

**Tuntopalautteen tukemat suorakäyttöiset
kaavionmuokkaustyökalut**

Jukka Raisamo

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Pro gradu -tutkielma
Kesäkuu 2002

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Jukka Raisamo: Tuntopalautteen tukemat suorakäyttöiset
kaavionmuokkaustyökalut
Pro gradu -tutkielma, 78 + 4 sivua
Kesäkuu 2002

Kaavionkäsittelyä tutkittaessa on perinteisesti pyritty kehittämään menetelmiä, jotka piirtävät ja järjestävät kaavioita automaattisesti. Viime vuosina on kuitenkin alettu arvostaa vuorovaikutteisia menetelmiä myös kaavioiden muokkauksessa, mutta usein tämä on rajoittunut vain lähinnä pieniin ja paikallisiin muokkauksiin yksittäisille olioille. Erilaisia kaavioiden laaja-alaisempaan muokkaamiseen suunniteltuja vuorovaikutteisia työkaluja ei ole yksittäisiä toteutuksia enempää kuitenkaan esitelty.

Tämän tutkielman päätarkoitus on esitellä joukko suorakäyttöisiä kaavionmuokkaustyökaluja, joiden avulla käyttäjä voi vuorovaikutteisesti laatia kaavioille haluamansa ulkoasun. Lisäksi tutkielmassa tarkastellaan ja arvioidaan tuntopalautteen käyttämistä visuaalista palautetta tukevana lisämodaliteettina huolellisuutta ja tarkkuutta vaativissa tehtävissä. Esimerkkinä tästä käytetään kaavionkäsittelysovellusta, jonka yleisestä toiminnasta sekä etenkin sovelluksen suorakäyttöisistä työkaluista pyritään tuntopalautteen avulla tekemään todentuntuisempia ja intuitiivisempia käyttöä.

Avainsanat ja -sanonnat: suorakäyttöisyys, kaavionkäsittelytyökalut, vuorovaikutteisuus, tuntopalaute, kaavion sommittelu, kaavion luettavuus.

Sisällys

1.	Johdanto	1
2.	Tuntopalaute	4
2.1.	Ihmisen aistit.....	4
2.2.	Tuntoaistin ominaisuuksia.....	6
2.2.1.	Tuntoaistin reseptorit.....	6
2.2.2.	Tuntoaisti käytännössä	9
2.3.	Tuntopalautteen määrittely.....	12
2.4.	Tuntopalaute työpöytäympäristössä.....	13
2.5.	Tuntopalautelaitteita	16
2.5.1.	Voimapalautehiiri.....	16
2.5.2.	Täriäpalautehiiri	18
2.5.3.	PHANToM.....	19
2.6.	Yhteenvedo.....	20
3.	Kaavionkäsittely	21
3.1.	Yleistä kaavionpiirrosta	21
3.2.	Kaavioiden luettavuus.....	22
3.2.1.	Luettavuuden kriteerejä	23
3.2.2.	Kriteerien samanaikainen noudattaminen.....	25
3.2.3.	Kriteerien arvottaminen.....	26
3.3.	Yhteenvedo.....	27
4.	Nykyiset kaavionkäsittelymenetelmät.....	28
4.1.	Automaattiset sommittelualgoritmit	28
4.1.1.	Staattiset algoritmit.....	28
4.1.2.	Dynaamiset algoritmit.....	29
4.2.	Komentopohjaiset tasaustyökalut.....	30
4.3.	Rajoitteet.....	33
4.3.1.	Snap-Dragging.....	33
4.3.2.	Vuorovaikutteiset rajoitteet	34
4.4.	Suora-asemointi	37
4.5.	Suorakäyttöiset tasaustyökalut.....	38
4.5.1.	Tasauspuikko	38
4.5.2.	Tasausluutatyökalu.....	40
4.6.	Yhteenvedo.....	42
5.	Uudet kaavionkäsittelytyökalut.....	45
5.1.	Työkalujen vuorovaikutteisuus	45
5.2.	Työkalujen toiminta.....	47

5.3. Vaatimuksia uusille työkaluille	48
5.4. Työkalujen esittely.....	49
5.4.1. Kaappausellipsi	49
5.4.2. Tasausellipsi.....	51
5.4.3. Tasavälipuikot.....	52
5.5. Yhteenveto.....	54
6. Tuntopalautteen lisääminen kaavionpiirtosovellukseen	56
6.1. Toteutusympäristö.....	56
6.2. TouchSense-tehostetyypit.....	57
6.3. Tuntopalautetehosteiden suunnittelu	59
6.3.1. Taustaa.....	60
6.3.2. Vaatimuksia.....	61
6.3.3. Tuntopalaute työkalujen tukena	63
6.4. Tuntopalautteen arviointi	66
6.5. Yhteenveto.....	68
7. Lopuksi	70
 Viiteluettelo.....	 72

1. Johdanto

Lisääntynyt tiedonprosessointitarve on aiheuttanut sen, että erilaiset kaaviot ovat tulleet yhä tärkeämmiksi apuvälineiksi kuvaamaan monimutkaisia tietorakenteita ja siten tukemaan vaikeaselkoisten asioiden ymmärtämistä. Menetelmät luoda, käsitellä ja muokata kaavioita eivät kuitenkaan ole kehittyneet vastaavalla nopeudella; edelleen lähes jokainen kaavionpiirtoon tarkoitettu sovellus käyttää hyväkseen joko *automaattisia sommittelualgoritmeja* (automatic layout algorithms) tai manuaalisesti ohjattavia *komentopohjaisia tasaustyökaluja* (command-based alignment tools). Kumpikaan näistä menetelmistä ei kuitenkaan tarjoa käyttäjälle itselleen kovin suurta vapautta järjestää kaavioita haluamaansa muotoon.

Onkin yleisesti tunnustettu, että kaavioiden käsittelyyn tarvitaan vuorovaikutteisia menetelmiä, joiden avulla käyttäjä itse voi järjestää kaavion *ulkoasun* (layout) semanttisesti oikeaksi kokemaansa muotoon. Kuitenkin, huolimatta kaupallisten graafisten käyttöliittymien myötä paljon mielenkiintoa saavuttaneesta *suorakäyttöisyyden* (direct manipulation) ihanteesta, tällaisia tehokkaita suorakäyttöisiä kaavionmuokkaustyökaluja ei edelleenkään ole laajalti sovellettu kaavioiden käsittelyyn.

Suorakäyttöisyys-käsitteen keksinyt Shneiderman [1982] on todennut, että käyttöliittymien tulisi olla suorakäyttöisiä, tarkoittaen pyrkimystä antaa käyttäjälle välineet näytöllä olevien kohteiden välittömään käsittelyyn. Ziegler ja Fähnrich [1988] sanovat suorakäyttöisyyden perustuvan tapahtumaketjuun, jossa käyttäjä valitsee kohteen ja suoritettavan toiminnon, minkä jälkeen järjestelmä reagoi tähän suorittamalla halutun tehtävän ja antamalla tästä käyttäjälle välittömän palautteen. Olennaista suorakäyttöisyydessä on siis käyttäjän ja sovelluksen vuorovaikutteisuus.

Toimiakseen kunnolla, suorakäyttöisen käyttöliittymän tulee olla intuitiivinen käyttöä. Hutchins *et al.* [1986] toteavat, että helposti omaksuttavat intuitiiviset käyttöliittymät perustuvat aiemmille kokemuksille samankaltaisissa tilanteissa. He korostavat myös, että suorakäyttöisyys on eri asia kuin helppokäyttöisyys – ainoastaan hyvin suunniteltuna ja toteutettuna suorakäyttöinen järjestelmä on samalla myös helppokäyttöinen. Tämän käsitteellistääkseen Hutchins *et al.* erottavat kaksi päätekijää, etäisyyttä, jotka vaikuttavat suorakäyttöisten käyttöliittymien omaksumiseen:

1. *etäisyys päämääristä fyysisen järjestelmään* (gulf of execution), ja
2. *etäisyys fyysisestä järjestelmästä päämääriin* (gulf of evaluation).

Etäisyydellä päämääristä järjestelmään Hutchins *et al.* tarkoittavat sovelluksen toiminnan loogisuutta, jolloin tietyn syötteen antaminen saa aikaan aina

saman, odotetun ja suoritettavaan tehtävään sopivan toiminnon. Etäisyys järjestelmästä päämääriin puolestaan tarkoittaa annetusta komennosta saadun palautteen totuudenmukaisuutta, eli käytännössä sen vastaavuutta sovelluksen suorittaman toiminnon kanssa. Näiden toimintojen tulee lisäksi tapahtua reaaliajassa, niiden tulee olla *suoravaikutteisia* (direct engagement) [Hutchins *et al.*, 1986]. Tällöin käyttäjälle tulee tunne, että hän voi teoillaan vaikuttaa sovelluksen toimintaan, aivan kuten todellisessa maailmassakin.

Halskov Madsen [1994, 2000] sanoo, että käyttöliittymien omaksumisprosessia voidaan auttaa valitsemalla niiden työkaluille metaforia, jotka ovat käyttäjille ennestään tuttuja joistakin aiemmista yhteyksistä. Esimerkiksi *tasauspuiikon* (alignment stick) [Raisamo ja Räihä, 1996] työkalukursori noudattaa viivainmetaforaa, joka auttaa käyttäjää ymmärtämään työkalun toimintatavan aiempien arkielämän kokemustensa avulla tehden siten sen käytöstä intuitiivista. Hyvän metaforan ominaisuuksiksi luetaankin yksinkertaisuus, yksiselitteisyys ja havainnollisuus sekä yhteys johonkin aiemmasta yhteydestä tuttuun toimintaan [Halskov Madsen, 2000]. Metaforan täyttäessä nämä vaatimukset käyttäjän on alusta alkaen helppo samaistaa työkalun toiminnallisuus sen visualisointiin, jolloin työkalun käytön vaatima kognitiivinen taakka vähenee.

Nykyiset graafiset käyttöliittymät tarjoavat käyttäjälle ylivoimaisen käytettävyyden ja vuorovaikutusmahdollisuuden aiempiin tekstipohjaisiin käyttöliittymiin verrattuna, mutta edelleen kommunikaatio ihmisen ja tietokoneen välillä on ihmisten väliseen moniulotteiseen kontaktiin verrattuna alkeellista. Syötteet tietokoneelle annetaan useimmiten joko näppäimistöllä, osoitinlaitteella tai joissakin harvinaisemmissa tapauksissa esimerkiksi puheena. Koneen käyttäjälle antama palaute puolestaan on pääosin visuaalista, joissakin tilanteissa myös ääntä. Tämä ei ole lähelläkään ihanteellista tilannetta, koska luonnostaan ihminen tyytyy vain harvoin havainnoimaan lähiympäristöään pelkästään näkö- ja kuuloaistien avulla. Koskettaminen, lähellä olevien kohteiden tunnusteleminen, onkin olennainen osa ihmisen ja muun fyysisen maailman vuorovaikutusta.

Tuntoaisti on tärkeä osa jokapäiväistä elämäämme. Sen avulla tiedämme esimerkiksi, mikä esine meillä on kädessämme ja mitä voimme sillä tehdä, tai kykenemme löytämään valokatkaisijan pimeässä. Tuntoaistin välityksellä ihminen saa paljon sellaista informaatiota lähiympäristöstään, mihin muiden aistien tarkkuus ei riitä tai mitä niiden ominaisuudet eivät kata. Voikin siis perustellusti sanoa, että tuntoaistin avulla todellisuus hahmottuu meille paljon elävämpänä.

Tietokoneiden käyttöliittymissä *tuntopalautetta* (haptic feedback) voidaan käyttää paitsi yksinään [Jansson ja Billberger, 1999] niin myös joko äänipalautteen rinnalla [McGee *et al.*, 2000], visuaalisen palautteen rinnalla [Akamatsu ja MacKenzie, 1996; Dennerlain *et al.*, 2000] tai sitten yhdistää nuo kaikki kolme eri palautemodaliteettia [Ramstein, 1995; Ramloll ja Brewster, 2002]. Tuntopalautteen avulla voidaankin esimerkiksi antaa sokeille mahdollisuus käyttää heille muuten hankalia graafisia käyttöliittymiä, mitä äänipalautteen käyttö tuntopalautteen rinnalla edelleen parantaa. Yhdessä visuaalisen tai sekä visuaalisen että äänipalautteen kanssa tuntopalaute mahdollistaa korkeatasoisen katse-käsi-koordinaation, jonka avulla voidaan esimerkiksi simuloida erilaisia suurta tarkkuutta vaativia kirurgisia operaatioita.

Visuaalisesta palautteesta poiketen ääni- ja tuntopalautetta ei useinkaan voi käyttäjää häiritsemättä käyttää jatkuvana vaan ne on ennemminkin suunniteltava kuvaamaan käyttöliittymässä tapahtuvia tilanvaihdoksia ja muita epäjatkuvia tapahtumia.

