suorittivat kokeen käyttäen samaa kursorikonditioiden järjestystä (3 jokaisella 6 eri järjestyksellä), olisivat aina suorittaneet eri sarjat. 80 satunnaisen valinnan katsottiin kuitenkin olevan riittävä määrä ja kaikki koehenkilöt suorittivat testit samoin valinnoin.

#### **6.2.6. Käyttömukavuuden arviointi kyselylomakkeella**

Uudet kursorit olivat oletuksen mukaan suositeltavissa vain jos käyttäjät pitäisivät niitä parempina kuin kuplakursoria. Tekijöitä, joiden uskottiin vaikuttavan kursorin käytön mielekkyyteen olivat ainakin subjektiiviset tuntemukset kursorin nopeudesta, virheettömien valintojen suorittamisen helppoudesta, kursorin käyttäytymisen loogisuudesta, sekä kursorin kokoon, muotoon ja niiden muutoksiin liittyvistä esteettisistä seikoista. Koehenkilön suoritettua kaikki testit jollakin muulla kokeen kursorikonditiolla kuin pistekursorilla hänelle annettiin täytettäväksi kursorin käytöstä saatuja tuntemuksia tiedusteleva kyselylomake (liite 1). Lomakkeessa esitettiin viisi kysymystä, joissa kysyttiin,

- kuinka nopea kursori oli välineenä kohteiden valitsemiseksi,
- kuinka helppoa kursoria käyttäen oli suorittaa kohdevalinta ilman virheitä,
- kuinka loogiselta kursorin käyttäytyminen tuntui,
- kuinka häiritsevältä kursorin koko ja muoto sekä niiden muutokset tuntuivat ja
- kuinka miellyttävältä kursorin käyttö tuntui kaikkiaan.

Vastaukset pyydettiin merkitsemään lomakkeelle rastimalla likert-asteikolla annetuista vaihtoehdoista 1-7 (negatiivisesta positiiviseen) oikea kohta kuhunkin kysymykseen. Lisäksi lomakkeessa pyydettiin käyttäjää kertomaan vapaamuotoisesti, mitä ajatuksia kursorin käyttö herätti. Lomakkeeseen oli varattu tyhjää tilaa tekstiä varten.



Kyselylomaketta ei kuitenkaan tarvinnut täyttää ensimmäisen kursorin, pistekursorin, osalta. Näin sen vuoksi, että pistekursorin arvioimisen ennen muita kursoreita ajateltiin mahdollisesti piilottavan eroja kursorien välillä, kun käyttäjillä ei välttämättä ollut objektiivista näkemystä pistekursorin toimivuudesta muihin, käyttäjälle ennestään tuntemattomiin kohdevalintamenetelmiin nähden. Käyttäjät olisivat esimerkiksi saattaneet arvioida pistekursorin varsin toimivaksi ja nopeaksi menetelmäksi valita kohteita näytöllä, minkä jälkeen, vaikka he olisivatkin havainneet myöhemmin esimerkiksi kuplakursorin vielä nopeammaksi, ero ei välttämättä olisi näkynyt likert-asteikolla annetuissa vastauksissa.

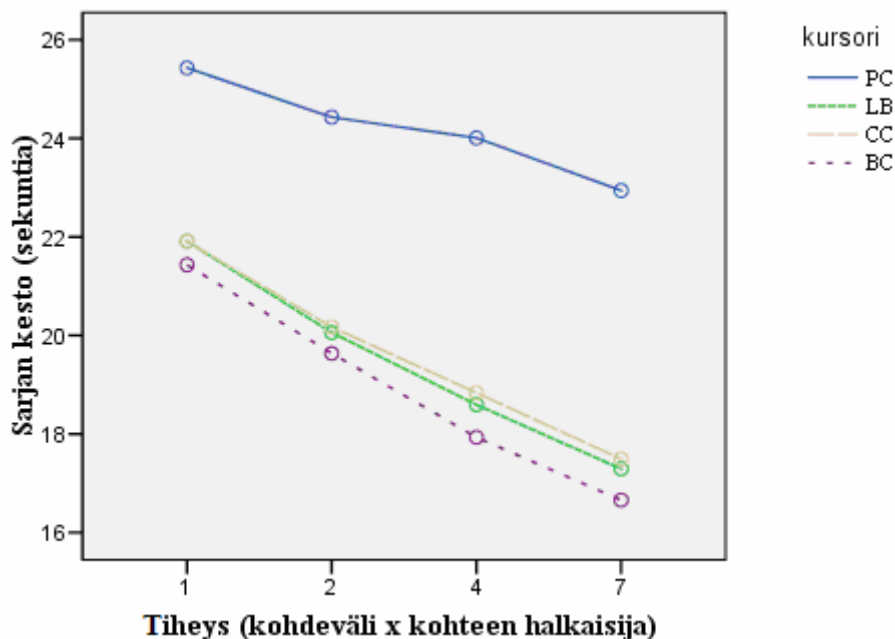
Lomakkeet olisi voinut olla mielekäästä täyttää vasta kun käyttäjä oli suorittanut kaikki kokeet jokaisella kursorilla, mutta kysymyksiin vastaaminen olisi voinut olla liian vaikeaa vasta kauan sen jälkeen kun jotakin kursoria oli alun perin käyttänyt. Kun lomake täytettiin välittömästi sen jälkeen kun koehenkilö oli saanut suoritettua kaikki sarjat tietyllä kursorilla, tuntemuksien kursoreista uskottiin myös säilyvän paremmin kokeen loppuun saakka. Kun koehenkilö lopulta oli suorittanut kaikki testit, hänelle annettiin vielä yksi kyselylomake (liite 2), jossa kursorit pyydettiin sijoittamaan paremmuusjärjestykseen. Sen varalta, että käyttäjä ei olisi muistanut järjestystä, jossa kursoreita käytti, käyttäjälle tarjottiin mahdollisuutta saada kursorit näkyviin jos hän ei ollut varma järjestyksestä. Yksikään koehenkilö ei kokenut tarvitsevansa muistin virkistystä vaan koehenkilöt muistivat ongelmitta järjestyksen, jossa olivat kursoreita käyttäneet.

### **6.3. Tulokset**

Ensisijaisesti kokeissa kerättiin dataa valintasarjojen valinta-ajoista, virheistä ja klikkausten sijainnista valittavana olleeseen kohteeseen nähden sekä kyselylomakkein käyttäjien mieltymyksistä. Kokeiden tuloksia käsitellään alakohdissa samassa järjestyksessä.

#### **6.3.1. Valinta-ajat**

Kursorien tehokkuuserojen vertailemiseksi tarkastellaan valinta-aikoja. Toistettujen mittauksen varianssianalyysi osoittaa, että kursorityypillä oli tilastollisesti merkitsevä vaikutus sarjan suoritusajaan ( $F_{3,51} = 143$ ,  $p < .01$ ). Hypoteesin mukaisesti kuplakursori oli konditioista nopein ja pistekursori selvästi hitain. Keskimääräiset valinta-ajat käyvät ilmi kuvasta 11.



Kuva 11. Sarjojen valinta-ajat tiheyksittäin.

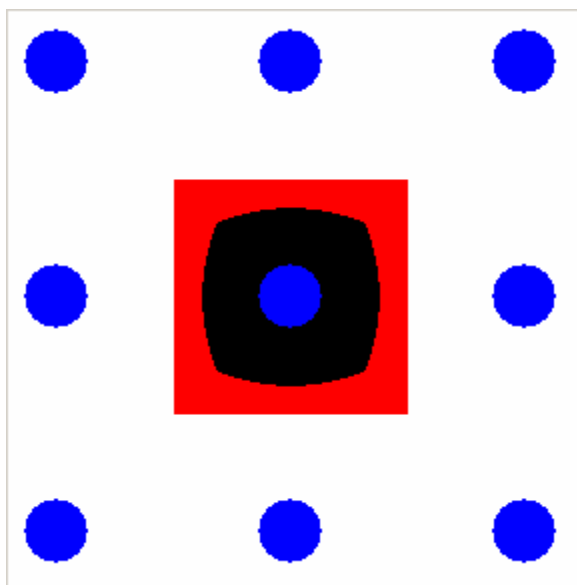
Eron saattoi olettaa johtuvan pitkälti nimenomaan siitä, että pistekursori oli varsin selvästi muita hitaampi, joten erojen merkitsevyyttä testattiin kursorien välillä myös pareittain. Myös tötterökursorin ja kuplakursorin välinen ero osoittautui tilastollisesti merkitseväksi ( $F_{1,17} = 7.5$ ,  $p < .05$ ) mutta jäi ratkaisevasti varsin pieneksi kaikilla tiheyksillä. Yllättävää tosin oli se, että laiska kuplakursori oli jopa nopeampi kuin tötterökursori. Kaikilla tiheyksillä yhteensä sarjojen suoritusajojen keskiarvot olivat kursorittain 24.20 s. (pistekursori), 19.46 s. (laiska kuplakursori), 19.60 s. (tötterökursori) ja 18.92 s. (kuplakursori). Tötterökursori oli siis kuplakursoriin nähden keskimäärin n. 3.5% hitaampi ja laiska kuplakursori jopa vielä hieman vähemmän.

Laiskan kuplakursorin ja tötterökursorin pieni ero valinta-ajoissa kuplakursoriin nähden vaikuttaa varsin kohtuulliselta, ottaen huomioon, että laiskemmalla laajenemisalgoritmeilla kohteen valinta-alue on pienempi. Onko ero jopa yllättävän pieni? Kuinka suuri ero olisi ollut todennäköinen Fittsin lain mukaan? Tämän selvittämiseksi on tarpeen selvittää ensin mikä tarkalleen ottaen ajatellaan valinta-alueen leveydeksi. Kaksiulotteisella kohdeasettelulla tämä on huomattavasti vaativampaa kuin yksiulotteisella kohdeasettelulla ja erityisesti tötterökursorin tapauksessa olisi vaikeaa löytää ongelmaton tulkinta valinta-alueesta, koska alue on harvoin symmetrinen muodoltaan. Tötterökursorin vertailu ei kuitenkaan ole tarpeen, koska se ei menestynyt valinta-ajoissa lainkaan laiskaa kuplakursoria paremmin. Koska kohteet oli aseteltu tasaväliseen ruudukkoon, laiskalle kuplakursorille ja kuplakursorille on mahdollista selvittää arviot valinta-alueiden leveyksistä tarkkuudella, joka mahdollistaa ainakin suuntaa-antavien laskelmien suorittamisen.