Tämän tutkielman tarkoituksena on kehittää joukko suoraikäyttöisiä kaavionkäsittelytyökaluja, joita voi käyttää kaavioiden monipuoliseen muokkaamiseen. Työkalut tekevät kaavioiden muokkaamisprosessista vuorovaikutteisen antamalla suurimman osan päätäntävällästä käyttäjälle, auttaen kuitenkin samalla tätä suoriutumaan tehtävästään. Tutkielmassa arvioidaan myös kehitettyjen työkalujen käyttöä ja toimintaa tukemaan tuotetun tuntopalautteen käytön etuja ja mahdollisuuksia näköaistia täydentävänä palautemodaliteettina. Tuntopalautteen tarkoitus on havainnollistaa käyttäjälle työkalun toimintaa ja tehdä sovelluksen graafisesta käyttöliittymästä samalla entistä intuitiivisempi ja suoraikäyttöisempi.

Tutkielman toisessa luvussa kerrotaan tarkemmin ihmisen tuntoaistista, sen hyödyllisistä erikoisominaisuuksista sekä tuntopalautteen sovellusmahdollisuuksista tietokoneen käyttöä tukevana lisäpalautemodaliteettina. Luvussa kolme tutustutaan kaavionkäsittelyn perusteisiin, minkä jälkeen luvussa neljä käydään läpi yleisimmät nykyisin käytetyt kaavionkäsittelymenetelmät. Viidennessä luvussa esitellään neljää uutta vuorovaikutteista kaavionkäsittelytyökalua sekä kuvaillaan niiden käyttöä ja toimintaa. Luvussa kuusi pohditaan tuntopalautteen lisäämistä uusiin työkaluihin sekä pohditaan sen tuomia mahdollisuuksia parantaa työkalujen käytettävyyttä. Tutkielman viimeinen, seitsemäs, luku sisältää yhteenvedon tutkielmasta sekä joitakin aiheita jatkotutkimukselle.

2. Tuntopalaute

Tämä luku on johdatus ihmisen tuntoaistin ominaisuuksiin sekä tuntopalautteen olennaisiin käsitteisiin ja käyttömahdollisuuksiin visuaalista palautetta tukevana lisäpalautekanavana tietokonesovelluksissa. Tarkoitus on tutustuttaa lukija tuntopalautteen yleisiin määritelmiin sekä sen käyttöön tietokoneen graafisessa työpöytäympäristössä.

Luvun ensimmäisessä kohdassa esitellään ihmisen aistit ja aistinprosessin yleinen kulku. Toisessa kohdassa kerrotaan lähemmin tuntoaistista sekä esitetään tilanteita ja tapoja, joissa ihminen tuntoaistia käyttää. Luvun kolmannessa kohdassa käsitellään tuntopalautetta esittäen määritelmiä, minkä jälkeen kohdassa neljä luodaan katsaus tuntopalautteen käyttöön työpöytäympäristössä. Kohdassa viisi esitellään tarkemmin kolme erityyppistä tuntopalautelaitetta, ja lopuksi kohdassa kuusi esitetään yhteenveto luvun sisällöstä.

2.1. Ihmisen aistit

Ihminen on sopeutunut käyttämään monipuolista aistiverkostoaan tehokkaasti hyväkseen informaation keruuseen ympäristöstään. Hän voi esimerkiksi tarkkailla ympäröivää maailmaa näköaistinsa avulla tai jähmettyä hetkeksi paikoilleen kuuntelemaan tarkemmin yllättävän äänen kuullessaan. Ihminen poikkeaa monista muista nisäkkäistä siinä, että näille elintärkeä hajuaisti ei ole meille erityisen oleellinen – ihmiselle tärkeimpiä aisteja ovatkin näkö-, kuulo- ja tuntoaisti [Goldstein, 1999].

Maku- ja hajuaisti ovat kemiallisia aisteja, jotka perustuvat lähiympäristön kemiallisten ärsykkeiden havaitsemiseen [Goldstein, 1999]. Kyky aistia hajuja on monille eläinlajeille pääasiallinen tapa hankkia tietoa ympäröivästä maailmasta. Ihmiselle haju- ja etenkin makuaisti ovat kuitenkin toissijaisia aisteja, koska niillä ei ole ollut meille ratkaisevaa merkitystä lajin selviytymisen kannalta. Goldstein [1999] sanoo näiden aistien olleen evolutionaarisesta näkökohdasta katsottuna ihmiselle hyödyllisiä lähinnä kahdella tapaa. Ensiksikin, ne ovat auttaneet ihmisiä havaitsemaan keholle haitallisia aineita, jotka tulisi hylätä, ja toiseksi tunnistamaan aineita, jotka ovat selviytymisen kannalta oleellisia ja tulisi siten myös käyttää ravinnoksi. Lisäksi yleisesti ajatellaan, että ihmisten ominaishajut, feromonit, ovat olleet etenkin ihmislajin alkuaikoina eräs osatekijä parinmuodostukselle.

Näkö- ja kuuloaistit ovat olleet koko ihmislajin olemassaoloajan tärkeitä apuvälineitä kommunikaation kehittymiselle, vaarojen havaitsemiselle ja niistä kertomiselle. Näkö- ja kuuloaisteja onkin joissakin yhteyksissä kutsuttu sosiaalisiksi aisteiksi, mutta toki niillä on myös muita käyttötarkoituksia kuin

pelkkään kommunikointiin liittyvät seikat. Silmien ja korvien välityksellä saamme huomattavan määrän informaatiota joka hetki, mistä vain pieni osa lopulta onnistuu kiinnittämään tietoisien mielenkiintomme. Näkö- ja kuuloaisteja voisikin melko perustellusti kutsua kaukoasteiksi, ei-mekaanisiksi aisteiksi, koska niillä ei voi fyysisesti päästä käsiksi tarkasteltavien ärsykkeiden lähteisiin.

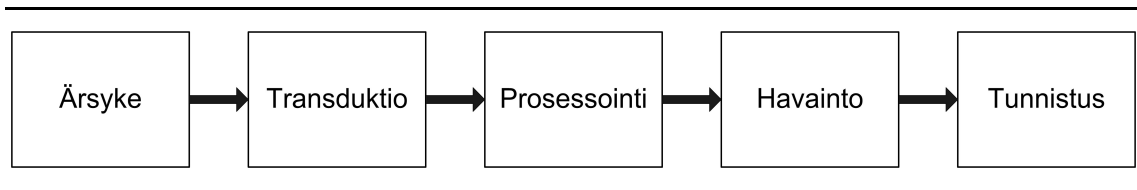
Tuntoaisti eroaa neljästä muusta aistista lähinnä kahdella tapaa: tuntoaistivien reseptoreja on muista aisteista poiketen joka puolella kehoa, ja tunto-reseptorien vastaanottamien monen erityyppisen ärsykkeen ja niiden herättämien aistimusten kirjo on monipuolinen. Tuntoaisti on mekaaninen aisti, joka vaatii tarkasteltavan esineen käsien ulottuville, jotta siitä voidaan saada jotakin informaatiota. Koskettamalla ja tunnustelemalla ihminen voi saada lähiympäristöstään paljon sellaista tietoa, jota muiden aistien avulla ei ole mahdollista havaita; esimerkiksi esineen pinnan tunteminen, sen lämpötilan, kovuuden ja karkeuden aistiminen ihoa vasten saa meidät vakuuttumaan kohteen todellisuudesta.

Tuntoaisti ei vaadi valoa tai ääntä toimiakseen, päinvastoin, itse asiassa tuntoaisti on herkimmillään, kun muut aistit eivät häiritse sen toimintaa. Ihmiset voivatkin tunnistaa entuudestaan tuttuja esineitä pelkän tuntoaistin avulla nopeasti, jopa parissa sekunnissa [Goldstein, 1999]. Goldstein sanoo myös, että ihmisen tuntoaisti on hyvä havaitsemaan ja tunnistamaan ärsykeitä, mutta melko huono erottamaan eriaikaisia palautteita toisistaan – toisin sanoen tuntoaistin niin kutsuttu ärsykemuisti ei ole kovinkaan hyvä. Kuulo- ja etenkin näköaisti voivat tässä tapauksessa vahventaa ja tarkentaa tunnustelemalla saatua informaatiota, kuten myös tarkasteltavan kohteen ollessa entuudestaan outo tai liian suurikokoinen käsillä tunnusteltavaksi.

Goldstein [1999] esittää mallin ihmisen aistimien hahmottamisprosessin etenemiselle (kuva 2.1). Ihmisen havaintoprosessia kuvaamaan on laadittu myös monia osin monimutkaisempiakin malleja, mutta Goldsteinin esittämä yksinkertainen ja suoraviivainen malli riittää tähän tarkoitukseen erinomaisesti. Goldsteinin kirjassa kuvan 2.1 malli on esitelty käytännön näköaistiesimerkin avulla, mutta muokkasinkin mallia vastaamaan nimenomaan tuntoaistin erityisominaisuuksia yhdistämällä erilliset *etäinen ärsyke* (distal stimulus) ja *läheinen ärsyke* (proximal stimulus) -vaiheet yhdeksi ärsykevaiheeksi ja poistamalla tunnistusvaiheen jälkeisen *toiminta* (action) -vaiheen.

Mallissa ihmisen aistinreseptorit vastaanottavat *ärsykkeen* (stimulus), joka *transduktioprosessissa* (transduction) muuntuu neuronien aktiopotentiaaleiksi. Tarpeeksi vahvat aktiopotentiaalit aktivoivat kokonaisia hermoverkkoja prosessoimaan saatuja ärsykeitä, minkä seurauksena syntyy *havainto*

(perception) kohteesta. Lopuksi saatu havainto *tunnistetaan* (recognition) kuuluvaksi johonkin merkityskategoriaan ja yhdistetään aiempaan kokemukseen.



Kuva 2.1. Hahmottamisprosessin askeleet. Muokattu lähteen [Goldstein, 1999] pohjalta.

2.2. Tuntoaistin ominaisuuksia

Tuntoaisti on *ihoaisti* (cutaneous sense), jota somatosensorinen, omasta kehosta tuleviin tuntoaistireseptoreihin liittyvä järjestelmä palvelee. Somatosensoriseen järjestelmään kuuluu tuntoaistin lisäksi raajojen käyttöön liittyvät *asentoaisti* (proprioceptive sense) ja *liikeaisti* (kinesthetic sense), sekä kipu- ja lämpöaisti.

Ihossa on useita eri tarkoitukseen erikoistuneita aistinreseptoreita, jotka välittävät tietoa pääasiassa kosketuksesta ja kivusta. Reseptorien määrä vaihtelee paikoittain, esimerkiksi sormenpäissä ja kielessä tuntoreseptoreita on erityisen tiheässä, kun taas selässä niitä on hyvin harvassa. Liikeaistin lihaksissa, nivelissä ja ihossa sijaitsevat reseptorit puolestaan antavat meille tietoa kehon eri osien sijainnista ja liikkeestä. Kyvyt aistia liikettä ja liikuttaa raajoja ovat kuitenkin toisistaan erillisiä; voimme esimerkiksi aistia raajan asennon myös silloin, kun sitä liikutetaan passiivisesti [Goldstein, 1999].

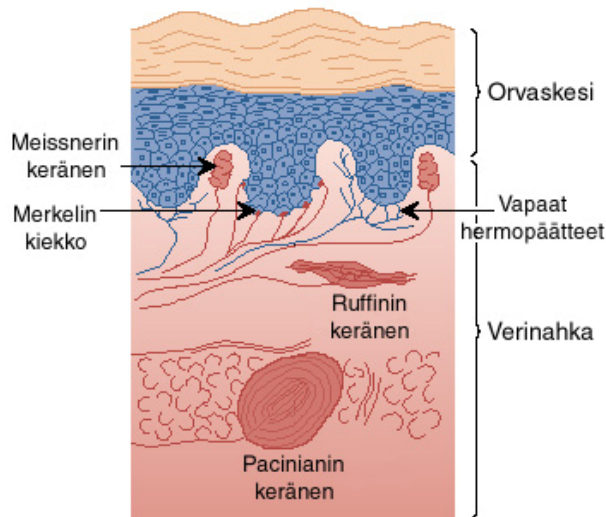
2.2.1. Tuntoaistin reseptorit

Iho on ihmisen selvästi suurin elin, ja sen pinta-ala on aikuisella miehellä keskimäärin 18.000 cm². Suurin osa tuntoaistin informaatiosta tulee kuitenkin käsien avulla, jotka ovat sekä havaitsevia että manipuloiivia kehonosia. Yksinomaan käsissä onkin laskettu olevan yhteensä noin 17.000 aistinreseptoria. [Goldstein, 1999]

Kädet ja niissä etenkin sormet ovat yksiä ihon herkimmistä tuntevista osista, sormenpäissä onkin laskettu olevan yhteensä jopa 135 tuntoaistinreseptoria neliösenttiä kohti [Smith, 1997]. Smithin [1997] mukaan sormet ovat pintarakenteiden erottamisessa herkimmillään noin 230 Hertsin taajuudella, mutta eivät kykene erottamaan kahta peräkkäistä yli 320 Hertsin voimasignaalia toisistaan. Burdea [1996] sanoo, että aistimme usein miten tunto- ja liikeärsyksiä paljon nopeammin kuin kykenemme vastaamaan niihin. Käsi ja sormet pystyvät esimerkiksi synnyttämään voimia, joiden taajuus on maksimissaan 5–10 Hertsiä. Tämä on melko vähän verrattuna siihen,

että liike- ja asentoaistimusten taajuusalue on 20–30 Hertsiä, yleisten kosketusaistimusten erottelutaajuus 0–400 Hertsiä ja hyvin hienopiirteisten pintarakenteiden yksityiskohtien havainnointi saattaa vaatia jopa 5.000–10.000 Hertsin värähtelytaajuuden.