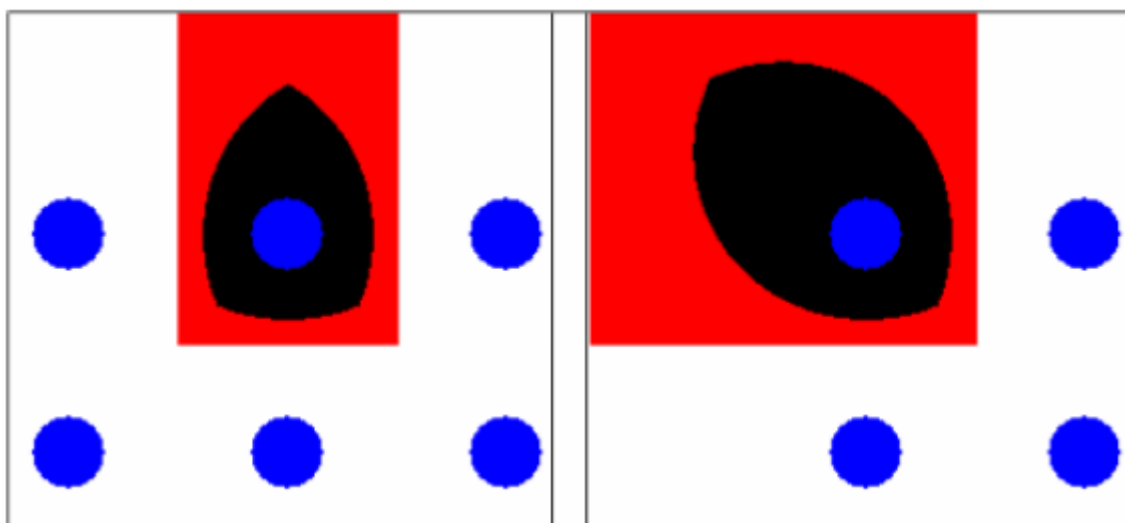
Kuplakursorin neliönmuotoisen valinta-alueen (kuvan 12 punainen alue) leveydeksi tulkitaan neliön leveys (aina kohdevälin verran tasavälisen kohteiden tapauksessa) niin kutsutun  $ID_{\min}$  mallin [MacKenzie, 1992] mukaan. Myös laiskan kuplakursorin valinta-alueen kohdalla (musta alue kuvassa 12) tukeudutaan samaan malliin ja valinta-alueen leveydeksi tulkitaan sen pienin halkaisija (leveys tai korkeus vaaka- tai pystysuorassa, aina  $2/3$  kuplakursorin valinta-alueen leveydestä vastaavalla tiheydellä). Tämä tulkinta on ongelmallinen, koska valinta-alue ei kuplakursorin valinta-alueen tapaan ole neliön muotoinen, eikä tulkinta näin ole täysin johdonmukainen tulkinnan kanssa, joka tehtiin kuplakursorin valinta-alueen leveydestä.

Myös  $ID_{\min}$  mallin käyttö on jokseenkin ongelmallista, koska malli ei ota huomioon lähestymiskulmaa, vaikka aluetta lähestyttiin kokeissa vaihtelevista suunnista. Olennaista on kuitenkin havaita, että vaikka molempien valinta-alueiden leveyksistä tehdyt tulkinnat ovat  $ID_{\min}$  mallin käytön vuoksi pessimistisiä (lähestymissuunnan huomioon ottaminen kasvattaisi kohteen leveyttä), laiskan kuplakursorin valinta-alueesta tehty tulkinta on sitä selvästi vähemmän. Koska kohteen leveyden pienentämisellä on valinnan vaikeutta ja valinta-aikaa kasvattava vaikutus, tehdyt tulkinnat pienentävät siis lopulta arviota siitä, kuinka suuri nopeusero kursorien välillä olisi ollut odotettu; arvio on konservatiivinen sen suhteen, kuinka paljon nopeampi kuplakursorin olisi pitänyt olla. Tehtyjen tulkintojen pätevyyttä tukee myös se, että vaikka lähestymiskulmaa ei kontrolloitu kokeissa eikä toteutuneita lähestymiskulmia oteta huomioon tässä laskennassa, käytännössä erot lähestymiskulmissa kursorien välillä olivat suhteellisen pieniä, koska kaikilla kursorilla suoritettiin samat valintasarjat.

Laskennassa käytetään myös oletusta, että reunimmaisten kohteiden valinnat eivät muodosta poikkeustapauksia, vaikka ne käytännössä sellaisia olivatkin. Oletuksen vaikutus tuloksiin riippuu erityisesti siitä, kuinka paljon tyhjää tilaa reunimmaisen kohteen ja reunan välillä on. Kokeissa tilaa reunimmaisen kohteen ja reunan välillä oli aina vähintään tavallisen kohdevälin verran, kuitenkin aina vähemmän kuin kaksi kohdeväliä. Kuplakursorin voisi kenties olettaa hyötynneen ylimääräisestä tilasta hieman, koska sillä kohteen valinta-alue jatkuu aina reunaan saakka, kun taas laiskalla kuplakursorilla valinta-alue pysyy tietyn pisteen jälkeen vakiona, vaikka tyhjää tilaa lisättäisiin. Kuva 13 havainnollistaa näitä poikkeustilanteita, kun valittava kohde on reunimmainen tai nurkassa oleva kohde ja tilaa kohteen ja reunan välillä on vähintään kohdevälin verran. Ilman tarkempaa selvitystä tyydytään kuitenkin vain oletamaan, että hyöty, joka poikkeustapauksista on syntynyt, on ollut verrattain pieni, hyötyi poikkeustilanteista kumpi kursori tahansa.



Kuva 12. Neliön muotoinen alue kuvaa kuplakursorin ja sen sisällä oleva pienempi alue laiskan kuplakursorin valinta-alueita keskimmaiselle kohteelle tasavälisesti aseteltujen kohteiden tapauksessa.



Kuva 13. Kuplakursorin ja laiskan kuplakursorin valinta-alueet poikkeustapauksissa, kun valittava kohde oli reunimmainen tai nurkassa ollut kohde.

Koska tutkimuksessa ei määritetty mallinnuksessa tarvittavia Fittsin lain vakioita  $a$  ja  $b$ , tukeudutaan kirjallisuudessa raportoituihin vakioiden arvoihin samankaltaisissa kokeissa. Soukoreff ja MacKenzie [2004] luettelevat Fittsin lakia käsitteleviä tutkimuksia ja muun muassa tutkimuksissa saatuja arvoja Fittsin lain vakioille  $a$  ja  $b$ . Neljässä listatuista tutkimuksista kontrollilaitteena käytettiin hiirtä ja koehenkilöt eivät olleet liikkeiltään rajoittuneita. Suorituskyky vaihteli näissä tutkimuksissa välillä 3.7-4.9 bittiä/s. Isokosken ja Raisamon tutkimuksessa testattiin kuitenkin kaikkiaan kuutta eri hiirtä ja näissä kokeissa suorituskyky oli aina välillä 4.4-4.6 bittiä/s. Käytetään arviona suorituskyvystä 4.4 bittiä/s. Vakiona  $a$  käytetään lisäksi arvoa 400 ms, mikä voi

olla tarpeettoman suuri, mutta ei ole ainakaan liian pieni. Kootusti tulkinat, joita valinta-aikojen laskennassa käytetään ovat siis  $a = 400$  ms, suorituskyky = 4.4 bittiä/s,  $W_{BC} = D$  ja  $W_{LB} = 2/3D$ , missä  $D = \{64$  pikseliä, 128 pikseliä, 160 pikseliä, 256 pikseliä $\}$ . Kohde-etäisyys  $A$  oli lisäksi aina kursorin etäisyys kohteeseen kun edellinen valinta oli suoritettu. Kohde-etäisyys käytännössä siis vaihteli kursoreittain hieman riippuen pisteestä, jossa edellisen kohteen valinnut klikkaus tapahtui. Koska kursoreilla suoritettiin samat valintasarjat, vaihtelu on kuitenkin väistämättä jäänyt verrattain pieneksi, mitä myös keskimääräiset valintojen pituudet heijastelevat;  $\text{avg}(A_{BC}) = 668.15$  pikseliä,  $\text{avg}(A_{LB}) = 669.31$  pikseliä.

Kokeissa kursoreilla suoritettujen onnistuneiden kohdevalinnat huomioon ottaen keskimääräisiksi odotusarvoiksi valinnan vaikeuksiksi saadaan edellisiä tulkintoja käyttäen kursoreittain 2.56 bittiä (kuplakursori) ja 3.05 bittiä (laiska kuplakursori), sekä valinta-ajoiksi edelleen 0.98 s. (kuplakursori) ja 1.09 s. (laiska kuplakursori). Eron voisi tämän perusteella olettaa olleen lähelle 10% toteutuneen alle 3% eron sijaan (joka kursorien välillä oli myös kun huomioon otettiin vain onnistuneet valinnat). Erityisesti kun otetaan huomioon, että laskennassa käytettyjä oletuksia voitaneen kuvailla jokseenkin pessimistisiksi ja arviota odotetusta nopeuserosta näin ollen suhteellisen varovaiseksi, eroa odotettavan ja toteutuneen nopeuseron välillä on pidettävä merkittävänä. Valinta-alueen pientymiseen nähden sekä laiskan kuplakursorin että tötterökursorin voidaan todeta olleen jopa yllättävän nopeita. Kursorit toimivat niin vähän hitaammin kuin kuplakursori, että niitä voitaneen perustella kuplakursorin vaihtoehtoina jos ne muuten havaitaan paremmiksi. Erityisesti laiskan kuplakursorin menestys on yllättävää, koska sen ei alun perin uskottu menestyvän lähellekään yhtä hyvin kuin kuplakursorin. Koska kupla myös laiskalla kuplakursorilla pumppaa edestakaisin varsin tiheään, hyvän menestyksen valinta-ajoissa voisi suoraviivaisimmin yhdistää nimenomaan kursorien pienempään kokoon, minkä toivottiinkin hyödyttävän uusia kursoreita.

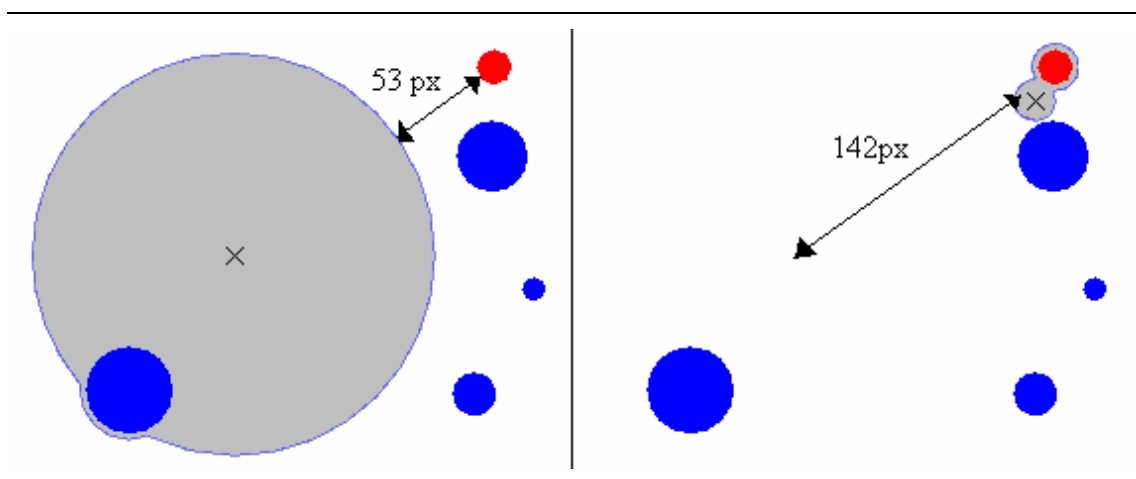
On kuitenkin yllättävää, että tötterökursorin ja laiskan kuplakursorin välillä ei ollut juurikaan eroa sarjojen suoritusajoissa. Kuplan koko pysyy molemmilla kursoreilla identtisenä, tötterökursori vaikuttaa olleen kahdesta kursorista hieman vähemmän virhealtis (mikä ei ole yllättävää) ja ero virhealttiudessa oli kaiken lisäksi varsin pieni ja osoittautui vain viitteelliseksi (katso kohta 6.3.2, virheet ja klikkausten sijainti kohteisiin nähden). Kursorien tasaväkisyys ei näin ollen selittyneen sen enempää poikkeavilla koon muutoksilla kuin pienemmällä virhemäärälläkään, tai edes sillä, että laiskaa kuplakursoria olisi käytetty nopeammin virhealttiuden kustannuksella. Vaikuttaakin siltä, että valinnan säilyttämisestä ei yksinkertaisesti ollut hyötyä tehokkuuden kannalta.