Tuntoaisti perustuu pääosin neljän tyyppisiin aistinreseptoreihin, jotka kaikki antavat erilaista tietoa kosketuksesta. Reseptorien herkkyys riippuu luonnollisesti siitä, kuinka lähellä ihon pintaa ne sijaitsevat [Goldstein, 1999]. Kipu- ja lämpöaistimuksia puolestaan välittävät ihon pintaosissa sijaitsevat vapaat hermopäätteet. Kuvassa 2.2 näkyy tuntoaistinreseptorien ja vapaiden hermopäätteiden sijainti ihokudoksessa, joka jakautuu kahteen osaan, orvaskeen ja verinahkaan.



Kuva 2.2. Aistinreseptorien ja vapaiden hermopäätteiden sijainti ihokudoksessa.

Johnson [2001] on tehnyt yhteenvedon vuosien varrella tuntereseptoreista tehtyjen tutkimusten tuloksista. Hänen mukaansa heti ihokerroksen alla sijaitsevat *Merkelin kiekot* (Merkel receptors) ovat erityisen herkkiä kosketeltavan kohteen pinnan yksityiskohdille, ja tuottavat siten tarkan kuvan kohteiden muodosta ja pintarakenteesta. Edellisten tavoin lähellä ihon pintaa sijaitsevat *Meissnerin keräset* (Meissner corpuscles) puolestaan välittävät tietoa koko käden liikesignaaleista, joista aivojen tuntokeskus saa tietoa ihoa koskettavien kohteiden liikkeestä ja tartuntaotteen pitävyydestä. Syvemmällä verinahassa sijaitsevat *Ruffinin keräset* (Ruffini corpuscles) välittävät tietoa koko käden ihon venymisestä, mistä aivojen tuntokeskus selvittää ihon pintaa vasten liikkuvan kappaleen liikesuunnan sekä käteen vaikuttavien voimien suunnan. Ihonalaisessa rasvakerroksessa sijaitsevat *Pacinianin keräset* (Pacnian corpuscles)

puolestaan ovat herkkiä korkeataajuuksiselle värinälle, jota käteen osuvat tai kädessä pidettävät esineet tuottavat.

Taulukkoon 2.1 on koottu tuntoaistin eri reseptorit, niiden parhaat herkkyystaajuudet, mukautuvuusnopeudet ärsykkeisiin, reseptorien aistimusalueet, reseptorien prosentuaaliset osuudet kädessä ja aistimukset, joita ne pääasiassa välittävät.

Reseptori	Herkkyystaajuus	Mukautuvuus	Aistimusalue	Osuus (%)	Aistimus
Merkelin kiekko	2 – 32 Hz	Hidas	Pieni, tarkka	29	Paine, tekstuuri
Meissnerin keränen	8 – 64 Hz	Nopea	Pieni, tarkka	58	Liike, värähtely
Ruffinin keränen	< 8 Hz	Hidas	Suuri, epätarkka	4	Ihon venytys
Pacinianin keränen	> 64 Hz	Nopea	Suuri, epätarkka	9	Tärinä, kutitus

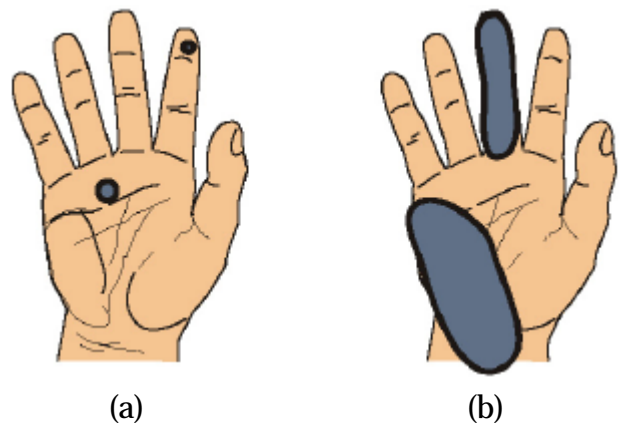
Taulukko 2.1. Tuntoaistin reseptorien mitattavia ominaisuuksia. [Wall ja Harwin, 2001]

Johnson [2001] sanoo, että yhdistetty psykofyysinen ja neurofysiologinen tutkimus on tuottanut melko kattavan kuvan tuntoaistin toimintamekanismeista, kuten esimerkiksi eri reseptorien vastuualueista ja ominaisuuksista. Aistinreseptorien ominaisuuksista on kuitenkin olemassa monia erilaisia tutkimustuloksia, esimerkiksi Goldstein [1999] ja Burdea [1996] esittävät kumpikin hieman erilaisia herkkyystaajuuslukemia ja aistimusominaisuuksia kuin taulukossa 2.1 esitetyt tiedot. Tällaiset eroavaisuudet ovat ymmärrettäviä, koska tuntoaisti on moniulotteinen ja hyvin läheisesti yhteydessä myös raajojen liike- ja asentoaisteihin [Smith, 1997]. Tämän huomaa esimerkiksi siitä, että monien kappaleen fyysisten perusominaisuuksien tunnistaminen (muun muassa kovuus, muoto ja paino) vaatii avukseen myös muita somatosensorisia aisteja.

Yksimielisiä eri tutkijat tuntuvat sen sijaan olevan aistinreseptorien mukautuvuudesta ja vaikutusalueesta. Nopeasti mukautuvat reseptorit reagoivat herkästi ärsykkeisiin, mutta niiden antama vaste sammuu heti, kun ärsyke loppuu tai pysyy muuttumattomana. Hitaasti mukautuvat reseptorit puolestaan jatkavat toimintaansa pitkän aikaa vaikka ärsyke pysyisikin vakiona tai jopa loppuisi. Eri reseptoreiden vaikutusalueiden koko ja tarkkuus puolestaan riippuvat niiden sijainnista ihossa. Lähes ihon pinnassa sijaitsevien Merkelin kiekkojen ja Meissnerin kerästen aistimusalueet ovat pieniä ja

tarkkoja, kun ne syvemmillä sijaitsevilla Ruffinin ja Pacinianin keräsillä ovat puolestaan suuria ja epätarkkoja.

Meissnerin ja Pacinianin keräset välittävät erilaisten työkalujen käytölle tarpeellisia aistimuksia, joita esimerkiksi useimmat tuntopalautelaitteet pyrkivät stimuloimaan keinotekoisien tuntoaistimusten tuottamiseksi. Näistä kahdesta reseptorityypistä Meissnerin keräset vastaanottavat matalataajuuksista värähtelyä ja yksittäisiä tärähdyksiä, jotka viestivät virtuaalisten pintojen yksityiskohdista. Pacinianin keräset puolestaan tekevät mahdolliseksi yleisen tuntoaistimuksen luomisen keinotekoisesti vastaanottamalla korkeataajuuksista värinää, jonka avulla ihmisen aistijärjestelmää voidaan huijata pitämään virtuaalisia esineitä oikeina. Kuvassa 2.3 on esitetty esimerkit kämmeneen ja sormenpäähän aiheutettujen ärsykkeiden tuottamien tuntoalueiden laajuudesta Meissnerin kerästen (kuva 2.3a) ja Pacinianin kerästen (kuva 2.3b) aistimina.



Kuva 2.3. Esimerkki Meissnerin kerästen (a) ja Pacinianin kerästen (b) tuntoalueiden laajuudesta sormenpäissä ja kämmenessä. [Vilis, 2000]

2.2.2. Tuntoaisti käytännössä

Smith [1997] sanoo tuntoaistin poikkeavan esimerkiksi näköaistista siinä, että sille ei ole tärkeää pelkästään se, mitä sen aistinreseptorit havainnoivat, vaan myös se, minkälaisia liikkeitä tuohon havainnointiin on käytetty. Goldstein [1999] esittääkin ihmisen voivan tunnustella ympäristöään kahdella eri tapaa, aktiivisesti tai passiivisesti. Aktiivinen tunnustelu tarkoittaa sitä, että ihminen hallitsee itse tunnustelemiseen käyttämiään liikkeitä. Passiivisessa tunnustelussa sen sijaan ihmisen kättä tai sormeä ohjaa jokin ulkopuolinen taho, esimerkiksi toinen ihminen tai kone. Aktiivisesti tunnustellessamme keskitymme kappaleen äärirajoihin ja kappaleen muihin yleisiin fyysisiin ominaisuuksiin, kun passiivinen tunnustelu puolestaan keskittää huomionsa pääasiassa kappaleen pintarakenteen yksityiskohdista saamiemme aistimusten

seuraamiseen. Aktiivinen tunnustelu on siis passiivista tarkempi hahmotamaan kohteita kokonaisuuksina, kun passiivinen tunnustelu herkistää ihmisen havaitsemaan pieniä yksityiskohtia tunnusteltavan kohteen pinnasta.

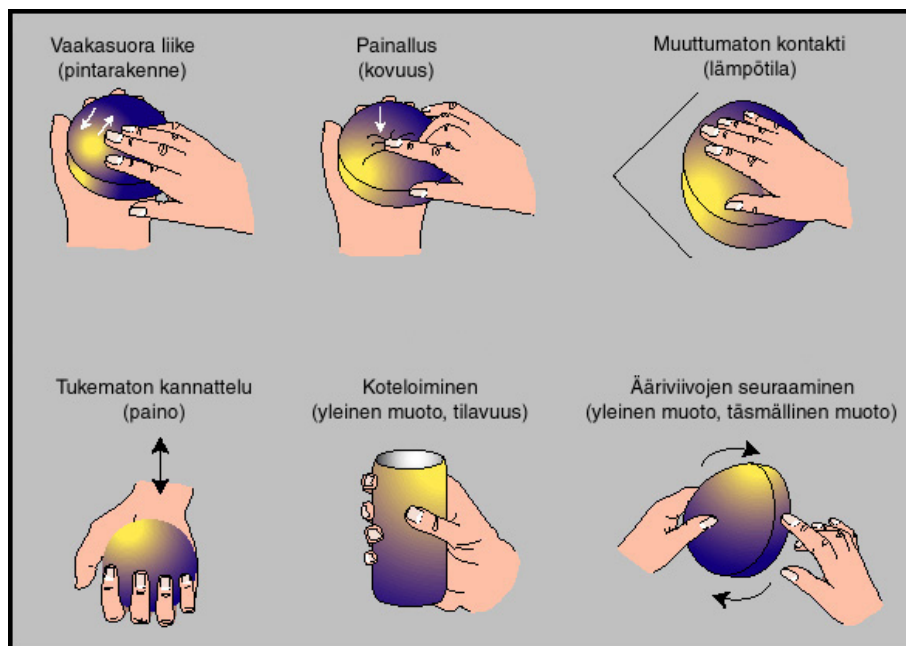
MacLean [2000] esittää, että fyysistä kanssakäyntiä ymmärtääkseen on koskettamisen ainutlaatuisia ominaisuuksia tarkasteltava ihmisiä seuraamalla. MacLeanin mukaan koskettaminen on kaksisuuntaista toimintaa, joka tukeutuu suurelta osin aktiiviseen tunnusteluun; se on eräänlainen ele, joka ilmaisee toiminnallisia viestejä yhtä hyvin kuin tunteitakin. Koskettaminen onkin usein sosiaalisesti latautunutta, koska se on aina tarkoituksellista, sosiaalisesti kajoavaa ja sitovaa. Maclean sanoo myös, että kosketus on monitahoinen aistinkanava, jolla on monta laadullisesti erilaista komponenttia ja jonka tuottama informaatio yhdistyy muilta aisteilta saatavaan informaatioon. Kosketuksen avulla voidaankin erittäin tarkasti hallita erilaisia kohteita ja erotella niitä toisistaan, mutta erilaisten tuntoaistimusten muistaminen ja mieleenpalauttaminen, niiden yhdistäminen aiempiin aistimuksiin, on vaikeampaa kuin esimerkiksi erilaisten hajujen, makujen, äänien ja värien [Goldstein, 1999].