Jos laajenevasta valinta-alueesta ei saatu hyötyä, herää kysymys, olisiko tötterökursorin itse asiassa pitänyt olla hitaampi kuin laiskan kuplakursorin, koska

tötterökursori oli hännän myötä usein huomattavasti suurempi? Voisi kuvitella, että liikesuunnassa erityisen pitkäksi kasvanut kursori olisi pinta-alaltaan suurena kärsinyt suuresta koosta samalla tavalla kuin kuplakursori. Suuremman koon negatiivinen vaikutus valinnan nopeuteen olisi linjassa paitsi Grossmanin ja Balakrishnanin tutkimuksen kanssa (kuplakursorin käyttö hidastui kun kohteet olivat kaukana toisistaan ja kuplakursori kasvoi usein suureksi) myös sen havainnon kanssa, että laiska kuplakursori ja tötterökursori melko selvästi näyttivät saaneen hyötyä pienemmästä koosta kuplakursoriin nähden. Kenties tötterökursorin hieman pienempi virheprosentti selittää kursorien tasaväkisyyden?

Kun verrataan vain virheetömiä sarjoja, kursorien välillä on kuitenkin entistäkin pienempi ja myös entistä selvemmin merkityksetön ero. Kaikki havainnot viittaisivat siis siihen, että valinnan säilyttämisestä ei näytä olleen valinta-aikojen suhteen myöskään haittaa. Tämä on jokseenkin ristiriitaista. Kuplakursori näytti kärsivän jokseenkin selvästi suuresta koostaan ei-niin-pieniin tötterökursoriin ja laiskaan kuplakursoriin nähden, mutta tötterökursori ei kuitenkaan kärsi suuresta pinta-alasta laiskaan kuplakursoriin nähden. Vaikka on mahdollista, että laiskalla kuplakursorilla on pitänyt esimerkiksi useammin suorittaa korjaavia liikkeitä takaisinpäin kun kohde on ensin ohitettu, mikä voisi selittää kursorien tasaväkisyyden, voidaan hyvällä syyllä pohtia, pitääkö oletus kursorin suuremman koon negatiivisesta vaikutuksesta valinta-aikoihin paikkansa.

Grossmanin ja Balakrishnanin oletus oli, että kursorin suuri koko hidasti kursoria pienellä tiheydellä, koska suuri koko häiritsi käyttäjiä [Grossman & Balakrishnan, 2005a]. He eivät kuitenkaan verifioineet tätä, eikä sitä tehty tässäkään tutkimuksessa, vaan se oletettiin päteväksi oletukseksi. Saattaa kuitenkin olla, että kursorin suuren koon häiritsevyyttä ongelmallisempaa on itse asiassa se, kuinka hyvin kursorin ja kohteen välisen etäisyyden perusteella voidaan arvioida jäljellä olevaa siirtomatkaa. Kun kuplakursori on pieni, silmämääräinen havainnointi korkeintaan yliarvioi tarvittavan siirron pituuden jos kursori laajenee matkalla, mutta kun kupla on suuri, tarvittava siirto voi olla huomattavasti arvioitua pidempi, jos kuplakursori kutistuu matkalla. Jälkimmäistä tapausta havainnollistetaan kuvan 14 esimerkillä. Alkutilanteessa oikeassa yläreunassa sijaitsevan kohteen valinta näyttää edellyttävän vain suhteellisen lyhyen siirron. Kun kursoria siirtää lähemmäs kohdetta kupla laajenee, mutta ennen maalikohtetta osuu kohteeseen, joka on kuvassa maalikohteen alapuolella. Tässä vaiheessa kupla alkaa kutistua ja kutistuu siihen saakka, kunnes kohdistin ohittaa maalikohtetta edeltävän kohteen, jolloin kupla alkaa laajentua uudelleen. Lopulta alkuperäinen maalikohde on kursoria lähinnä oleva kohde ja kuplan säde ulottuu kohteeseen (kuvan lopputilanne). Valinta vaati kuitenkin huomattavasti pidemmän siirron kuin silmämääräisesti olisi saattanut arvioida.



Kuva 14. Vasemmalla puolella on alkutilanne, kun kuplakursorilla ollaan valitsemassa kohdetta oikeassa ylänurkassa. Oikealla puolella tilanne, jossa kupla lopulta ulottuu kohteeseen.

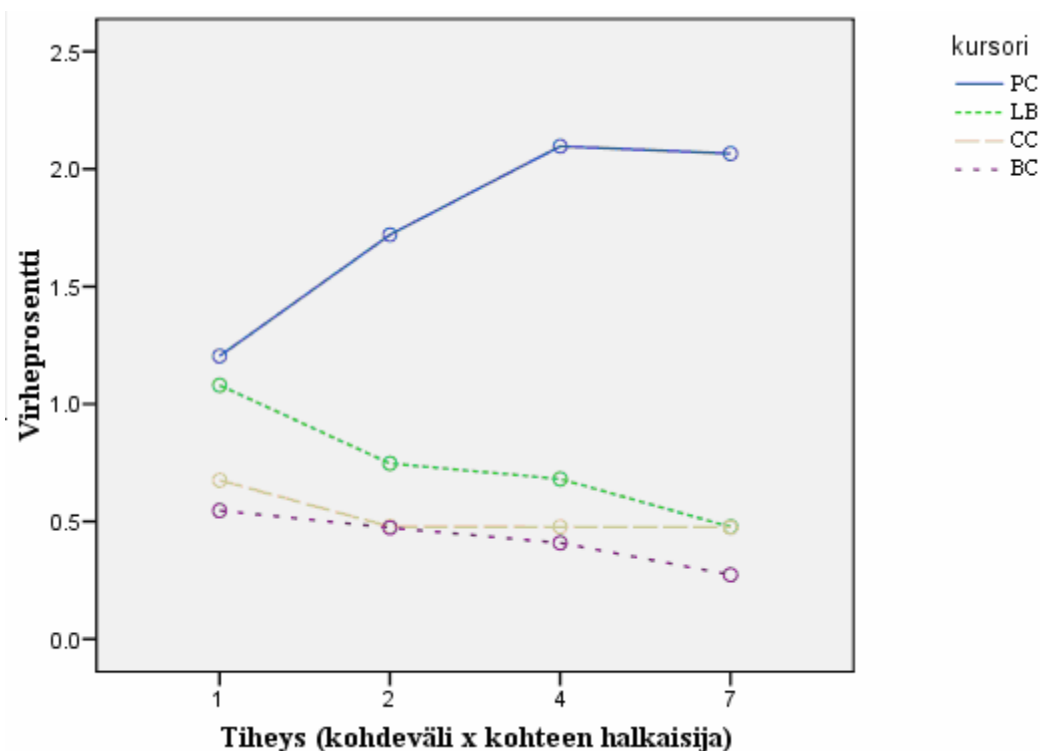
Kuplakursorin kohdalla esitetyn kaltaisia tilanteita voi tapahtua varsin usein. Vaikka tämä vaatii vahvistusta, havainto on mielenkiintoinen. Se tarjoonee Grossmanin ja Balakrishnanin tulkintaa perustellumman selityksen sille, miksi kuplakursori hidastui odotusten vastaisesti pienellä kohdettiheydellä. Se selittäisi vakuuttavasti myös miksi tötterökursorin suurempi pinta-ala ei näyttänyt vaikuttavan negatiivisesti sen valinta-aikoihin laiskaan kuplakursoriin verrattuna (tötterökursorilla ja laiskalla kuplakursorilla kuplan etäisyys valittavaan kohteeseen on aina identtinen). Ehdotettu selitys selittäisi myös miksi nämä kursorit näyttivät menestyvän jopa odotettua paremmin valinta-ajoissa kuplakursoriin nähden. Kupla on molemmilla uusilla kursoreilla pienempi kuin kuplakursorilla eikä kupla voi olla *lähellä* jotakin kohdetta, ellei kohde todella ole myös lähellä tulla valituksi. Kuplan ja kohteen etäisyyden perusteella voidaan näistä syistä uusilla kursoreilla tehdä luotettavampia arvioita tarvittavan siirron pituudesta.

### 6.3.2. Virheet ja klikkausten sijainti kohteisiin nähden

Virheelliset sarjat hyväksyttiin sellaisenaan, mutta virhemääristä pidettiin lukua. Klikkauksen sijainniksi tulkittiin kohta, jossa hiiri vapautettiin. Onnistunut valinta edellytti, että hiiren painike sekä painettiin alas että vapautettiin kun oikea kohde oli valittuna. Muulloin klikkaus tulkittiin virheeksi. Virheiksi ei kuitenkaan laskettu tuplakkauksia (yhteensä 17 kappaletta), joiden nopeasti onnistunutta klikkausta seuranneina ei haluttu tulkita vaikuttaneen kursorin virhealttiuteen, eikä sen enempää valinta-aikoihinkaan.

Yleisesti ottaen virheitä tapahtui huomattavan vähän, pistekursorilla kuitenkin huomattavasti muita kursoreita enemmän, 1.77% klikkauksista, kun muilla kursoreilla virheprosentti oli keskimäärin 0.57%. Pareittain kolmen aluekursorin välillä suoritettujen toistettujen mittausten varianssianalyysit eivät osoittaneet eroa virhekeskiarvoissa tilastollisesti merkitseväksi tötterökursorin ja kuplakursorin välillä. Testit osoittivat

kuitenkin, että laiska kuplakursori oli mahdollisesti ( $F_{1,17} = 4, p < .1$ ) hieman tötterökursoria, sekä selvästi ( $F_{1,17} = 9, p < .01$ ) kuplakursoria virhealttiimpi, vaikka myös laiskan kuplakursorin virheprosentti oli matala, suurinta tiheyttä lukuun ottamatta aina alle prosentin. Kuva 15 havainnollistaa kursorien virheprosentteja.



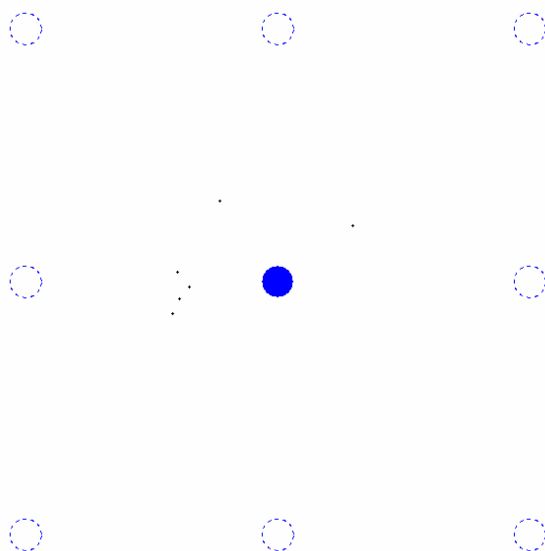
Kuva 15. Kursorien virheprosentit tiheyksittäin.

Virhealttius näyttääkin olevan asia, missä valinnan säilyttämisestä on hieman hyötyä. On käytännöllistä, että kursori ei menetä valintaa kohteeseen välittömästi sen jälkeen kun kupla ei enää ulotu kohteeseen, vaan valinta on yhä suoritettavissa onnistuneesti.

Tötterökursori oli kuitenkin kokeen kolmesta aluekursorista ainoa, jolla virheiden määrä ei laskenut enää toiseksi suurimman tiheyden jälkeen, vaan virheiden määrä pysyi suurin piirtein samana pienemmilläkin tiheyksillä. Subjektiiivinen havainto koetilanteista oli, että tötterökursoria käyttäen hiiren liikuttaminen vaikutti olleen erityisesti pienimmällä tiheydellä usein varovaista ja epävakaa kun kursori kasvoi pitkäksi. Kursorin epävakait liikkeet tuntuivat kulminoituvan erityisesti virheisiin, jotka johtuivat juuri siitä, että käyttäjä klikkasi hitaalla nopeudellakin liian aikaisin, juuri ennen kohteeseen osumista. Tämä havainto näyttää datan perusteella pitävän paikkansa. Kuvassa 16 näkyvät tötterökursorilla pienimmällä tiheydellä syntyneet virheet tasapainotettuna lähestymiskulman mukaan niin, että kuvassa klikkauskohdat näkyvät siten, kuin keskimäinen kohde olisi yritetty valita vasemmalta oikealle suuntautuneella liikkeellä. Katkoviivoin piirretyt kohteet ovat kuvassa näkyvissä vain mittakaavan hahmottamiseksi ja keskimäisen kohteen



ympärillä olevat kohteet ovatkin saattaneet olla missä tahansa suunnassa todelliseen lähestymissuuntaan nähden. Tästä syystä valinta-alueettakaan ei voida piirtää kuvaan, koska se on aina vaihdellut klikkauksittain todellisesta lähestymiskulmasta riippuen. Kuten kuvasta havaitaan, virheitä oli erittäin vähän (6 kpl) ja lisäksi pienimmällä tiheydellä virheet johtuivat muillakin kursoreilla pääsääntöisesti liian lyhyistä siirroista. Tästä huolimatta se, että suuri osa virheistä (4/6) on keskittynyt varsin pienelle sektorille lähelle optimaalista lähestymissuuntaa (kuva 16) tukee subjektiivista havaintoa siitä, että erityisen pitkäksi kasvaessaan kursoria on vaikea liikuttaa ja kursoria liikutetaan varovaisesti, minkä voisi hyvinkin ymmärtää selittävän virhevalintojen keskittymän pienellä sektorilla lähestymissuunnassa hieman ennen kohdetta. Lisäksi edellä siis havaittiin, että laiska kuplakursori oli hieman muita kokeen aluekursoreita virhealttiimpi. Näistä havainnoista huolimatta virhealttiutta on vaikea käyttää vahvana argumenttina minkään aluekursorin puolesta tai vastaan, koska virhemäärät olivat niillä kaikilla varsin alhaisia.



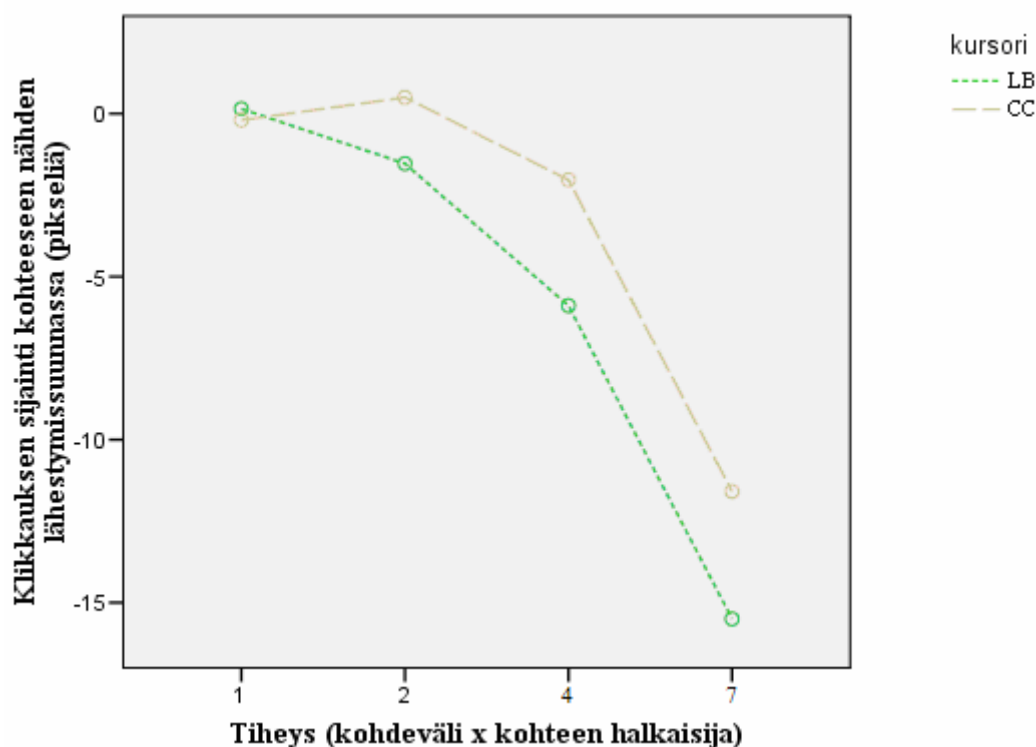

---

Kuva 16. Virheklikkaukset tötterökursorilla pienimmällä kohdettiheydellä.

Testisovellus tulkitsi virheellisesti myös viisi virheklikkausta tuplaklikkauksiksi. Nämä virheet korjattiin niiltä osin, että virheprosenttien osalta esitetyt luvut ovat tarkkoja, mutta osumakartoista nämä klikkaukset puuttuvat, koska niistä ei kokeita suoritettaessa tallentunut kaikki olennainen tieto, joka tallennettiin automaattisesti virheklikkauksina rekisteröidyistä klikkauksista. Kun puuttuvien virheklikkausten määrä oli näin pieni, puuttuvaa dataa ei nähty tarpeelliseksi etsiä lokitiedostoista, joihin kaikki kokeiden data erikseen tallennettiin (ilman minkäänlaista esikäsittelyä).

Kun tarkastellaan valintojen keskimääräisiä virheitä liikesuunnassa (kohteen keskipisteeseen nähden), kun ensimmäinen klikkaus johti kohteen onnistuneeseen valintaan, kohde valittiin aluekursoreilla odotetusti keskimäärin ennen kuin kohdistin

oli ohittanut kohteen keskustan ja kuplakursorin suuri valinta-alue mahdollisti klikkaamisen selvästi aikaisemmin kuin kahdella muulla aluekursorilla. Kuplakursorin ero näyttäisi olevan varsin selvä ja vastaa oletusta siitä, kuinka uudet cursorit kärsivät siihen nähden pienemmästä valinta-alueesta. Lähestyttäessä kohdetta tötterökursorilla ja laiskalla kuplakursorilla kumpikaan ei osu kohteeseen ennen toista ja mahdolliset erot olisivat mielenkiintoisia. Tötterökursorilla keskimääräinen siirron pituus (klikkauskohtan etäisyys lähtöpisteestä) oli noin 4.5 pikseliä pidempi kuin laiskalla kuplakursorilla, joskin tötterökursorilla myös lähtöpiste oli keskimäärin noin 2 pikseliä kauempana kohteesta, mikä pienentää eroa liikkeen pituudessa kohteen keskusta nähden. Tötterökursorilla valinta suoritettiin silti, suurinta tiheyttä lukuun ottamatta, keskimääräisesti 2-4 pikseliä myöhemmin kuin laiskalla kuplakursorilla (kuva 17). Ero oli tilastollisesti merkitsevä ( $F_{1,17} = 7$ ,  $p < .05$ ). Esimerkiksi pienimmällä tiheydellä tötterökursorilla klikkaus tapahtui keskimäärin 11.59 ja laiskalla kuplakursorilla 15.49 pikseliä (kuplakursorilla keskimäärin 21.60 pikseliä) ennen kohdetta.



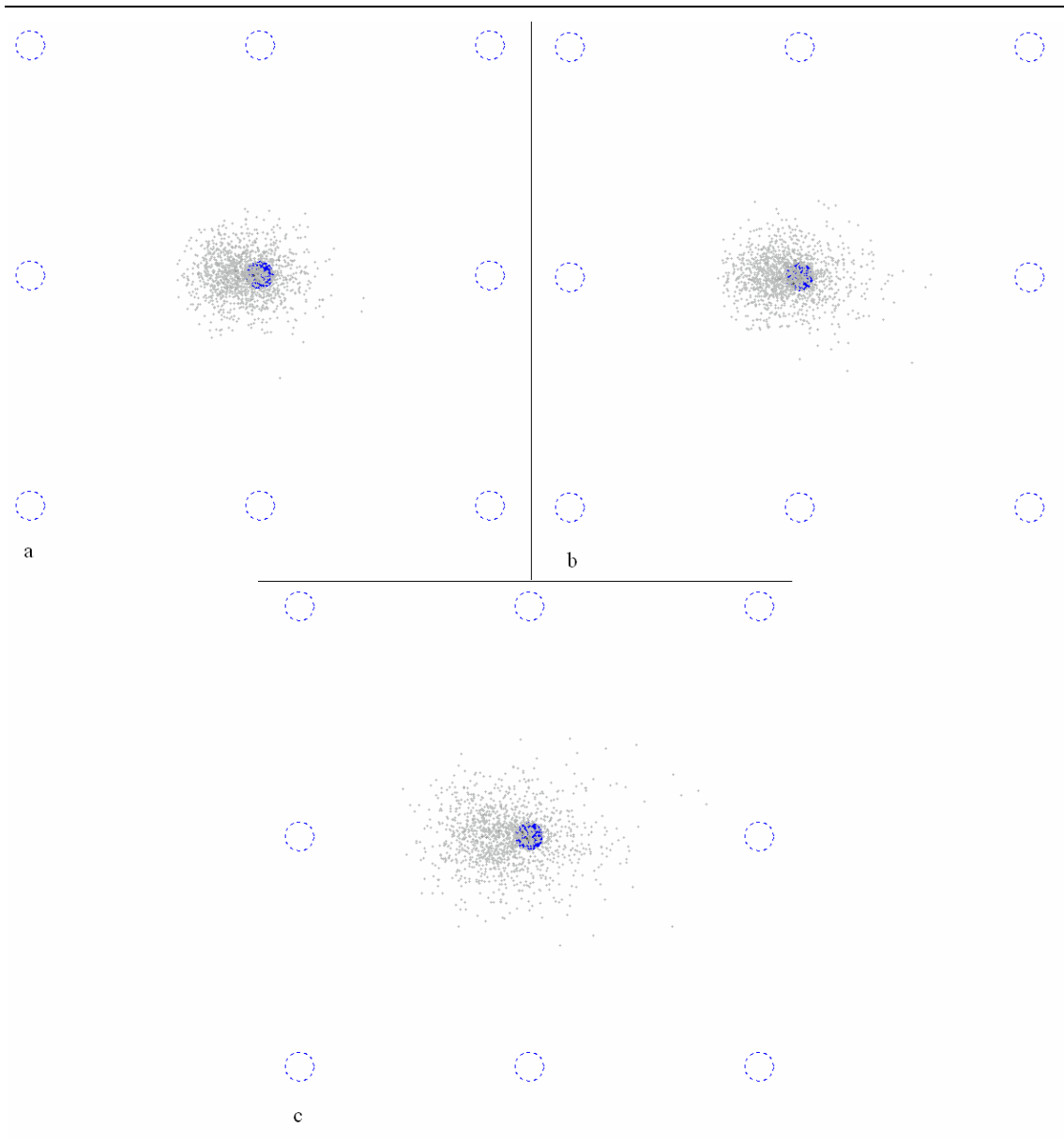
Kuva 17. Klikkauksen keskimääräinen sijainti kohteeseen nähden lähestymissuunnassa tiheyksittäin laiskalla kuplakursorilla ja tötterökursorilla.