MacLean [2000] luettelee myös erilaisia syitä koskettamiselle. Näitä ovat esimerkiksi jonkin tehtävän tekeminen, kohteen tilan tai ominaisuuksien tutkiminen, kommunikointi, reaktion aiheuttaminen esimerkiksi tönäisemällä ja toiminnan suorittamisen vahvistaminen. Hieman vapaammassa olosuhteissa kosketamme myös kokeaksemme esteettistä nautintoa tai mukavuutta, hypistelläksemme jotakin purkaaksemme jännitystä ja lähestyäksemme fyysisesti tai henkisesti toista ihmistä tai eläintä.

Tuntoaistin ja kosketuksen välityksellä voimme saada monenlaista tietoa kohteen fyysisistä ulottuvuuksista. MacLean [2000] sanookin, että koskettamalla voidaan lisäksi esimerkiksi arvioida kohteen dynaamisia ja materiaalisia ominaisuuksia, valvoa meneillään olevaa toimintaa, rakentaa käsitteellisiä malleja järjestelmän näkymättömistä osista ja tehdä johtopäätöksiä muista ihmisistä.

Usein ihminen käyttää tietoisesti vain näköaistiaan suorittaessaan erilaisia tarkkuutta vaativia tehtäviä, katse on tuolloin tiukasti tekemisen keskustassa, mutta samalla kuitenkin tuntoaisti kertoo meille tarkasti, mitä seuraavaksi kannattaa tehdä. Lederman ja Klatzky [1993] ovat tutkineet käsillä tapahtuvan tutkimisen luonnetta ihmisillä. Tutkimustensa pohjalta he ovat muodostaneet erilaisista esineiden tunnustelutavoista yleismaailmallisia stereotyyppisiä malleja, *tutkimusmenettelytapoja* (exploratory procedure, EP), joita ihmiset käyttävät kappaleiden erilaisten ominaisuuksien tunnustelemiseen.

Kuvassa 2.4 on kuusi erilaista tutkimusmenettelytapaa, joilla ihminen tyypillisesti tunnukselle kädessään olevaa esinettä. *Vaakasuoralla liikkeellä* (lateral motion) ihminen tutkii esineen pintarakennetta, esineen kovuutta arvioidaan ja kokeillaan *painamalla* esineen pintaa (pressure), lämpötilan aistimista varten *käsi pidetään paikallaan* (static contact) joko kiinni esineen pinnassa tai sen välittömässä läheisyydessä, esineen painoa arvioidaan yleisesti *kannattelemalla sitä kädessä* (unsupported holding) ja liikuttamalla kättä ylös-alas-suunnassa, esineen yleistä muotoa ja tilavuutta voidaan arvioida *koteloimalla* (enclosure) se kämmenen sisään, ja yleistä sekä täsmällistä muotoa *seuraamalla esineen ääri viivoja* (contour following).



Kuva 2.4. Erilaisia tutkimusmenettelytapoja ja niiden avulla esineistä saatavia ominaisuuksia. [Lederman ja Klatzky, 1993]

Goldstein [1999] sanoo, että näköaisti hallitsee tuntoaistia muotojen tunnistuksessa. Käytännökin tavallisesti pääasiassa näköaistia tunnistamaan esineen erilaisia muotoja, koska se on tuntoaistia nopeampi ja usein myös tarkempi tapa tarkastella kappaleen yleisiä fyysisiä ominaisuuksia. Kuitenkin, esimerkiksi kappaleen pinnan karkeutta tarkastellessamme, sekä näkö- että tuntoaisti vaikuttavat riittävän valoisissa olosuhteissa suunnilleen yhtä paljon kappaleen fyysisistä ominaisuuksista saamaamme havaintoon [Goldstein, 1999].

Tuntoaistin *erottelukynnys* (Just Noticeable Difference, JND) niin painon kuin muidenkin aistittavien ominaisuuksien on suhteen vakio [Goldstein, 1999]. Erottelukynnykseksi kutsutaan pienintä mahdollista eroa kahden samantyyppisen ärsyksen välillä, jonka ihminen voi aistia. Tutkimusten

mukaan erottelukynnys riippuu mittana käytetyn ärsykkeen voimakkuudesta ollen aina noin viisi prosenttia siitä; esimerkiksi sata grammaa painava kappale voidaan erottaa toisesta kappaleesta vain siinä tapauksessa, että niiden painoero on yli viisi grammaa.

2.3. Tuntopalautteen määrittely

Löyhästi määriteltynä ensimmäiset tuntopalautekäyttöliittymät syntyivät samalla, kun ensimmäinen kädellä käytettävä syöttölaite kehitettiin. Tämä sen vuoksi, että esimerkiksi niin näppäimistöillä kirjoittamisesta kuin hiiren käytöstäkin saadaan tunto- ja kehon asento- ja liikeaistien kautta palautetta, joka viestii meille syöttölaitteen toiminnasta [Akamatsu ja MacKenzie, 1996; Miller ja Zeleznik, 1998]. Miller ja Zeleznik [1998] kutsuvatkin tätä palaute-muotoa *tahattomaksi kosketuspalauteeksi* (accidental tactile feedback), koska käyttäjän näin saama palaute ei ole varta vasten tuotettua eikä sen laatu siten riipu mitenkään sovelluksesta.

Tahattomassa tuntopalauteessa käyttäjä voi tuntea painaneensa jotakin nappia tai liikuttaneensa hiirtä pöydällä, mutta saatu palaute ei kuitenkaan suoraan esimerkiksi kerro, minkä sovelluksen toiminnon käyttäjä on valinnut suoritettavaksi tai minkä kohteen päällä hiirikursori kulloinkin on. Tahattomasti saatu kosketuspalaute on siis minimaalista ja useimmiten myös täysin erilaista kuin se, jota arkitodellisuudessa koemme vastaavissa tilanteissa. Esimerkiksi käyttöliittymän rakennetta tai erilaisten kappaleiden fyysisiä ominaisuuksia tahaton palaute ei kykene kuvaamaan. Tällaista palautetta hyödyntääksemme joudumme tavalla tai toisella aina opettelemaan ja muistamaan saadun palautteen merkityksen – toisin kuin todellisessa maailmassa toimiessamme, jossa saatu palaute liittyy vahvasti suoritettuun tehtävään. On toki selvää, että myös todellisen maailman palautemallit on joskus opittu, mutta vanhoja syvällä olevia malleja kannattaa ennemminkin yrittää hyödyntää tuntopalautteen suunnittelussa kuin yrittää opettaa käyttäjille kokonaan uusia palautemalleja.

Münch ja Dillman [1997] esittävät tuntoaistin perusominaisuuksiin kuuluvan tilallisen ja ajallisen rakenteen, fyysisen kontaktin tarpeen, aistimustapojen aktiivisen tai passiivisen luonteen sekä usean eri kanavan mukana olon. Nämä kaikki erilaiset ominaisuudet vaikeuttavat osaltaan tuntopalautteen määrittelyä, joka onkin kaikkea muuta kuin yksikäsitteinen.

Burdea [1996] määrittelee tuntopalautteen jakamalla ihmiselle tuntoaistimuksia tuottavan palautteen kahteen alaluokkaan, ihon tuntoreseptoreita stimuloivaan *kosketuspalauteeseen* (tactile feedback) ja tuntoaistin lisäksi myös kehon liike- ja asentoaistiin tukeutuvaan *voimapalauteeseen* (force feedback). Burdea esittää myös näkemyksen, jonka mukaan kosketuspalaute ja *haptinen*

palaute (haptic feedback) tarkoittavat samaa asiaa. Yleisesti kosketuspalautteesta on lisäksi erotettu omaksi alaluokakseen *tärinäpalaute* (vibrotactile feedback), jota käytetään kuvaamaan lähinnä muutamien edullisten kaupallisten tuntopalautelaitteiden matalataajuuksista värinää. Tämä on hieman ristiriitaista, koska periaatteessa lähes kaikki tuntopalautelaitteet värisevät, värinän taajuus vain on osassa laitteita niin suuri, ettei ihminen kykene sitä havaitsemaan vaan aistii sen erilaisina kiinteinä pintarakenteina.

Tuntopalautteen määritelmät ja etenkin niiden käyttö on siis jo englanninkielisessä lähdeaineistossa hyvin kirjavaa, eikä syyttä, muodostuuhan tuntoaisti monikanavaisesta aistiverkostosta, ja on vielä lisäksi hyvin läheisesti kytköksissä kehon asento- ja liikeaisteihin. Tässä tutkielmassa käytetään termiä *tuntopalaute* (touch feedback) kuvaamaan yleisesti tuntoaistiin liittyvää keinotekoisesti tuotettua palautetta. *Haptinen palaute* puolestaan kuvaa monipuolista tunto-, asento- ja liikeaistien kautta saatavaa kolmiulotteista kosketuspalautetta. Haptisen palautteen rinnalla on kaksi muuta palauteryhmää, ihon kautta passiivisesti aistittava *kosketuspalaute* sekä *voimapalaute*, jonka raajojen liikuttamiseen perustuva palaute voi olla kaksisuuntaista, eli antaa käyttäjälle mahdollisuuden myös aktiivisesti ohjata tai vastustaa laitteen antamaa palautetta.

2.4. Tuntopalaute työpöytäympäristössä

Nykyiset graafiset käyttöliittymät tarjoavat käyttäjälle ylivoimaisen käytettävyyden ja vuorovaikutusmahdollisuuden niitä edeltäneisiin tekstipohjaisiin käyttöliittymiin verrattuna. Tästä huolimatta kommunikaatio ihmisen ja tietokoneen välillä on yhä ihmisten väliseen kontaktiin verrattuna alkeellista. Syötteet tietokoneelle annetaan käytännössä poikkeuksetta joko näppäimistöllä, osoitinlaitteella tai joissakin tapauksissa myös esimerkiksi puheena. Koneen käyttäjälle antama palaute puolestaan on pääosin visuaalista, joissakin tilanteissa myös ääntä. Todellisessa maailmassa ihminen kuitenkin vain harvoin tyytyy havainnoimaan lähiympäristöään pelkästään näkö- ja kuuloaistien avulla.

Ihminen saa tuntoaistimuksia kaikesta koskettamastaan riippumatta siitä, onko hänen tarkoituksensa ollut varsinaisesti tunnustella kohdetta vai ainoastaan esimerkiksi suorittaa jokin tehtävä. Tuntopalautteen lisäämisessä tietokoneen graafiseen työpöytäympäristöön täytyy täten keskittyä pohtimaan, mitä käyttäjä odottaa tuntevansa käyttäessään tietokonetta ja suorittaessaan toimintoja. Kaikki turha palaute siirtää käyttäjän huomion varsinaisesta tehtävästä havaitun ärsyksen tutkimiseen. Oakley *et al.* [2000] painottavat, että tuntokäyttöliittymien suunnittelijoiden täytyy ottaa myös tarkasti huomioon

käyttämänsä tuntopalaute laitteen valmiudet ja ominaisuudet, jotta käytetyt tuntotehosteet todella parantavat käytettävyyttä.

Kuten kohdassa 2.2 sanottiin, tuntoaisti on mekaaninen aisti, jonka avulla ihminen on sananmukaisesti kosketuksissa toimintaympäristöönsä. MacLean [2000] uskoo, että suuri osa luonnollisesta kosketuksestamme tyydyttää esteettisiä ja mukavuudellisia yllykkeitä. Hänen mukaansa tuntopalautea käyttämällä käyttöliittymiin voidaan lisätä esimerkiksi seuraavan kaltaisia ulottuvuuksia:

1. mukava kosketeltava laukaisee ensin halun koskettaa ja sen jälkeen pidellä tai sivellä kohdetta,
2. miellyttävä liike ja dynamiikka ovat parhaita, kun ne ovat samalla myös informatiivisia,
3. tuntopalaute-ergonomian tulisi ehkäistä kramppeja, puristusta, hermopainetta ja heikkojen sormien raskasta käyttöä,
4. kaksisuuntainen vuorovaikutus ympäristön kanssa antaa sisällä olemisen tunteen,
5. lihasmuistia voidaan käyttää usein toistuvien ja tuttujen tehtävien jäsentämiseen tyyliiteltyihin tai lyhenneltyihin eleisiin, ja
6. mahdollisuus personoida laitteen ominaisuuksia tekee siitä erityislaatuisen sekä kertoo laitteen omistussuhteesta ja arvosta.

Graafisten käyttöjärjestelmien työpöytäympäristö pyrkii kaikkine kansioineen, kuvakkeineen ja ikkunoineen matkimaan todellista maailmaa. Graafisessa työpöytäympäristössä yleisesti käytetty WIMP-metafora (windows, icons, menus, pointing) ei kaikesta kehityksestään huolimatta kykene täysin jäljittelemään todellisen maailman vuorovaikutustapahtumia, eikä sen sitä toki tarvitse tehdä. Tuntopalauteen kohdalla asian tulisi olla aivan samoin, emme voi olettaa, että tietokoneen ruudulla näkyvät tai näkymättömät kohteet tuntuvat todelliselta, jos niitä ei oikeasti ole edes olemassa.