Ero voi mahdollisesti selittyä sillä, että tötterökursorilla kohde toisinaan valittiin vasta alkuperäisen valinta-alueen jälkeen tai sillä, että laiskalla kuplakursorilla valinta vaati useammin korjaavia liikkeitä takaisinpäin kun kohde ensin ohitettiin. Voi kuitenkin olla, että valinnan säilyttäminen on antanut mahdollisuuden olla välittämättä niin paljoa cursorin pysäyttämistä ja kursoria on tästä syystä liikutettu hieman

optimistisemmin ja pidempiä liikkeitä tapahtui enemmän. Jos satunnaiset klikkaukset tötterö-kursorilla alkuperäisen valinta-alueen ulkopuolelle selittäisivät eron, eron pitäisi olla mediaanitapauksessa paljon pienempi. Ero mediaanitapauksissa on kuitenkin vain hieman pienempi, suurinta tiheyttä lukuun ottamatta suuruusluokkaa 2-3 pikseliä. Tämän voisi tulkita viitteenä siitä, että tötterökursoria on mahdollisesti todellakin siirretty hieman laiskaa kuplakursoria optimistisemmin. Ero hiiren liikkeiden pituuksissa on kuitenkin pieni ja olisi ollut yllättävää, jos ero olisi näkynyt positiivisena vaikutuksena tötterökursorin valinta-ajoissa.

Erot klikkausten jakaumissa ovat parhaiten näkyvissä pienimmällä tiheydellä (osumakartat kuvassa 18). Tötterökursorilla (18 (b)) klikkauksia on hieman enemmän kohteen takana (keskimmäiseen kohteeseen nähden oikealla) ja osa on selvästi kauempana kohteesta kuin laiskalla kuplakursorilla (18 (a)). Huomattavaa on kuitenkin, että jakaumista ei löydy kovin suuria eroja ja erityisen pitkiä siirtoja tötterökursorilla on varsin vähän. Laajenevasta valinta-alueesta ei näin ollen siis osumakartankaan mukaan ollut paljon hyötyä. Kuplakursorilla (18 (c)) klikkaukset jakaantuvat silminnähten laajemmalle alueelle ja laajaa valinta-aluetta on selvästi kyetty hyödyntämään tehokkaammin.

Keskimääräinen poikkeama liikkeen suunnassa oli odotetusti aina suurempi kuin liikkeen suuntaan nähden poikittaisessa suunnassa. Kaikilla tiheyksillä yhteensä poikkeama oli liikkeen suunnassa keskimäärin pistekursorilla 1.21, laiskalla kuplakursorilla ja tötterökursorilla 1.36 sekä kuplakursorilla 1.48 kertaa suurempi kuin liikkeen suuntaan nähden poikittaisessa suunnassa. Tötterökursorin ja laiskan kuplakursorin välillä ei myöskään ollut suurta eroa keskimääräisten poikkeamien suuruudessa (keskimäärin liikkeen suunnassa tötterökursorilla 15.59 ja laiskalla kuplakursorilla 15.94 pikseliä, liikkeen suuntaan nähden poikittaisessa suunnassa tötterökursorilla 11.46 ja laiskalla kuplakursorilla 11.69 pikseliä). Koska erot vaikuttaisivat olevan häviävän pieniä, niiden tilastollista merkitsevyyttä ei selvitetä. Havainto samansuuruisista poikkeamista liikkeen suunnassa ja liikkeen suuntaan nähden poikittaisessa suunnassa tukevat kuitenkin edellisiä havaintoja siitä, että valinnan säilyttäminen ei juurikaan vaikuta valintaan.



Kuva 18. Osumat pienimmällä tiheydellä laiskalla kuplakursorilla (a), tötterökursorilla (b) ja kuplakursorilla (c).

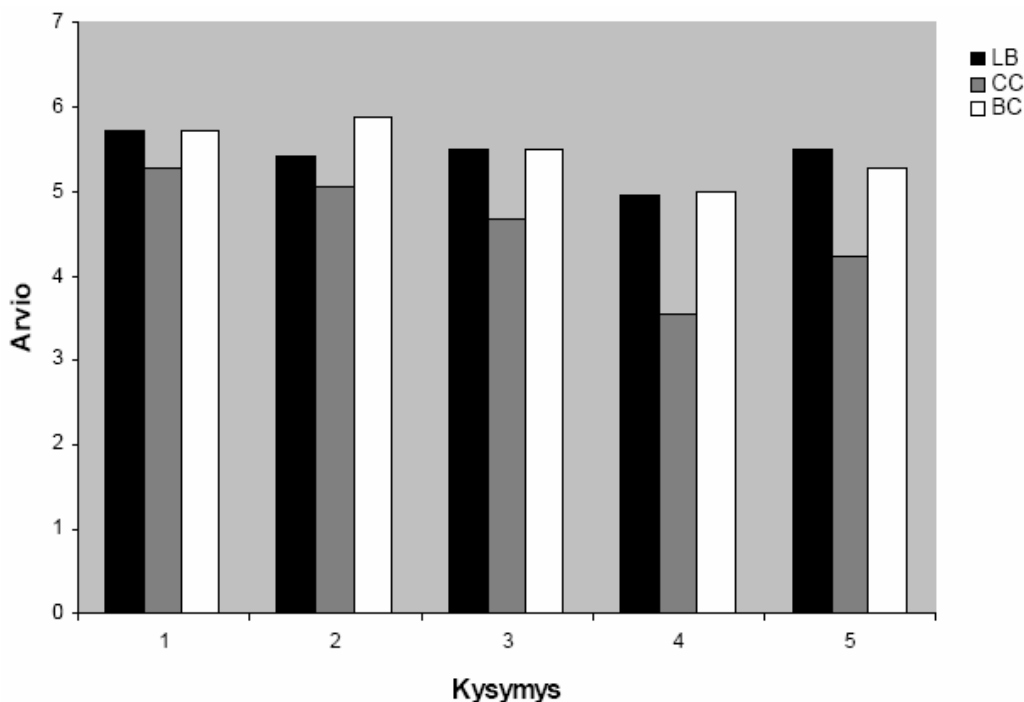
Keskenään johdonmukaiset havainnot näyttäisivätkin viittaavan vahvasti siihen, että hyöty valinta-alueen laajenemisesta jää tötterökursorilla käytännössä kokonaan realisoitumatta. Ainoat selvät erot laiskan kuplakursorin ja tötterökursorin välillä näyttäisivät lopulta olevan, että tötterökursorilla kohteen valinta tapahtuu tyypillisesti hieman kauempana lähtöpisteestä ja virheitä tapahtuu hieman vähemmän. Näitä kahta kursoria lieneekin järkevintä arvioida ja vertailla keskenään ennen kaikkea niiden kvalitatiivisten ominaisuuksien ja koehenkilöiden tuntemusten perusteella.

Kuplakursoria vastaan argumentteja ei löydetty. Hieman pienempi virheprosentti ja klikkausten painottuminen selvästi lähemmäs lähtöpistettä kuin muilla kursorilla sen sijaan puhuvat kuplakursorin puolesta.

### 6.3.3. Käyttömukavuus

Kyselylomakkein kerätty data käyttäjien mieltymyksistä viimeistään osoittaa, että tötterökursori ei ole varteenotettava kilpailija kuplakursorille. Tötterökursori sai huonoimman arvion kaikissa viidessä kyselylomakkeen kysymyksessä (kuva 19). Aineistolle suoritettujen Wilcoxon signed ranks testien tulokset osoittivat, että erot arvioissa olivat tilastollisesti merkitseviä kysymysten 1-2 ( $p < .05$ ) sekä 4-5 ( $p < .01$ ) kohdalla verrattaessa tötterökursoria kuplakursoriin sekä kysymysten 1 ( $p < .05$ ) ja 3-5 ( $p < .01$ ) kohdalla kun tötterökursoria verrattiin laiskaan kuplakursoriin. Kysymyksessä kursorin nopeudesta ei ollut tilastollisesti merkitseviä eroja laiskan kuplakursorin ja tötterökursorin välillä, mikä vastaa pitkälti valinta-ajoista kerättyä dataa. Toinen kysymys, jossa tilastollisesti merkitsevää eroa ei ollut, oli kysymys kursorin käyttäytymisen loogisuudesta, kun tötterökursoria verrattiin kuplakursoriin.

Suurimmat erot tuntemuksissa muihin kursoreihin nähden syntyivät kysymyksiä 4 ja 5 kohdalla. Kursoria ei siis koehenkilöiden mukaan yksinkertaisesti ollut miellyttävä käyttää. Ottaen huomioon erityisesti huono menestys kysymyksen 4 kohdalla (”kuinka häiritsevältä kursorin koko ja muoto sekä niiden muutokset tuntuivat”), käyttömukavuus on todennäköisesti kärsinyt juuri valinnan säilyttämisestä, mikä selvimmin on vaikuttanut kursorin kokoon ja muodon muutoksiin. Tätä tukee se, että kuplan koko ja muoto sekä näiden muutokset, jotka ovat täsmälleen samanlaisia myös laiskalla kuplakursorilla, eivät laiskan kuplakursorin kohdalla näy vaikuttaneen negatiivisesti arviointiin, sillä käyttäjien mieltymyksissä ei kysymyksen 4 kohdalla ollut tilastollisesti merkitsevää eroa laiskan kuplakursorin ja kuplakursorin välillä.



Kuva 19. Keskimääräiset arviot kursorista kysymyksittäin.

Tötterökursori sai myös selvästi kokeen kahta muuta aluekursoria ja jopa hieman pistekursoria enemmän sijoituspisteitä kun käyttäjiä pyydettiin sijoittamaan kursorit paremmuusjärjestykseen. Tämä vahvistaa jo edellisen perusteella selvältä vaikuttanutta havaintoa, että käyttäjät eivät pitäneet kursorista.

Vapaamuotoinen palaute tötterökursorin käytön herättämistä tuntemuksista vastaa pitkälti tilastollisesti kerättyä aineistoa, sillä valinnan säilyttäminen oli selvästi koettu vastenmieliseksi. Palautteessa kuvataan värikkäästi, kuinka valinnan säilyessä kohteeseen kursori ”tuntui tahmealta”, ”tarttui” kohteisiin tai ”ei päästännyt irti”. Muutamat totesivat spesifisesti, että kursori oli heidän mielestään häiritsevä tai vaikeampi hahmottaa kun valinnan säilyttävä osa harvalla kohdevälillä kasvoi pitkäksi. Yleisesti ottaen voisi sanoa, että palautteen mukaan ”kursorin käyttö herätti ärtymystä”, kuten eräs selvästi yleistä kantaa edustanut kommentti kuului.

Havaintojen perusteella näyttääkin olevan kiistatonta, että käyttäjät eivät pidä tötterökursorista ja tämä johtuu suureksi osaksi nimenomaan valinnan säilyttämisestä, koska kursori hävisi nopeudessa suhteellisen vähän, eikä virhealttiudessaakaan ollut suurta eroa. Tässä valossa onkin mielenkiintoista nähdä kuinka hyvin laiska kuplakursori menestyi. Sitä ei siis ainakaan koon ja muodon sekä näiden muutosten osalta arvioitu kuplakursoria huonommaksi, mikä vaikutti olleen ratkaiseva kysymys tötterökursorin kohdalla.

Eroja käyttötuntemuksissa laiskan kuplakursorin ja kuplakursorin välillä ei syntynyt muidenkaan kysymysten kohdalla. Kuvassa 19 on havaittavissa pieniä, tasaisesti molempia kursoreita suosivia eroja kysymyksittäin, mutta Wilcoxon signed ranks testit eivät osoita, että erot olisivat tilastollisesti merkitseviä. Myös paremmuusjärjestystä arvioitaessa kursorit menestyivät varsin tasaväkisesti kun puolet koehenkilöistä pitivät laiskaa kuplakursoria alkuperäistä kuplakursoria parempana ja yksi koehenkilö sanoi, ettei kyennyt erottamaan kursorien välistä eroa ja arvioi ne yhtä hyväksi. Mieltymyksissä näyttää kuitenkin olleen varsin selvä sukupuolten välinen ero. Kaikissa viidessä kyselylomakkeen kysymyksessä miespuolisten henkilöiden keskimääräiset arviot puolsivat kuplakursoria ja vastaavasti naispuolisten koehenkilöiden kohdalla keskimäärin paremmaksi arvioitiin laiska kuplakursori. Yksittäisten kysymysten kohdalla Wilcoxon signed ranks testit osoittavat sukupuolten välisen eron tilastollisesti merkitseväksi lopulta vain kysymyksen 5 kohdalla ( $p < .01$ ). Kun kysymyksessä kuitenkin kysyttiin kuinka miellyttävältä kursorin käyttö tuntui kaikkiaan, lienee perusteltua todeta, että ero mieltymyksissä sukupuolten välillä on tilastollisesti merkitsevä. Sukupuolten välinen ero esiintyi vastauksissa myös paremmuusjärjestystä arvioitaessa. Yhdeksästä naispuolisesta koehenkilöstä vain yksi arvioi kuplakursorin paremmaksi kuin laiskan kuplakursorin (yksi arvioi kursorit lisäksi yhtä hyväksi) kun miespuolisista henkilöistä vain kaksi piti laiskaa kuplakursoria alkuperäistä kuplakursoria parempana. Ottaen huomioon, että vertailut suoritettiin *post*

*hoc*, eroa voidaan tässä kohtaa pitää vain viitteellisenä, vaikka se osoittautuikin tilastollisesti merkitseväksi ( $p < .05$ ).

Hieman yllättävää oli, että vaikka pistekursori oli testatuista kursoreista selvästi hitain ja virhealttein, se ei kuitenkaan sijoittunut viimeiseksi paremmuusjärjestystä arvioitaessa. Vain kolmasosa (6/18) koehenkilöistä piti sitä huonoimpana kursorina ja yli kolmasosa (7/18) joko parhaana (2 koehenkilöä) tai toiseksi parhaana (5 koehenkilöä). Aluekursoreiden selvästi suurempi nopeus ja pienempi virhealttius huomioon ottaen tämän voisi olettaa johtuvan koehenkilöiden tottuneisuudesta pistekursorin käyttöön. Muutamit kommentit heijastelivatkin tätä, kun osa koehenkilöistä oli vastaustensa mukaan kokenut tottuneensa kursoriin tai kursoreihin ja alkuun omituiselta tai häiritsevältä tuntunut kursori oli loppua kohden alkanut tuntua miellyttävämmältä. Koska paremmuus on kuitenkin jokseenkin subjektiivinen käsite, pistekursoria voi pienenä ja muuttumattoman kokoisena perustella vakuuttavasti myös esteettisillä kursorin kokoon ja muotoon sekä näiden muutoksiin liittyvillä seikoilla. Osa ihmisistä saattaa pitää kooltaan jatkuvasti muuttuvaa kursoria liian vastenmielisinä pitääkseen sitä yhtään pistekursoria parempana.

## **7. Pohdinta**

Pohdinta on jaettu neljään osaan siten, että ensin, kohdassa 7.1, käsittelen realistista työpöytäympäristöä yksinkertaistaneen koeasetelman vaikutusta tuloksiin. Tämän jälkeen, kohdassa 7.2, kerron tekijöistä, joilla saattoi olla vaikutusta uusien kursorien toimivuudesta saatuihin tuloksiin. Kohdassa 7.3 palaan kohde-etäisyyden arvioimisen vaikeudesta tehtyyn havaintoon ja pohdin, kuinka ongelman voisi ottaa huomioon. Tämän jälkeen, kohdassa 7.4, kerron vielä, kuinka tavoitteena oli alunperin kehittää tötterökursorista muunnelma, jonka olisi voinut toivoa nopeuttavan kohdevalintaa kuplakursoriinkin nähden. Esiteltyäni ensin suunnittelun lähtökohdat, käsittelen lyhyesti myös ongelmia, joiden vuoksi kursori ei lopulta ollut täysin toimiva.

### **7.1. Koeasetelmasta**

On epäilemättä mahdollista, että kokeissa käytetty koeasetelma tasavälisine kohteineen ei yksinkertaisuudessaan kuvannut riittävän tarkasti realistista työpöytäympäristöä mahdollistaakseen kursorien luotettavan vertailun. Koeasetelmassa ei esiintynyt sen enempää kursorien kannalta erityisen negatiivisia kuin positiivisiakaan tapauksia, joita syntyi kun kohdevälit vaihtelisivat. Uusille kursoreille erityisen vaikeita tapauksia olisivat olleet tilanteet, joissa nimenomaan kohteen lähellä on muita kohteita ja maalin alkuperäinen valinta-alue on kuplakursoriin nähden huomattavasti pienempi. Jos myös kohteen takana on kohde, tötterökursori kärsisi lisää, koska hyöty kasvavasta valinta-alueesta jäisi pieneksi. Toisaalta valinta-alue kasvaisi joillakin kohdeasetteluilla tötterökursoria hyödyttävällä tavalla siten, että tilanne voisi olla huomattavasti vaikeampi kuplakursorille. Tällaisia tilanteita olisivat erityisesti tilanteet, joissa valittavan kohteen takana on paljon tilaa. Kuplakursorin kannalta ongelmallisia tilanteita olisivat olleet myös tilanteet, joissa kohdetiheys olisi ollut joko erittäin pieni tai kohteet olisivat muuten sijainneet näytöllä niin, että kursorin koonmuutokset ja osumat kohteisiin, joita matkalla kohteeseen syntyy, olisivat olleet erityisen häiritseviä tai yllättäviä. Tasaväliseen ruudukkoon sijoitettujen kohteiden tapauksessa koonmuutokset ja kohdevalinnat olivat säännöllisempiä ja ennustettavampia.

Enemmän yksinkertaisesta koeasetelmasta hyötyivät todennäköisesti kuitenkin nimenomaan uudet kursorit, joille olisi ollut ongelmallista, jos maalin lähellä olisi usein ollut muita kohteita. Koska kumpikaan uusista kursoreista ei osoittautunut kuplakursoria nopeammaksi, eikä tulosten peruseella pyritä suosittelemaan kumpaakaan sellaisenaan, yksinkertaisella koeasetelmalla ei lopulta kuitenkaan ollut vaikutusta tutkimuksen olennaisimpien tulosten suhteen.



## 7.2. Uusien kursorien optimoinnista

Sekä laiskan kuplakursorin että tötterökursorin toteutus epäilemättä olisi vielä optimoitavissa sekä valinta-aikojen että mahdollisesti myös miellyttävyyden osalta. Ensinnäkin raja-arvo, joka määrää kuinka suuri lähimmän ja toiseksi lähimmän kohteen etäisyyksien suhteen on oltava, että kupla ulottuu kohteeseen, määriteltiin tämän tutkimuksen puitteissa vain sopivasti kuplakursorista erottuvaksi ja voi olla vielä huomattavasti optimoitavissa. Toisekseen, ainoa vaatimus laiskan kuplakursorin säteen laskevalle algoritmille oli, että kuplan säde ulottuu kohteeseen ja kohde tulee valituksi vasta kun lähin kohde on sopivalla suhteella lähempänä kuin toiseksi lähin kohde. Avoimeksi jätettiin, kuinka kuplan säteen pitäisi määräytyä rajan ulkopuolella tai kun suhde kasvaa vielä valinnan jälkeenkin. Tässä tutkimuksessa käytetty toteutus algoritmista toimi niin, että kuplan koko lähtökohtaisesti pieneni suhteen kasvaessa valinnan jälkeen, koska tavoitteena oli pitää kursorin koko mahdollisimman pienenä, minkä uskottiin olevan eduksi pyrittäessä kompensoimaan pienempää valinta-aluetta. Toisin toteutettuna säteen olisi voitu myös antaa jatkaa laajenemista niin pitkään kuin lähestyttävä kohde on entistä selvemmin lähin kohde. Tästä olisi saattanut ollut hyötyä monellakin tavalla. Kursorin koko ei esimerkiksi olisi laajentunut ja kutistunut niin usein edestakaisin. Laajeneminen ja kutistuminen olisivat olleet myös informatiivisempia ja kursorin käytön miellyttävyys olisi voinut parantua. Samalla kursori olisi tosin luonnollisesti myös kasvanut suuremmaksi mikä vaikuttaisi olevan ongelmallista joka tapauksessa.

Tötterökursorin oletettiin hyötyvän valinnan säilyttämisestä sen vuoksi, että valinnan säilyttämisen myötä kohteeseen osuminen johtaa liikesuuntaan laajenneeseen valinta-alueeseen. Liikesuuntaan laajeneva valinta-alue puolestaan sallii ennen kaikkea optimistisemmat, nopeammat hiiren liikkeet. Informaalit kokeilut ennen kokeita antoivat syytä olettaa, että näin kursoria olikin mahdollista käyttää huomattavasti laiskaa kuplakursoria nopeammin (ei kuitenkaan kuplakursoria nopeammin). Kokeilujen perusteella voitiin kuitenkin myös ennustaa optimistisempien hiiren liikkeiden kasvattavan virheprosenttia jokseenkin huomattavasti. Käyttäjien saama ohjeistus testien suorittamiseksi ”mahdollisimman nopeasti ja tarkasti” ja tässä nimenomaan huomautus tarkkuudesta ei näin ollen varmastikaan kannustanut erityisen optimistisiin liikkeisiin.

Valinnan säilyttämisestä näytti kuitenkin olleen hieman hyötyä, koska tötterökursori oli hieman laiskaa kuplakursoria vähemmän virhealtis. Voisikin olla mielenkiintoista selvittää, kuinka laiska kuplakursori menestyisi, jos valinnan säilyttämistä sovellettaisiin vain tietyissä tilanteissa virheiden vähentämiseksi. Esimerkiksi jos hiiren nopeus on tietyn rajan alla ja liike hidastuu edelleen, voitaisiin ennustaa, että käyttäjä on aikeissa valita jonkin tietyn kohteen. Valinnan säilyttäminen tässä tilanteessa saattaisi hyvinkin laskea virhealttiutta ilman negatiivista vaikutusta

käyttömukavuuteen, joka valinnan säilyttämisellä oli, kun sitä sovellettiin tilanteesta riippumatta.