McGee *et al.* [2001] sanovat, että virtuaalisen pinnan rakenne voi lisätä kohteen realismin tuntua välittämällä monenlaista tietoa esimerkiksi sen identiteetistä, tyypistä, sijainnista ja tehtävästä. Tuntopalauteen käyttö on kuitenkin usein ongelmallista, koska vain harvoin kykenemme tuottamaan aitoa maailmaa vastaavaa palautetta käyttäjälle. Tämä ero todellisen ja tietokoneen virtuaalisen maailman välillä korostuu etenkin edullisimmilla tuntopalaute laitteilla, kuten matalataajuuksista tärinäpalautetta tuottavilla hiirillä, koska ne kykenevät tuottamaan vain hyvin epätarkkaa informaatiota esimerkiksi virtuaalisten kappaleiden erilaisista pintarakenteista. Tämän vuoksi tuntopalautea tulisikin työpöytäympäristössä hyödyntää enemmän

antamaan palautetta erilaisten tilojen muutoksista ja tapahtumista kuin yrittää suoraan siirtää todellisen maailman ominaisuuksia ja piirteitä tietokonesovel-
lukseen. Oma lukunsa on totta kai erilaiset virtuaalitodellisuudet, joissa luonnollisesti pyritään kaikin keinoin mahdollisimman luonnolliseen vuorovaikutukseen tietokoneen avulla luodun keinotekoisen vuorovaikutus-
ympäristön kanssa.

Christian [2000] sanoo graafisten käyttöliittymien tulon olleen suuri huolenaihe sokeiden tietokoneen käyttäjien joukossa, koska suoraikäyttöisiä graafisia käyttöliittymiä on perusluonteensa vuoksi mahdoton käyttää ilman visuulista palautetta. Eräs tuntopalautteen käytön suurimmista hyödyistä onkin, että siitä hyötyvät etenkin hyvin suunniteltuna ja toteutettuna suoraan myös vammaiset käyttäjät – ylimääräinen palautekanava antaa kaikille käyttäjille mahdollisuuden saada enemmän informaatiota entistä havainnollisemmin. Toinen yleisesti tunnustettu etu tuntopalautteen käytölle on, että sen avulla voidaan huomattavasti parantaa ihmisten sosiaalista kommunikointia erilaisissa tietokonevälitteisissä tilanteissa [MacLean, 2000].

Useat eri tutkimukset ovat osoittaneet, että tuntopalautteen käyttö parantaa merkittävästi käyttäjän suorituskykyä ja tyytyväisyyttä monissa työpöytäympäristölle oleellisissa tehtävissä, kuten yksittäisten kohteiden valintatilanteissa [Dennerlein *et al.*, 2000; Akamatsu ja MacKenzie, 1996; Rosenberg ja Brave, 1996], ohjaamis- ja valintatehtävissä [Dennerlein *et al.*, 2000] sekä useamman kohteen valinnassa [Oakley *et al.*, 2001]. Tutkimuksissa käytetyt tuntopalautelaitteet ovat kuitenkin olleet joko yksittäisiä prototyyppisiä tai liian kalliita tavallisen tietokoneen käyttäjän hankittavaksi. Tämän vuoksi onkin ollut tarpeellista kehittää halvempia, mutta silti ominaisuuksiltaan riittäviä ja helppokäyttöisiä tuntopalautelaitteita, joita jokaisen tietokoneen omistajan on mahdollista hankkia itselleen. Edullisten tuntopalautelaitteiden ongelmana on kuitenkin usein niiden tuottaman palautteen heikko laatu, jonka avulla on lähes mahdotonta yrittää jäljitellä todellisesta maailmasta tuttuja tuntoaistimuksia. Oakley *et al.* [2000] kuitenkin esittävät, että tärkeintä tuntopalautteen lisäämisessä työpöytäympäristöön ei aina välttämättä olekaan sen tuomat parannukset suoritusaikoihin ja -tarkkuuteen vaan ne positiiviset kokemukset, joita ihmiset lisääntyneen palauteinformaation kautta saavat.

Raisamo [1999] sanoo, että kaksikäätisyyden on ehdotettu tekevän suoraikäätisyydestä entistäkin suoraikäätisempää tuoden sovelluksen käytön lähemmäksi todellisen maailman toimintaa. Myös tuntopalautteen voidaan perustellusti esittää lisäävän sovellusten suoraikäätisyyttä, tekevän siitä entistäkin suoremman ja todenmukaisemman käyttää. Oikein suunniteltuna tuntopalautteen voikin sanoa lisäävän suoraikäätisyyttä lähentäen etäisyyttä

niin järjestelmästä päämäärään kuin päämäärästä järjestelmään. Toisin sanoen, tuntopalautteen avulla käytetyt metaforat saadaan elävämmiksi; esimerkiksi jonkin kohteen raahaamisen kuvaamiseksi käytetty tärinä- tai voimapalautteen avulla on mahdollista kertoa käyttäjälle paitsi itse raahaamisoperaatiosta niin myös vaikkapa siitä, kuinka suuri tiedosto on kyseessä.

Viime vuosina onkin vähitellen herännyt laajempi kiinnostus tuntopalautteen hyödyntämisestä osana jokaisen tietokoneenkäyttäjän arkea. Ensimmäinen sysäys tähän oli Immersionin [Immersion, 2002] vuonna 1996 julkistama I-Force-teknologia, joka nykyään on uudelleennimetty TouchSense-teknologiaksi. TouchSense on tuntopalautteen teollisuusstandardi, jonka avulla on mahdollista esimerkiksi integroida tuntopalaute osaksi Microsoft Windows-käyttöliittymää tai helposti ja kätevästi lisätä tuntopalautetta erilaisiin peleihin, omiin sovelluksiin tai verkkosivuihin.

2.5. Tuntopalautelaitteita

Erilaisia tuntopalautelaitteita on kehitelty jo vuosikymmeniä. Minsky *et al.* [1990] esitteli ensimmäisen realistisen työpöytäympäristökäyttöön soveltuvan *tuntopalauteohjaussauvan* (tactile and force feedback joystick), jolla oli mahdollista tuntea ja erotella toisistaan eri karkeuksisia hiekkapaperipintoja. Ohjaussauva oli osa Sandpaper-systeemiä, ja tietävästi myös ensimmäinen todellinen voimapalautteen avulla tuntoaistimuksia tuottava laite. Nykyisin erilaisia lähinnä pelikäyttöön tarkoitettuja voimapalauteohjaimia on kaupallisilla markkinoilla runsaasti, mutta Minskyn ohjaussauvan tasoista tuntopalautetta ne eivät kykene kovinkaan tehokkaasti tuottamaan.

Seuraavaksi esittelen kolme erityyppistä tuntopalautelaitetta: kaksi hiirtä, joista toinen tuottaa sekä voima- että tärinäpalautetta ja toinen pelkkää tärinäpalautetta, sekä haptista palautetta tuottavan laitteen.

2.5.1. Voimapalautehiiri

Lähes täysin peli- ja viihdekäyttöön tarkoitettujen erilaisten voimapalauteohjaussauvojen ja -rattien jälkeen ensimmäinen kuluttajien saataville tullut edullinen tuntopalautetta antava syöttölaite oli Logitech-yhtiön vuonna 1999 esittelemä WingMan Force Feedback Mouse -niminen voimapalautehiiri [Logitech, 2002]. Kuvassa 2.5 näkyvä hiiri mahdollisti yhdessä Immersionin TouchWare Desktop -sovelluksen kanssa tuntopalautteen liittämisen osaksi kotitietokoneen Windows-käyttöjärjestelmän työpöytää, ja se olikin alun perin tarkoitettu tehostamaan juuri työpöytäympäristön käytettävyyttä. Hiiren valmistuslisenssin ostanut Logitech alkoi kuitenkin markkinoida voimapalautehiirtä ainoastaan pelikäyttöön, koska sen voima- ja tärinätehosteiden ajateltiin olevan erinomaisia rikastamaan tietokonepelejä.

Voimapalautehiiri mahdollistaa yhtä aikaa sekä voima- että värinäpalautteen käyttämisen tietokonesovelluksissa, minkä vuoksi hiiri on hintaluoksaan edelleen ainoa vartenotettava monipuolisen tuntopalautteen käytön tietokoneen työpöytäympäristössä mahdollistava laite. Hiiren avulla on mahdollista jäljitellä todellisen maailman fyysisiä lakeja ja ominaisuuksia, kuten vaikkapa painovoimaa, erilaisten kohteiden painoa sekä kappaleiden kitkaominaisuuksia. Eräs huono puoli voimapalautehiiressä kuitenkin on sen toimintaa rajoittava alusta, johon hiiri on kiinteästi kytketty voimien tuottamista varten.



Kuva 2.5. Logitechin Wingman Force Feedback Mouse -voimapalautehiiri.
[Logitech, 2002]

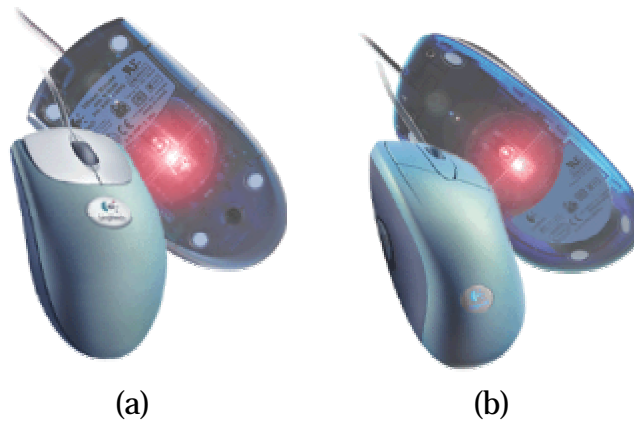
Voimapalautehiiri ei lopulta saavuttanut kovinkaan laajaa suosiota oheislaitemarkkinoilla vaikka on yhä tälläkin hetkellä monipuolisin saatavilla oleva tuntopalauttehiiri. Logitech itse vauhditti voimapalautehiiren katoamista markkinoilta julkistamalla vuonna 2000 kaksi uutta värinäpalautetta tuottavaa hiirtä. Pian tämän jälkeen voimapalautehiiri tuomittiinkin kaupallisesti epäonnistuneeksi tuotteeksi, ja yhtiö ilmoitti vuonna 2001, ettei aio tukea hiirtä enää uusissa WingMan-ajuri versioissaan.

Dennerlein *et al.* [2000] ovat tutkineet voimapalautehiiren käyttöä työpöytäympäristössä. Tutkimuksissa kävi ilmi, että liikeajat voimapalautteen kanssa olivat sekä kohteenvalinta- että ohjaus- ja kohteenvalintatehtävissä merkittävästi parempia kuin ilman voimapalautetta. Dennerlein *et al.* ehdottavatkin, että voimapalautehiirellä on hyvät mahdollisuudet parantaa työpöytäympäristön tehokkuutta.

Myös Yu ja Brewster [2002] ovat käyttäneet voimapalautehiirtä tutkimuksissaan ja osoittaneet, että sitä voidaan tehokkaasti käyttää hyväksi kuvaamaan kaksiulotteisia graafeja. Tutkimuksen mukaan hiiren suorituskyvyssä ei ollut visuaalisen ja äänipalautteen rinnalla käytettynä merkittävää eroa hiirtä huomattavasti kalliimpaan haptista palautetta tuottavaan laitteeseen verrattuna.

2.5.2. Tärinäpalautehiiri

Nykyisin jo neljä suurta hiirenvalmistajaa, Logitech, Saitek, HP ja Belkin, ovat tuoneet markkinoille omat versionsa tärinäpalautehiirestä – unohtamatta tietenkään Kensingtonin tärinäpalautea tuottavaa ohjauspalloa. Tässä keskitytään kuitenkin markkinoille ensimmäisinä tulleisiin Logitechin tärinäpalautehiiriin, kuvassa 2.6 esitettyyn perusmalliseen iFeel Mouseen (kuva 2.6a) ja oikeakätisille käyttäjille voimakkaasti muotoiltuun iFeel MouseManiin (kuva 2.6b). Kaikki esitetyt ominaisuudet pätevät kuitenkin yhtäläillä muiden valmistajien vastaaviin tuotteisiin.



Kuva 2.6. Logitechin iFeel Mouse (a) ja iFeel MouseMan (b) -tärinäpalautehiiret. [Logitech, 2002]

Tärinäpalautehiiret olivat markkinoille tullessaan syrjäyttämänsä voimapaalautehiirtä edullisempia ja myös enemmän perinteiseen työpöytäkäyttöön suunniteltuja. Hiiren tärinäpalaute tuotetaan sen sisällä olevan pienen moottorin avulla, jonka pyörimisnopeutta voi portaattomasti hallita ja jota voi myös kallistaa hieman erituntuisten palautetehosteiden tuottamiseksi.