### 7.3. Kohde-etäisyyden arvioimisesta

Grossmanin ja Balakrishnanin oletivat, että kuplakursorin hidastuminen pienellä kohdetiheydellä voitaisiin selittää sillä, että kuplakursorin suuri koko häiritsi käyttäjiä. Tässä tutkimuksessa saatujen tulosten perusteella selitys vaikuttaa puutteelliselta. Havainnot kuplakursorin suuren koon negatiivisesta vaikutuksesta valinta-aikoihin lienevät parhaiten selitettävissä visuaalisella harhalla, jonka mukaan kohdevalintaan tarvittavan siirron pituus on helppo aliarvioida. Harha syntyy, kun kursorin ja kohteen välistä etäisyyttä arvioi kursorin ollessa suuri, mutta kursori kutistuu matkalla kohteeseen. Jos tarvittava siirto on arvioitua pidempi ja nopeus, jolla hiirtä liikutetaan, riippuu arvioidusta kohde-etäisyydestä valinta ymmärrettävästi myös kestää pidempään. Arvio kohde-etäisyydestä voi osoittautua dramaattisestikin vääräksi, jos kursori kasvaa ensin varsin suureksi ja kutistuminen tapahtuu myöhäisessä vaiheessa kursorin ollessa näennäisesti jo varsin lähellä maalikohdetta. Tällöin myös virheklikkauksia voisi kuvitella syntyvän varsin herkästi.

Jos kuplakursorin suuren koon negatiivinen vaikutus valinta-aikoihin todella johtuu ehdotetulla tavalla siitä, että kohde-etäisyyksien arvioiminen on vaikeampaa, havaintoa voitaneen käyttää hyväksi kohdevalinnan nopeuttamisessa. Kohde-etäisyyksien arvioimista voinee kuplakursorin tapauksessa kehittää jonkin verran pelkästään muuttamalla kursorin visualisointia. Kuplakursori pyrkii aina sulkemaan kohteen kokonaan sisälleen, vaikka riittäisi, että kursori vain ulottuu lähimpään kohteeseen. Jos kursori aina vain ulottuisi lähimpään kohteeseen, tästä voisi olla hyötyä kahdesta syystä. Sen lisäksi, että kursori olisi näin pienempi, mahdollinen odottamaton kutistuminen ennen maalikohdetta tapahtuisi myös aiemmin. Koska hiirtä liikutetaan nopeammin alkuvaiheessa ja hitaammin, kun kohdistin lähestyy maalikohdetta, koon kutistumisesta voisi olettaa olevan sitä enemmän haittaa, mitä lähempänä maalikohdetta kutistuminen tapahtuu.

Kuplakursoria voisi olla mahdollista kehittää myös muuttamalla laajenemis-algoritmia niin, että kupla ottaisi huomioon muutkin, kuin vain lähimmän kohteen. Jos koko ei olisi sidottu yhtä tiukasti lähimmän kohteen etäisyyteen (koko voisi toisinaan olla suurempikin), koon muutosten ei tarvitsisi olla niin äkkinäisiä, millä voisi olla positiivinen vaikutus valinta-aikoihin. Ymmärrettävästi koon kasvaminen entisestään voisi olla myös ongelmallista.

Vaikka kuplakursoria voisi olla mahdollista kehittää hieman vain muuttamalla sen visualisointia, perimmäinen ongelma on edelleen kuplan ajoittain suuri koko. Suuri koko valehtelee tarvittavan siirron pituudesta, jos kuplan on kutistuttava, ennen kuin kursori osuu maalikohteeseen. Onkin varsin vaikea arvioida kuinka paljon kohde-

etäisyyksien arviointia on mahdollista kehittää pienentämättä kursorin kokoa. Ongelman etäisyyden arvioimisesta voisi paljon suoraviivaisemmin myös poistaa kokonaan, jos kohteiden valinta-alueet piirrettäisiin näkyviin ja itse kursori olisi perinteinen pistekursori (kohteiden täytyisi kuitenkin ymmärrettävästi olla tavalla tai toisella edelleen valittavissa klikkaamalla kohteen valinta-alueetta, eikä itse kohdetta).

#### 7.4. Liikkeen suunnan suosimisesta

Grossman ja Balakrishnan ehdottivat, että kohdevalintaa voisi olla mahdollista kehittää entistäkin nopeammaksi muokkaamalla kursoria niin, että se esimerkiksi ellipsin muotoisena laajeni enemmän hiiren liikkeen suuntaan kuin hiiren liikkeen suuntaan nähden sivusuunnassa [Grossman & Balakrishnan, 2005a, p. 290]. Näin kohteiden valinta-alueista muodostuisi kuplakursoriinkin nähden suurempia. Tämänkaltaisesta ratkaisusta esimerkkinä on esitetty sittemmin ainakin *sädekursori* (engl. *beam cursor*) [Yin & Ren, 2006], kohdevalintamenetelmä kynäkäyttöisille näytöille. Sädekursori toimii siten, että kun kynällä koskee näyttöä, näytöllä näkyville kohteille pyritään laskemaan optimaalisen suuruiset valinta-alueet niin, että kohdistinta lähimmät kohteet ovat valittavissa minimaalisella kohdistimen siirrolla kohteen suuntaan. Liikkeen seurauksena kursori yhdistetään kohteeseen eräänlaisen säteen visualisoimana ja kohde voidaan valita nostamalla kynä ylös. Sädekursorilla suoritetuissa kokeissa Yin ja Ren havaitsivat sädekursorin toimineen kuplakursoriakin nopeammin, joskin kokeet suoritettiin kynäkäyttöisellä näytöllä, kuplakursorin ollessa ensisijaisesti hiirikäyttöinen kohdevalintamenetelmä.

Hiiren liikkeen suunnan suosiminen kursorin muotoa muuttamalla vaikuttaisikin olevan looginen tapa kehittää kuplakursoria ja tätä yritettiinkin alunperin myös tötterökursorin avulla. Jos tötterökursori olisi ollut mahdollista toteuttaa niin, että sitä olisi kontrolloitu tötterön puolesta välistä eikä kuplan keskeltä, kohteiden valinta-alueet olisivat kasvaneet huomattavasti. Kun tötterön pää (kupla) on suurin kohta kursoria, kohteisiin olisi toivon mukaan pitänyt edelleen olla myös suhteellisen helppo osua. Näin toteutettuna kursoriin ymmärrettiin kuitenkin liittyvän myös ongelmia sen suhteen, kuinka sitä tulisi mukauttaa sen osuessa kohteisiin näytöllä. Keskeinen ongelma tötterökursorin soveltamisessa näin oli se, että kun kontrollipiste oli tötterön keskellä, myös kontrollipiste liikkui aina kun häntä yhdistettiin uuteen kohteeseen. Kontrollipisteen siirtäminen puolestaan vaikutti aina myös siihen, millä suhteella seuraavat liikkeet kursoria kasvattivat tai kutistivat; kun häntä yhdistetään uuteen kohteeseen myös liikkeen suunta suhteessa valittuun kohteeseen ymmärrettävästi muuttuu.

Sen lisäksi, että oli epäselvää, kuinka tötterökursorin pitäisi reagoida valinnan siirtyessä kohteesta toiseen, myös kursorin toteuttaminen oli vaikeampaa kun sitä kontrolloitiin tötterön keskeltä. Koska kursorin haluttiin kasvavan tasaisesti kontrolli-

pisteen molemmin puolin (kursorin pituus aina kaksi kertaa kontrollipisteen etäisyys valittuun kohteeseen), ei ollut aiempaan tapaan yksiselitteistä, kuinka kuplan säde pitäisi laskea niin, että tötterön pituus kasvaa tasaisesti hiiren liikkeiden myötä ja kursorin osuessa kohteisiin. Kuplan säteelle tyydyttiinkin vain laskemaan epätarkka likiarvo ja kuplaa siirrettiin niin, että tötterön pituus oli aina haluttu. Kuplan koko ei tämän seurauksena käytännössä enää muuttunut täysin saumattomasti. Erityisesti kursorin osuessa uusiin kohteisiin kuplan koon muutokset olivat varsin suurpiirteisiä. Ratkaisevilta osin kursori kuitenkin toimi samalla tavalla kuin tötterökursori aikaisemminkin ja kursori käyttäytyi jokseenkin siedettävästi niin pitkään kun sitä malttoi liikuttaa hitaasti. Kursoria oli kuitenkin varsin vaikea kontrolloida kun kursoria liikutti nopeammin. Koska kursori ei tästä syystä lopulta ollutkaan siinä määrin käyttökelpoinen, että sen olisi voinut odottaa menestyvän vertailussa muihin kokeen aluekursoreihin nähden, sitä ei otettu mukaan tässä tutkimuksessa testattujen kursorien joukkoon.

Kursorin muodon muuttaminen liikkeen suuntaa suosivaksi on epäilemättä potentiaalinen tapa kehittää kohdevalintaa entisestään. Kokeilut tötterökursorilla antoivat kuitenkin syytä ymmärtää, että pitkäksi kasvavan, kohteeseen kiinnitetyn kursorin ohjaaminen sen keskeltä voi olla vaikeaa ja toimivan kohdevalintamenetelmän suunnittelu näin ei ole ongelmaton. Sädekursorin [Yin & Ren, 2006] soveltama ajatus valinta-alueiden laskemisesta dynaamisesti kohdistimen sijainnin mukaan vaikuttaisi mielekkäällä tavalla yksinkertaistavan problematiikkaa sen suhteen, kuinka valinnan voisi helpommin *ohjata* oikeaan kohteeseen.

## 8. Yhteenveto

Tämän tutkimuksen tavoitteena oli kehittää Grossmanin ja Balakrishnanin kuplakursoria edelleen. Laiskalla kuplakursorilla tähän pyrittiin muokkaamalla kursoria niin, että kupla valitsee uuden kohteen vasta kun kursorin lähellä on yksi kohde, joka on selvästi muita lähempänä. Tästä uskottiin olevan hyötyä useasta eri syystä, muun muassa sen vuoksi, että kursori on näin pienempi eikä ole koskaan epäselvyyttä siitä, mikä kohde on kuplaa lähinnä oleva kohde.

Koska tämä ratkaisu myös pienentää kohteiden valinta-alueita kuplakursoriin nähden, valinta-alueiden pienentymistä yritettiin kompensoida tötterökursorin tapauksessa antamalla kursorin säilyttää valinnan kohteeseen, johon kursori oli viimeksi ollut kosketuksessa. Valinnan säilyttäminen kohteeseen visualisoitiin eräänlaisen kuplasta kohteeseen piirretyn hännän avulla, joka pysyi yhdistettynä kohteeseen kunnes kursori osui seuraavaan kohteeseen. Tötterökursorinkaan ei odotettu olevan nopeampi kuin kuplakursori, mutta sen ei oletettu välttämättä häviävän erityisen dramaattisesti. Jos tötterökursori ei häviäisi paljoa tehokkuudessa, se voisi olla suositeltavissa, jos käyttäjät pitäisivät sitä kuplakursoria mieluisampana käyttää.