Tärinäpalautehiiren tuottama palaute mahdollistaa lähinnä vain erilaisten yksittäisten tärehdysten, tärinän ja värinän käyttämisen. Tärinähiiren yksinkertainen ja epätarkka palaute soveltuukin käytettäväksi lähinnä antamaan melko yleisen tason lisäinformaatiota, joka tukee sovelluksen visuaalista palautetta. Tärinäpalauteen avulla voi esimerkiksi huomauttaa käyttäjää sovelluksen tapahtumista ja yrittää jollakin tasolla kuvata sen toimintoja. Hiirtä onkin käytetty jo jonkin verran viihdeteollisuudessa tuottamaan esimerkiksi erilaisia pelikokemusta elävöittäviä tehosteita tietokonepeleissä ja vierailijan mielenkiintoa herättäviä tehosteita verkkosivustoissa.

Tärinäpalautehiiren suosio näyttäisikin olevan vakaassa nousussa ja siitä tulee ensimmäinen tuntopalaute-laite, joka on yleisesti käytössä kotitietokoneiden työpöytäympäristössä. Tärinäpalautehiiren käytöstä ei

kuitenkaan ainakaan vielä ole olemassa tutkimuksia, joten mitään varmaa sen käytön mahdollisista eduista ei voida sanoa.

2.5.3. PHANToM

PHANToM (personal haptic interface mechanism) on kuuden vapausasteen haptinen syöttölaite, joka pystyy tuottamaan tarkkaa haptista palautetta [Massie ja Salisbury, 1994; Massie, 1996]. Ensimmäinen versio laitteesta kehitettiin vuonna 1994 MIT Touch Labissa [MIT, 2002], minkä jälkeen SensAble Technologies [SensAble, 2002] on jatkokehittänyt sitä yhdessä MIT:n tutkijoiden kanssa. PHANToMin avulla pystyy tuntemaan esimerkiksi hyvin aidon oloisesti erilaisia pintamateriaaleja ja kappaleiden fyysisiä ominaisuuksia, minkä vuoksi laite onkin ominaisuuksiltaan ylivoimainen esimerkiksi Logitechin valmistamiin kaksikulotteista ja epätarkkaa tärinä- tai voimapalautetta tuottaviin hiiriin nähden. PHANToM on samalla kuitenkin Logitechin hiiriä huomattavasti kalliimpi, mutta silti monipuolisuutensa sekä samantasoisiin kilpailijoihinsa nähden erinomaisen hinta-laatu-suhteensa ansiosta varteenotettava laite myös tavalliseen työpöytäympäristöön.



Kuva 2.7. SensAble Technologiesin PHANToM Desktop -haptista palautetta antava tuntopalautelaite. [SensAble, 2002]

PHANToMin tuottama monipuolinen haptinen palaute on välillä hämmästyttävänkin todellisen tuntuista, vaikka todellisen maailman suoran kädellä tunnuksilemisen sijaan palaute aistitaan epäsuorasti laitteen ohjainkynän välityksellä. PHANToMia on kuitenkin onnistuneesti käytetty esimerkiksi erilaisissa muovailu- ja lääketieteellisissä sovelluksissa, joissa vaaditaan äärimmäisen tarkkaa ja mahdollisimman luonnollista tuntopalautetta. Tämän hetkisen hintansa puolesta (noin 16.800 euroa) laite ei kuitenkaan ole kaikkien saatavilla, mutta se saattaa olla esimerkkinä lähitulevaisuudessa kehitettävälle edullisemmille tuntopalautelaitteille.

2.6. Yhteenveto

Tuntoaisti on ihmiselle usein eräällä tavalla näkymätön aisti, koska sen tuottama palaute vahvistuu lähes poikkeuksetta joko näkö- tai kuulopalautteella. Tästä huolimatta tuntoaisti on meille elintärkeä aisti, emme osaa edes kuvitella, miltä maailma sananmukaisesti tuntuisi ilman tuntoaistia. Se esimerkiksi auttaa meitä toimimaan vuorovaikutuksessa meitä ympäröivän maailman kanssa, estää meitä loukkaamasta vahingossa itseämme ja kertoo meille kehomme eri tiloista. Tästä kaikesta huolimatta tuntopalautetta ei ole vielä kovinkaan laajalti hyödynnetty tietokonemaailmassa palautemodaliiteettina. Syynä tähän on sekä laitteiden kehittymättömyys että omalta osaltaan myös saatavilla olevan tutkimustiedon puute.

Tuntopalautetta ja sitä tuottavia laitteita on jo pitkään ollut olemassa, mutta mitään laajempaa ja yleistä käyttökohdetta niille ei vielä ole ilmaantunut. Alalla tehdyissä tutkimuksissa on tähän mennessä käytetty pääosin joko yksittäisiä prototyyppisiä tai laitteita, jotka maksavat vielä nykyisin aivan liikaa yleistyäkseen. Hyvälaatuinen tuntopalaute onkin jo vuosikymmeniä ollut tiedeyhteisöjen yksinoikeus, jonka on useaan otteeseen todistettu olevan hyödyllinen lisä moniin eri tarkoituksiin, mutta jota ei tästä huolimatta ole sovellettu kovinkaan laajalti. Tuntopalautteen käytössä työpöytäympäristössä suurin ongelma tuntuisi olevan se, että käyttäjät eivät ole tottuneet siihen ja etteivät etenkin markkinoilla olevilla edullisilla tuntopalautelaitteilla tuotetut ärsykkeet useinkaan vastaa arkielämässä kohtaamiamme.

Tämän tutkielman eräs tarkoitus on antaa tuntopalautteelle yksi hyödyllinen käyttötapa osana suorakäyttöistä kaavionpiirto-ohjelmasta. Tämän esimerkin avulla pyritään osoittamaan, että jo tälläkin hetkellä saatavilla olevien edullisten tuntopalautelaitteiden tuottaman palautteen avulla voidaan parantaa tavallisen työpöytäsovelluksen toimintaa ja helpottaa käyttäjän kognitiivista taakkaa. Tämä onkin tärkeää etenkin kaavionpiirrossa, jossa kaavion tehokas vuorovaikutteinen muokkaaminen vaatii todentuntuja välineitä, joiden käyttö vie mahdollisimman vähän käyttäjänsä huomiota. Seuraavassa luvussa esitellään lyhyesti kaavionkäsittelyn perusteita, minkä jälkeen pureudutaan tarkemmin erilaisiin kaavionkäsittelymenetelmiin pohjana tuntopalautteen liittämiseksi osaksi vuorovaikutteisia työkaluja.

Tuntopalautteen hyödyntämistä käytännössä tarkastellaan lähemmin tutkielman luvussa kuusi, joka käsittelee tuntopalautteen suunnittelemista sekä sen lisäämistä tässä tutkielmassa esiteltävään suorakäyttöiseen kaavionpiirto-ohjelmaan.

3. Kaavionkäsittely

Kaavionkäsittelyyn liittyvässä tutkimuksessa on keskitytty pääosin kaavion viemän tilan tai sen muodon optimointiin erilaisten laskennallisten menetelmien, sommittelualgoritmien, avulla [Herman *et al.*, 2000]. Liian usein menetelmien vuorovaikutusta käyttäjän kanssa onkin laiminlyöty tai se on unohdettu kokonaan. Tässä luvussa keskitytään pohtimaan kaavionkäsittelyyn liittyviä aiheita, painottaen etenkin visuaalisia ja ihmisläheisiä näkökohtia.

Luvun ensimmäisessä kohdassa tutustutetaan lukija kaavionkäsittelyn peruslähtökohtiin ja aihepiirin keskeiseen termistöön. Toisessa kohdassa pureudutaan tarkemmin kaavioiden luettavuuteen vaikuttaviin seikkoihin, luettavuuden eri kriteereihin ja niiden noudattamiseen sekä arvottamiseen. Lopuksi kolmannessa kohdassa esitetään yhteenveto luvun sisällöstä.

3.1. Yleistä kaavionpiirrosta

Kaavio (graph) on abstrakti sommitelma, jota käytetään tiedon mallintamiseen [Di Battista *et al.*, 1999]. Tietojärjestelmien koko ja samalla myös niiden kehittämiseen vaaditun työn määrä alkoi kasvaa nopeasti 1960-luvulla, minkä johdosta kaavioita alettiin käyttää laajemmin apuna ohjelmistojen suunnittelussa ja niiden rakenteen ymmärtämisessä ja dokumentoinnissa. Ei ole kuitenkaan olemassa yhtä universaalia kaaviotyyppeä, jolla voisi kuvata hyvin minkä tahansa tapahtuman tai mallin. Tämän vuoksi vuosien saatossa on kehitetty lukuisia erilaisia, eri tehtäviin tarkoitettuja kaaviotyyppejä, joilla kaikilla on omat sovellusalueensa, joista esimerkkeinä voidaan mainita ohjelmistotekniikka, tietokannat, tietojärjestelmät, tekoäly, lääketiede, biologia, kemia, rakennustoiminta ja kartanpiirustus [Di Battista *et al.*, 1999].

Böhringer ja Newbery Paulisch [1990] esittävät kaksi heidän mielestään merkittävintä syytä, miksi kaavionpiirto-sovellusten käyttö on parina viime vuosikymmenenä yleistynyt. Ensimmäiseksi, ihminen on yleensä parempi käsittämään tietoa, kun se esitetään kuvallisesti eikä tekstimuodossa. Tämä heidän mukaansa siksi, että tiedon rakenteelliset ominaisuudet, kuten yksitasoisuus, symmetria ja hierarkia, ovat suoraan nähtävissä hyvin piirretystä kaaviosta, kun tekstimuotoisesta esityksestä ne täytyy aina kaivaa itse esille. Toiseksi syyksi kaavionpiirto-sovellusten yleistymiseen he mainitsevat yhä tehokkaammat tietokoneet, jotka mahdollistavat yhä monipuolisempien ja suorituskykyisempien sovelluksien toteuttamisen.

Tutkimusta kaavionpiirrosta tehdään useilla eri tieteenaloilla, mukaan lukien diskreetti matematiikka, algoritmiikka, tiedonhallinta sekä ihmisen ja tietokoneen vuorovaikutus. Useat eriluontoiset sovellusalueet ja paljolti myös

jakautuneet tutkimusintressit tuottavat paljon erilaista tietoa erityyppisten kaavioiden käyttökelpoisuudesta, erilaisten algoritmien kyvystä optimoida kaavio tiettyyn ulkoasuun ja kaavionpiirtosovellusten kehittämistä aiempaa vuorovaikutteisemmiksi ja visuaalisesti monipuolisemmiksi.

Eri tarkoituksiin kehitettyjen kaaviotyyppejen toisistaan poikkeavista luonteista huolimatta yhteistä lähestulkoon kaikille niille on, että ne koostuvat aina vähintäänkin *olioista* (entity) ja näitä yhdistävistä *suhteista* (relation). Oliot kuvataan kaavioissa symboleina, yleensä pisteinä tai laatikoina, ja suhteet viivoina, jotka yhdistävät olioita kuvaavia symboleita. Nämäkin graafiset standardit voidaan kuitenkin toteuttaa eri tavoin sovellusalueen vaatimuksista riippuen; esimerkiksi matemaattiset sovellukset käyttävät usein ainoastaan suoria viivoja kuvaamaan olioiden välisiä suhteita, kun tietokantasovelluksissa on puolestaan tapana käyttää pelkästään pysty- ja vaakasuorista viivoista koostuvia ortogonaalisia yhdistelmiä.

Kaavioiden käytön mielekkyys tiedon visualisointivälineinä riippuu olennaisesti niiden luettavuudesta, kyvystä välittää kantamansa informaatio lukijalleen yksikäsitteisesti ja nopeasti. Piirrettäessä kaavioita tietokoneen avulla eräs suurimpia hidasteita mallinnusprosessin etenemiselle on kaavion sommittelun, sen ulkoasun, muokkaaminen. Böhringer ja Newbery Paulisch [1990] määrittelevät kaavion ulkoasun sen sisältämien olioiden ja suhteiden sijaintien yhdistelmäksi, joka voidaan laatia käytetystä sovelluksesta riippuen joko manuaalisesti tai automaattisesti.

Ding ja Mateti [1990] puolestaan erottavat kaksi syytä, miksi hyvännäköisten kaavioiden piirtäminen on vaikeata: kaavion ulkoasun valinta riippuu sen käyttötarkoituksesta, ja esteettinen miellyttävyys on hyvin omakohtaista; eri kulttuuritaustan tai sivistystason omaavilla on usein erilaiset esteettiset mieltymykset. Seuraava kohta tutustuttaakin lukijan lähemmin kaavioiden luettavuuteen ja siihen vaikuttaviin seikkoihin.

3.2. Kaavioiden luettavuus

Di Battista *et al.* [1994, 1999] toteavat, että tiedon visualisointi on hyödyllistä ainoastaan, jos tietorakennetta kuvaava kaavio on helposti luettava. Tällöin kaavion avulla mallinnettu tieto välittyy lukijalle nopeasti ja yksikäsitteisesti ilman tarvetta perehtyä tarkemmin kaavion kuvaaman tietorakenteen sisältöön. Hyvä kaavio siis auttaa lukijaansa ymmärtämään kuvatun tietorakenteen, mutta huonosti suunniteltu tai piirretty kaavio voi helposti sekoittaa lukijan ja johtaa tämän harhaan [Di Battista *et al.*, 1999]. Luettavuuden kriteerit pyrkivät antamaan yleisohjeita, joita noudattamalla kaavion ulkoasusta on tarkoitus saada yhdenmukainen kokonaisuus.