Molempia uusia kursoreita verrattiin kaksiulotteisessa ympäristössä 18 koehenkilön voimin suoritetuissa kokeissa sekä suorituskyvyn että laadullisten ominaisuuksien suhteen pistekursoriin ja kuplakursoriin. Tötterökursori hävisi odotetusti kuplakursorille, joskin ero jäi toivotulla tavalla varsin pieneksi. Kokeissa ei kuitenkaan havaittu, että valinnan säilyttäminen olisi vaikuttanut positiivisesti valinta-aikoihin, kun laiska kuplakursori oli jopa hieman sitä nopeampi. Valinnan säilyttäminen kohteeseen, ainakin siten kuin se tässä tutkimuksessa visualisoitiin, kuitenkin häiritsi käyttäjiä, mikä kävi selvästi ilmi käyttäjien täyttämistä arviointilomakkeista. Tötterökursori arvioitiin kokeen kahta muuta aluekursoria huonommaksi kaikissa kyselylomakkeen viidessä kohdassa ja erityisesti kysymyksessä, jossa käyttäjiä pyydettiin arvioimaan kursorin koon ja muodon sekä näiden muutosten miellyttävyyttä. Vapaamuotoinen palaute vahvisti pitkälti päättelyn valinnan säilyttämisen vastenmielisyydestä. Suuri osa koehenkilöistä ilmaisi suorasanaisesti kokeneensa kursorin ärsyttäväksi. Tämä näytti erityisesti johtuneen siitä, että valinnan säilyttämisen myötä kursori kasvoi usein varsin pitkäksi, jolloin kursori tuntui tarttuvan kohteisiin ja sitä oli vastenmielistä käyttää. Valinnan säilyttämisestä oli tulosten perusteella hyötyä lopulta vain varsin vähän, kun tötterökursorilla virheprosentti oli hieman laiskaa kuplakursoria pienempi. Tötterökursoria ei tulosten perusteella ole siis aihetta suositella käytettäväksi kuplakursorin sijasta.

Laiska kuplakursori menestyi kokeissa kuitenkin huomattavasti paremmin. Laiska kuplakursori oli jopa hieman nopeampi kuin tötterökursori ja käyttäjät pitivät sitä huomattavasti tötterökursoria miellyttävämpänä, jopa yhtä miellyttävänä tai

miellyttävämpänä käyttää kuin kuplakursoria. Tilastollisesti merkitsevää eroa kuplakursoriin ei syntynyt missään kyselylomakkeen kohdassa ja kursorit menestyivät tasaisesti, kun käyttäjiä pyydettiin sijoittamaan ne paremmuusjärjestykseen.

Mieltymys kuplakursoriin tai laiskaan kuplakursoriin oli riippuvainen sukupuolesta. Naispuoliset koehenkilöt pitivät enemmän jälkimmäisestä ja miespuoliset koehenkilöt edellisestä. Vaikka kursorien välistä nopeuseroa ei selvitetty erikseen sukupuolten välillä, voisi pohtia, arvostivatko miespuoliset koehenkilöt enemmän nopeutta ja naispuoliset koehenkilöt kursorin koon perusteella arvioitua kursorin ulkonäköä? Nämä olivat kuitenkin kursorien selkeimmät erot, vaikka laiska kuplakursori kärsi myös kuplakursoria hieman suuremmasta virheprosentista (joka laiskalla kuplakursorillakin pysyi varsin matalana). Nopeus tuskin oli kuitenkaan ainoa arviointiperuste miespuolisillekaan koehenkilöille, sillä miespuolisista koehenkilöistä vain kaksi yhdeksästä piti kokeiden selvästi hitainta kursoria, pistekursoria, kokeiden huonoimpana.

Yllättävää oli kuitenkin ennen kaikkea se, että laiska kuplakursori oli kokeissa varsin nopea. Se hävisi kuplakursorille niin vähän, että tämä nähtiin syynä miettiä uudelleen miksi kuplakursorin suurella koolla on negatiivinen vaikutus valinta-aikaan. On mahdollista, että kuplakursorin kuplan perusteella on usein vaikea arvioida kohde-etäisyyttä kun kupla on suuri. Se, mikä merkitys kohde-etäisyyden arvioimisen intuitiivisuudella tarkkaan ottaen on kohdevalinnan nopeuteen ja miten kohde-etäisyyden arvioimista voitaisiin parhaiten kehittää jää kuitenkin selvitettäväksi toisaalla. Tässä tutkimuksessa esitetyt tulokset antavat kuitenkin viitteitä siitä, että Grossmanin ja Balakrishnanin oletus ongelman syystä on puutteellinen. Kohde-etäisyyden arvioimisen vaikeus selittänee havainnot koon ongelmallisuudesta vakuuttavammin kuin suuren koon häiritsevyys. Tämä on mielenkiintoinen tutkimustulos, jolla voi hyvinkin olla merkitystä dynaamisten kohdevalintamenetelmien suunnittelulle tulevaisuudessa.

Tässä tutkimuksessa testattujen uusien kursorien ei lopulta oletettu nopeuttavan kohdevalintaa, eivätkä ne tähän pystyneetkään. Loogisin tapa nopeuttaa kohdevalintaa vaikuttaisikin edelleen olevan Grossmanin ja Balakrishnanin ehdottama hiiren liikkeen suunnan huomioon ottaminen. Tämä kasvattaisi kohteiden valinta-alueita ja näin ollen myös valinnan nopeutta. Potentiaalisia lähtökohtia kohdevalinnan nopeuttamiseksi tällä tavalla lienevät ainakin Grossmanin ja Balakrishnanin [2005a] ehdottama strategia kursorin muodon muokkaamisesta tai sädekursorin demonstroima kohdistimen sijainnin mukaan optimoitu näytön alueen jakaminen kohteiden valinta-alueiksi [Yin & Ren, 2006]. Tötterökursorin toivottiin alunperin olevan muutettavissa ensin mainitulla tavalla toimivaksi sijoittamalla kursorin kontrollipiste kursorin keskelle kuplan keskustan sijaan. Kuten kohdassa 7.4 kuvattiin, tähän liittyi kuitenkin ongelmia, joita ei pystytty siinä määrin menestyksekkäästi ratkaisemaan, että kursoria olisi ollut aihetta testata

tässä tutkimuksessa. Ottaen huomioon kohtaamani ongelmat, sekä Yinin ja Renin raportoima sädekursorin hyvä menestys sille suoritetuissa kokeissa, olen taipuvainen uskomaan, että nimenomaan sädekursori saattaa olla näistä vaihtoehdoista parempi lähtökohta kohdevalinnan nopeuttamiseksi tulevaisuudessa.

## Viiteluettelo

- [Accot and Zhai, 2003] Johnny Accot and Shumin Zhai, Refining Fitts' law models for bivariate pointing. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 19-26.
- [Asano et al., 2005] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima and Fumio Kishino, Predictive interaction using the delphian desktop. In: *Proceedings of the 18<sup>th</sup> Annual ACM Symposium on User Interface Software and Technology*, ACM Press, 133-141.
- [Baudisch et al., 2003] Patrick Baudisch, Edward Cuttrel, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin Bederson and Alex Zierlinger, Drag-and-drop and drag-and-pick: techniques for accessing remote screen content on touch-and pen-operated systems. In: *Proceedings of INTERACT*, IOS Press, 57-64.
- [Blanch et al., 2004] Renaud Blanch, Yves Guiard and Michel Beaudouin-Lafon, Semantic pointing: improving target acquisition with control-display ratio adaptation. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 519-526.
- [Fitts, 1954] Paul M. Fitts, The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* **47**, 381-391.
- [Grossman and Balakrishnan, 2005a] Tovi Grossman and Ravin Balakrishnan, The Bubble Cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 281-290.
- [Grossman and Balakrishnan, 2005b] Tovi Grossman and Ravin Balakrishnan, A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction* **12**, 3, 435-459.
- [Guiard et al., 2004] Yves Guiard, Renaud Blanch and Michel Beaudouin-Lafon, Object pointing: a complement to bitmap pointing in GUIs. In: *Proceedings of the 2004 Conference on Graphics Interface*, Canadian Human-Computer Communications Society, 9-16.
- [Isokoski and Raisamo, 2004] Poika Isokoski and Roope Raisamo. Speed and accuracy of six mice. *Asian Information-Science-Life*, **2**, 2, 131-140.
- [Jensen et al., 2002] Chris Jensen, Lotte Finsen, Karen Søggaard and Hanne Christensen, Musculoskeletal symptoms and duration of computer and mouse use. *International Journal of Industrial Ergonomics* **30**, 4-5, 265-275.



- [Kabbash and Buxton, 1995] Paul Kabbash and William Buxton, The “Prince” technique: Fitts’ Law and selection using area cursors. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM Press, 273-279.
- [MacKenzie, 1989] Scott MacKenzie, A note on the information-theoretic basis for Fitts’ law. *Journal of Motor Behavior* **21**, 323-330.
- [MacKenzie, 1992] I. Scott MacKenzie, Fitts’ law as a research and design tool in human-computer interaction. *Human Computer Interaction* **7**, 91-139.
- [MacKenzie and Buxton, 1992] I. Scott MacKenzie and William Buxton, Extending Fitts’ law to two-dimensional tasks. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 219-226.
- [McGuffin and Balakrishnan, 2002] Michael McGuffin and Ravin Balakrishnan, Acquisition of expanding targets. In: *Proceedings of the SIGCHI conference on Human factors in Computing Systems*, ACM Press, 57-64.
- [Shannon and Weaver, 1949] Claude E. Shannon and Warren Weaver, *Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [Soukoreff and MacKenzie, 2004] R. William Soukoreff and I. Scott MacKenzie, Towards a standard for pointing device evaluation: Perspectives on 27 years of Fitts’ law research in HCI. *International Journal of Human-Computer Studies* **61**, 6, 751-789.
- [Worden et al., 1997] Aileen Worden, Nef Walker, Krishna Bharat and Scott Hudson, Making computers easier for older adults to use: Area Cursors and Sticky Icons. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 266-271.
- [Yin and Ren, 2006] Jibin Yin and Xiangshi Ren, The Beam Cursor: A Pen-based Technique for Enhancing Target Acquisition. In: *People and Computers XX, Proceedings of HCI 2006*, Springer, 119-134.

## Liite 1: Kyselylomake kursorin arvioimiseksi

nimi : \_\_\_\_\_

**Ole hyvä ja vastaa seuraaviin kysymyksiin raksimalla ruutu asteikolta:**

-Kuinka nopealta kursorin käyttö tuntui annetuissa kohdevalintatehtävissä?

Erittäin hitaalta						Erittäin nopealta
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

-Kuinka helppoa kursorilla oli mielestäsi valita punainen kohde ilman virheitä?

Erittäin vaikeaa						Erittäin helppoa
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

-Kuinka loogiselta kursorin käyttäytyminen tuntui?

Erittäin epäloogiselta						Erittäin loogiselta
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

-Kuinka häiritsevältä kursorin koko ja muoto, sekä niiden muutokset tuntuivat?

Erittäin häiritsevältä						Ei ollenkaan häiritsevältä
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

-Kuinka miellyttävältä kursorin käyttö tuntui kaikkiaan?

Erittäin epämiellyttävältä						Erittäin miellyttävältä
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Kerro vielä vapaamuotoisesti mitä ajatuksia kursorin käyttö annettuja kohdevalintatehtäviä suorittaessasi herätti:**

---



---



---

**Liite 2: Kyselylomake paremmuusjärjestyksen arvioimiseksi**nimi :

---

Ole ystävällinen ja aseta kursorit vielä paremmuusjärjestykseen  
(1=paras, 2=toiseksi paras jne.)

<b>Kursori</b>	<b>Sijoitus</b>
Ensimmäinen	<input type="checkbox"/>
Toinen	<input type="checkbox"/>
Kolmas	<input type="checkbox"/>
Neljäs	<input type="checkbox"/>