3.2.1. Luettavuuden kriteerejä

Kaavioiden luettavuuteen vaikuttavat eniten niiden graafiset ominaisuudet, joiden ominaisuuksia määrittellään *esteettisillä kriteereillä* (aesthetic criteria). Kriteerit määrittelevät kaavion graafisille ominaisuuksille vaatimukset, joita pyritään noudattamaan kaavion piirroksessa mahdollisimman tarkasti. Usein tietyn kaavion piirroksessa huomioitavat kriteerit riippuvat siitä, minkä tyyppistä kaaviota ollaan piirtämässä.

Kaavion luettavuuteen vaikuttavia kriteerejä määrittelevät muun muassa Sugiyama *et al.* [1981], Di Battista *et al.* [1994, 1999], Newbery Paulisch [1993] sekä Coleman ja Parker [1996]. Yhteensä edellisissä lähteissä mainitaan parikymmentä erilaista esteettistä kriteeriä, joista olen koonnut taulukkoon 2.1 tärkeimmät luettavuuden kriteerit selityksineen.

Kriteeri	Selitys
Olioiden etäisyys	Oliot eivät saa olla liian lähellä tai kaukana toisistaan
Olioiden välit	Olioiden välien tulee olla yhdenmukaisia
Olioiden asettelu	Vanhempiolion tulee olla keskellä lapsiolioihin nähden, lapsiolioiden puolestaan symmetrisesti vanhempansa alla
Suhdeviivojen pituus	Suhdeviivat eivät saa olla liian lyhyitä tai pitkiä
Suhdeviivojen yhdenmukaisuus	Samantasoisten suhdeviivojen tulee olla pituudeltaan ja paksuudeltaan yhdenmukaisia
Olioiden ja suhdeviivojen päällekkäisyys	Suhdeviivojen ja etenkin olioiden päällekkäisyyttä tulee välttää
Suhdeviivojen suoruus	Mutkia tai kulmia suhdeviivoissa tulee välttää
Kaavion ala	Kaavion vaatima ala tulee minimoida
Symmetria	Kaavion symmetrisyys tulee maksimoida
Ryhmittely	Samaan kokonaisuuteen kuuluvat oliot tulee asetella lähelle toisiaan, lisäksi eri olioryhmät tulee erotella toisistaan riittävän selvästi

Taulukko 2.1. Tärkeimmät esteettiset kriteerit selityksineen.

Coleman ja Parker [1996] jakavat esteettiset kriteerit neljään ryhmään: *syntaktisiin* (syntactic), *semanttisiin* (semantic), *staattisiin* (static) ja *dynaamisiin* (dynamic) kriteereihin. Syntaktiset kriteerit koskevat kaavion fyysisiä yleisominaisuuksia, kuten sen rakennetta ja ulkoasun visuaalisia periaatteita. Semanttiset kriteerit puolestaan pyrkivät ottamaan huomioon olioidenväliset merkitysyhteydet ulkoasua laadittaessa, antaen määräyksiä esimerkiksi toisiinsa liittyvien olioiden sijoittelusta lähelle toisiaan. Staattiset kriteerit määrittävät kaavion ulkoasun aina uudelleen huomioimatta tehtyjen muutos-

ten laajuutta tai kaavion aiempaa ulkoasua. Dynaamiset kriteerit mahdollistavat kaavion joustavamman muokkauksen ottamalla huomioon kaavioon tehdyt muokkaustoimenpiteet pyrkien minimoimaan muutokset kaavion ulkoasussa.

Kaavion luettavuuden kriteerit ovat johdettavissa psykologiassa yleisesti määritellyistä hahmolaeista, joista kertovat esimerkiksi Katz [1948] ja Goldstein [1999]. Hahmopsykologian mukaan ihmiselle on luonnollista jäsentää näkökentässään näkyvät kohteet itsenäisiksi ja yhtenäisiksi kokonaisuuksiksi. Tätä ilmiötä selittääkseen hahmopsykologit ovat laatineet hahmolakeja, joiden tarkoitus on kuvata niitä periaatteita, joilla maailman hahmottaminen tapahtuu. Taulukossa 2.2 on lueteltu ja selitetty lyhyesti yleisimmin tiedossa olevat hahmolait.

Hahmolaki	Selitys
Läheisyyden laki	Lähekkäiset kohteet hahmotetaan yhteenkuuluviksi
Symmetrian laki	Mitä symmetrisempi kuvio osista muodostuu, sitä helpommin katsoja havaitsee kohteeksi koko kuvion eikä sen yksittäisiä osia
Samankaltaisuuden laki	Samankaltaiset kohteet tulkitaan kuuluviksi yhteen ja yhdeksi kuvioksi
Sulkeutuvuuden laki	Pienehköt lähes tai kokonaan umpinaiset alueet tulkitaan yhdeksi alueeksi
Yhteisen liikkeen laki	Yhdessä tai yhtäläisellä tavalla liikkuvat kohteet tulkitaan liittyvän yhteen
Pinta-alan laki	Mitä pienempi rajattu alue on, sitä helpommin se nähdään hahmona
Kokemuksen laki	Havainnot suosivat tuttuja ja yksinkertaisia muotoja

Taulukko 2.2. Yleisimmät tiedossa olevat hahmolait selityksineen. [Katz, 1948; Goldstein, 1999]

Viime vuosisadan alkupuoliskolla kehitettyjä hahmolakeja on sovellettu monissa erilaisissa yhteyksissä, koska ne tuntuvat olevan universaaleja ihmisten näköhavaintomalleja. Eräs mielenkiintoinen hahmopsykologinen tutkimustulos on, että värit ovat tutkimuksissa osoittautuneet muotoja määräävämmiksi kappaleen ominaisuuksiksi yhdenmukaisuuksia etsittäessä [Goldstein, 1999]. Esimerkiksi kaksi keskenään samanväristä mutta erimuotoista kappaletta katsotaan kuuluvan paremmin yhteen kuin samanmuotoiset eriväriset kappaleet. Tätä tulosta voisi myös käyttää nykyistä monipuolisemmin hyväksi kaavioiden luettavuutta tutkittaessa.

Kaavion luettavuuden parantamiseksi laadittuja esteettisiä kriteerejä on määritelty useita kymmeniä, mutta niiden noudattamisessa on omat ongelmansa. Usein haluaisimme laatia kaavion, joka noudattaa yhtäaikaisesti useaa esteettistä kriteeriä. Tahdomme esimerkiksi sekä optimoida kaavion viemän tilan että minimoida kulmien määrän suhdeviivoissa.

3.2.2. Kriteerien samanaikainen noudattaminen

Muun muassa Szirmay-Kalos [1994] ja Tamassia [1997] esittävät, että kaikkia esteettisiä kriteerejä ei useinkaan voida noudattaa samanaikaisesti. Tamassian [1997] mukaan tällaiset kaavionpiirron ongelmat voidaan kuvata *monitavoitteelliseksi optimointiongelmiiksi* (multi-objective optimization problem). Tämä tarkoittaa, että usean esteettisen kriteerin täydellinen noudattaminen yhtä aikaa on usein joko täysin mahdotonta tai se ei ole järkevää, koska ratkaisun laskemiseen kuluisi aivan liian paljon aikaa [Szirmay-Kalos, 1994].

Kohdatessamme tällaisen ristiriitatilanteen, jossa useamman esteettisen kriteerin yhtäaikainen täydellinen noudattaminen ei ole mahdollista, joudumme tekemään kompromissin ihanneulkoasun ja todellisen toteutettavissa olevan ratkaisun välillä. Käytännössä tämä tarkoittaa käytetystä sovelluksesta riippuen, että joko kaavion piirtoon käytetyn algoritmin tai sovelluksen käyttäjän täytyy arvottaa kriteerit, joiden mukaan kaavio järjestellään. Di Battista *et al.* [1999] mainitsevat kaksi yksinkertaista havaintoa, joihin useimmat kaavioiden piirtoon kehitetyt menetelmät perustuvat:

1. esteettiset kriteerit ovat usein ristiriidassa keskenään, ja
2. vaikka ristiriitoja ei olisikaan, on algoritmiikan ja tehokkuuden kannalta usein vaikea tyydyttää monia kriteerejä yhtäaikaisesti.

Noudatettavien esteettisten kriteerien ollessa ristiriidassa keskenään kompromisseja kaavion ulkoasun muokkaamisessa ei voida välttää. Tällöin pyritään mieluiten noudattamaan mahdollisimman hyvin niitä esteettisiä kriteerejä, jotka ovat olennaisimpia juuri kyseisen, työn alla olevan, kaavion luettavuudelle. Newbery Paulischin [1993] mukaan kulloinkin käytettävien estetiikkojen valinta riippuu ainakin piirrettävän kaavion tyypistä, piirtoon käytetystä sovelluksesta ja käyttäjän mieltymyksistä. Esteettisiä kriteerejä täytyy siis jollakin tapaa pystyä arvottamaan, valitsemaan niistä kuhunkin tilanteeseen sopivimmat. Automaattiset sommittelualgoritmit noudattavat ristiriitatilanteita kohdatessaan yleensä ennalta määrättyjä sääntöjä, jolloin käyttäjälle itselleen jää vain minimaalinen mahdollisuus vaikuttaa kaavion tulevaan ulkoasuun. Ihanteellisessa tilanteessa käyttäjä voisi aina määrätä käytetyt esteettiset kriteerit tai määrätä niistä ratkaisevimmat, mutta harvalla nykyisellä sovelluksella se on mahdollista, etenkin tehokkaasti.

3.2.3. Kriteerien arvottaminen

Dengler ja Cowan [1998] ovat tutkineet ovatko kaavioiden semanttiset ominaisuudet yhdenmukaisia vai satunnaisia, ja jos ne ovat yhdenmukaisia niin mitä ne ovat. Tutkimuksessa olleilla ryhmillä, kokeneilla ohjelmoijilla ja vähän tietokonetta käyttäneillä, ei havaittu eroja kaavioiden tulkitsemisessä. Tutkimuksen mukaan tärkein semanttinen ominaisuus on, että ihmiset katsovat kaavioita hierarkkisesti. Dengler ja Cowan kokosivat tutkimustuloksistaan viisi sääntöä, joita useimmat tutkimuksen koehenkilöistä käyttivät kaavioiden tulkitsemisessä:

1. Oliot aseteltuna symmetrisesti, ympyrän muotoon, linjaan tai ruudukkoon, tulkitaan ominaisuuksiltaan samankaltaisiksi tai asemaltaan yhtäläisiksi.
2. Keskeisesti tai kaavion yläosaan sijoitetuilla olioilla tulkitaan olevan erityisiä ominaisuuksia tai niitä pidetään asemaltaan muita korkeampina.
3. Linearisesti järjestetyt oliot otaksutaan sarjaksi alkaen vasemmalta oikealle tai ylhäältä alas.
4. Suhdeviivoja käytetään kaavioiden tulkinnan määrittämiseen.
5. Minkä tahansa yllä olevan piirteen puuttuminen tulkitaan semanttisesti kyseisen ominaisuuden puuttumiseksi.

Denglerin ja Cowenin kokoamat säännöt tuntuisivat sopivan melko hyvin yleistettäväksi erilaisten kaavioiden tulkinnasta puhuttaessa. Eräs lähtökohdalaisesti vaikea seikka tosin on kaavioiden hierarkkisuus, joka on suuresti kulttuurista ja osin myös yksilöllisistä tottumuksista riippuvainen seikka. Lisäksi säännöt eivät ota lainkaan huomioon olioiden etäisyyksiä, jotka ovat nähdäkseni yksi tärkeimmistä kaavion semanttisista ominaisuuksista. Esitely lista ei kuitenkaan Denglerin ja Cowanin [1998] itsensäkään mukaan ole täydellinen vaan tarvitsee tuekseen kokeellisia jatkotutkimuksia. He jatkavat todeten, että havaitsemisen kannalta tehokkaan ulkoasun luominen automaattisesti vaatii kuitenkin menetelmiä, jotka noudattavat selvästi nykyistä monimutkaisempia asettelusäännöstöjä.

Purchase *et al.* [1997] sekä Purchase [1997] ovat tutkineet esteettisten kriteerien arvottamista, jotta kaavionkäsittelyyn tarkoitettujen sovellusten tekijät voisivat perustellusti päättää eri kriteerien käytöstä ristiriitatilanteissa. Molemmat tutkimukset selvittivät kaavioiden syntaktisten ominaisuuksien vaikutusta sen luettavuuteen, eivät kaavion semanttisen merkityksen välittymistä lukijalle. Tutkimusten perusteella tärkeimmät kaavioilta halutut ominaisuudet tiedon visualisoinnissa ovat ulkoasun selkeys ja kaavion luettavuuden

nopeus, jotka ovat vahvasti riippuvaisia toisistaan. Selkeyteen päästään parhaiten minimoimalla ensisijaisesti toisensa leikkaavien suhteiden määrä ja toissijaisesti suhdeviivojen kulmien määrä. Kaavion nopeaan yleishahmottamiseen puolestaan päästään maksimoimalla kaavion symmetria, johon liittyy läheisesti olioiden välisten etäisyyksien huomioonottaminen ja muodon säännöllisyys. Kaavion suorakulmaisuuuden ja yhdysviivojen kulman muutos puolestaan ei ole merkittävä kummassakaan tapauksessa.

3.3. Yhteenveto

Yhä useammin ihmiset piirtävät kaavioita toisille ihmisille mallintaakseen ja selittääkseen jonkin monimutkaisen rakenteen. Tiedon visualisoinnissa kaaviomuotoon on tärkeää, että kaaviot ovat mahdollisimman luettavia ja tarkkoja. Esteettiset kriteerit ovat ohjeistoja, joita noudattamalla voidaan laatia syntaktisesti ja semanttisesti onnistuneita kaavioita. Erilaisia kriteerejä kaavion ulkoasulle on kuitenkin määritelty useita kymmeniä ja vain harvoin voidaan useaa eri kriteeriä noudattaa yhtäaikaisesti. Tällöin on tehtävä kompromisseja kaavion ulkoasun muodon ja selkeyden suhteen, jolloin tärkeimmät käytetyt kriteerit on valittava ja arvoitettava.

Käytännössä voi olla kuitenkin usein hyvinkin vaikea päättää, mitkä kriteerit oikeastaan ovat juuri kyseiseen tilanteeseen ja etenkin kyseiselle henkilölle tärkeimpiä, ja tulisi siten olla myös muita korkeammalle arvoitettuja. Kaavionpiirron peruslähtökohtana onkin nykyään yhä useammin tuottaa kaavioita ihmisten käyttöön, jolloin on luonnollista, että helposti luettavan ja ymmärrettävän kaavion laadinnassa ihminen on usein ylivoimainen koneeseen nähden. Tämän vuoksi uusia kaavionkäsittelymenetelmiä kehitettäessä on mielestäni paras antaa käyttäjälle itselleen mahdollisimman paljon valtaa päättää ulkoasun lopullisesta muodosta – ajatus, johon tässä tutkielmassa myöhemmin esitettävät uudet vuorovaikutteiset työkalut perustuvat.

4. Nykyiset kaavionkäsittelymenetelmät

Nykyiset kaavionpiirtosovellukset antavat mahdollisuuden muokata piirretty kaavion ulkoasua muutamalla eri tekniikalla. Yleisin sovelluksissa käytetty vuorovaikutteinen kaavion ulkoasun muokkaustapa on *komentopohjaiset tasaustyökalut* (command based alignment tools), mutta myös automaattiset sommittelualgoritmit ovat yleisesti käytössä. Lisäksi kummankin edellisen rinnalla on useimmiten myös olioiden *suora-aseointi* (direct positioning).

Tässä luvussa esitellään edellä mainittujen lisäksi myös kaksi muuta kaavioiden muokkaamiseen sovellettua menetelmää, vuorovaikutteisesti käytettävät rajoitteet ja suorakäyttöiset tasaustyökalut. Luvun lopuksi esitetään lyhyt yhteenveto esitellyistä kaavionkäsittelymenetelmistä ja vertaillaan niiden ominaisuuksien soveltuvuutta kaavioiden vuorovaikutteiseen muokkaamiseen.

4.1. Automaattiset sommittelualgoritmit

Ensimmäinen kaavioiden visuaaliseen sommitteluun kehitetty algoritmi löytyy vuodelta 1963 Knuthin [1963] artikkelista jossa esiteltiin *vuokaavioiden* (flowchart) piirtoon suunniteltu algoritmi. Algoritmi oli tarkoitettu pääasiassa suurien tietojärjestelmien ohjelmoijille algoritmien visualisointi- ja dokumentointityökaluksi. Artikkelissa esitellyssä systeemissä tietokone luo käyttäjän syöttämistä parametreista järjestetyn vuokaavion sekä viitelistan, joka kuvaa kaavion olioiden ominaisuuksia.

Nykyään automaattisia sommittelualgoritmeja on olemassa hyvin sekalainen kirjo, mutta yhteistä useimmille niistä on keskittyminen yksinomaan kaavion ulkoasun optimointiin. Sommittelualgoritmi päättää jokaisen olion ja olioiden välisen suhteen sijoittamisen kaaviossa [Newbery Paulisch, 1993]. Käytännössä sommittelualgoritmien tarkoitus on järjestää kaavion ulkoasu automaattisesti tiettyjen, sille ennalta asetettujen esteettisten ja rakenteellisten vaatimusten mukaisesti. Sommittelualgoritmit olen tässä kohdassa jakanut niiden toimintaperiaatteen mukaan kahteen kategoriaan, staattisiin ja dynaamisiin algoritmeihin. Di Battista *et al.* [1994] ovat koonneet kattavan kirjallisuuskatselmuksen, johon erilaisista sommittelualgoritmeista enemmän kiinnostuneiden kannattaa tutustua.

4.1.1. Staattiset algoritmit

Kaikille automaattisille algoritmeille on jollakin tapaa määritelty esteettiset kriteerit, joita se pyrkii mahdollisimman pitkälle noudattamaan muokatessaan käsiteltävän kaavion ulkoasua. Useimmat algoritmit on tarkoitettu vain tietyn tyyppisten kaavioiden laatimiseen, jolloin ne noudattavat vain tälle kaaviotyyp-

pille olennaisia esteettisiä kriteerejä. Usean esteettisen kriteerin yhtäaikainen noudattaminen vaatii kuitenkin algoritmilta paljon, koska sen vaatimat matemaattiset mallit ovat usein hyvin monimutkaisia ja hitaita suorittaa [Szirmay-Kalos, 1994; Tamassia, 1997]. Lisäksi staattiset algoritmit eivät kykene ottamaan huomioon olioiden keskinäisiä semanttisia yhteyksiä ilman lisätietoja [Raisamo ja Niemi, 2000], eivätkä siten säilytä käyttäjän kaaviosta muodostamaa käsitteellistä mallia [Eades *et al.*, 1991; Misue *et al.*, 1995]. Juuri tästä syystä algoritmit soveltuvatkin pääasiassa kaavion syntaktisten ominaisuuksien optimointiin, eivät niinkään kaavion semanttisen luettavuuden parantamiseen.

Di Battista *et al.* [1999] sekä Böhringer ja Newbery Paulisch [1990] sanovat, että suurimpana käytännön ongelmana staattisten sommittelualgoritmien käytölle on esteettisten kriteerien subjektiivinen luonne. Algoritmien käyttämien säännösten täytyisi olla räätälöitävissä täyttääkseen erilaisten käyttäjien kaavion ulkoasulle asettamat henkilökohtaiset vaatimukset, mutta tällöinkin saavutettu ulkoasu vain harvoin vastaa odotettua. Muita ongelmia algoritmien käytössä ovat sommitteluprosessin vuorovaikutteisuuden minimaalisuus tai sen laiminlyönti kokonaan, ja algoritmien tekemien muutosten reaaliaikainen visualisointi, animointi, käyttäjälle. Lisäksi vain harvat algoritmit ottavat uuden ulkoasun laskemisessa huomioon kaavion senhetkisen ulkoasun, jolloin käyttäjän perehtyneisyys kaavioon katoaa [Böhringer ja Newbery Paulisch, 1990; Eades *et al.*, 1991; Misue *et al.*, 1995].

4.1.2. Dynaamiset algoritmit

Dynaamiset sommittelualgoritmit (dynamic layout algorithms) ovat yksi näkökulma parantaa vuorovaikutusta kaavioiden muokkaamisessa automaattisia algoritmeja käytettäessä. Dynaamisilla algoritmeilla saavutettu vuorovaikutus on kuitenkin rajoittunutta, eivätkä ne siten anna käyttäjälle todellista valtaa päättää kaavion ulkoasusta. *Kehittyvät sommittelualgoritmit* (evolutionary layout algorithms) kuuluvat myös dynaamisiin algoritmeihin ja ne oppivat käytön aikana käyttäjänsä mieltymyksistä.

Dynaamiset sommittelualgoritmit sommittelevat kaaviosta uudelleen vain sen osan, johon tehty muutos on vaikuttanut. Cohen *et al.* [1992] kuvaavat artikkelissaan yleisen kehyksen dynaamisille sommittelualgoritmeille. Kehyksen mukaan kaavion automaattisessa sommittelussa tulee esteettisten kriteerien lisäksi optimoida myös muokkausoperaatioiden ajankäyttö, joka tapahtuu ottamalla huomioon aiemmin sommiteltuun kaavioon tehtyjen muutosten ajankohta ja muutosten laajuus.

Cohen *et al.* [1992] sanovat dynaamisten algoritmien vastaavan perinteisiä sommittelualgoritmeja paremmin ihmisten ulkoasun muokkaamiselle asetta-

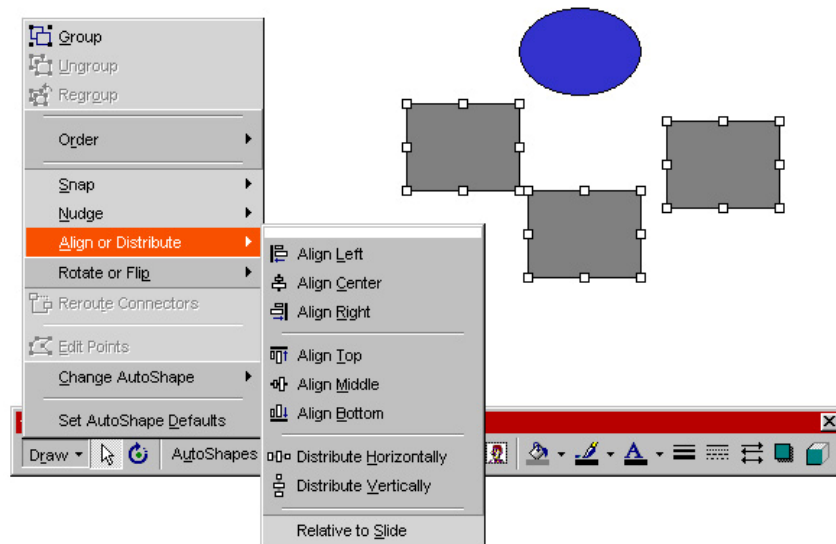
miin vaatimuksiin, koska ne kohdistavat uudelleensommittelun vain kaavion muokattuihin osiin. Szirmay-Kalos [1994] esittelee artikkelissaan algoritmin, jonka on tarkoitus etsiä kaaviosta oliojoukkoja ja järjestää ne ihmiselle luonnollisiksi kokonaisuuksiksi. Lähinnä tämä tarkoittaa erilaisten heuristiikkojen käyttämistä yhteen kuuluvien olioiden etsimiseen kaaviosta ja ulkoasun ryhmittelemistä näiden mukaan symmetrisiä lohkoja sisältäväksi kokonaisuudeksi. Valmiissa kaaviossa toisiinsa suorassa yhteydessä olevat oliot ovat ryhmiteltyinä lähekkäin ja toisiinsa kytkeytymättömät kaukana toisistaan.

Kehittyvät algoritmit puolestaan oppivat sommittelussa käytettävät esteettiset kriteerit käyttäjän mieltymyksistä, jotka määrittellään esimerkiksi esimerkiksi sommittelupareja arvottamalla ja aiemmin laadittujen kaavioiden ominaisuuksia tarkkailemalla [Masui, 1994]. Nämäkään algoritmit eivät kuitenkaan pysty ottamaan huomioon yksittäisten olioiden tai olioryhmien semanttisia yhteyksiä ja käyttäjän saavuttama vuorovaikutteisuuskin on epäsuoraa. Lisäksi valittavat esimerkkisommittelut ovat kaikki ennalta määritettyjen kriteerien mukaan laadittuja, joten käytännössä kehittyvissä sommittelualgoritmeissakin sovellus muodostaa sommittelussa käytettävän kriteeristön rajatusta joukosta käyttäjän ennen sen hetkisen kaavion piirtämistä valitsemia esteettisiä kriteerejä. Kehittyvillä sommittelualgoritmeilla on lisäksi samat ongelmat kuin muillakin ihmisen käyttäytymistä ennustamaan pyrkivillä järjestelmillä: sovellus voi olla vain harvoin täysin varma, mitä käyttäjä kulloinkin todellisuudessa haluaa tehdä. Tämä ongelma korostuu kaavionpiirroksessa, jossa kaavion merkitys voi muuttua hyvinkin paljon sovelluksen tehdessä pienen virheen päätelmissään.

4.2. Komentopohjaiset tasaustyökalut

Tasaustekniikoita, joissa ensin valitaan tasausoperaatioon mukaan otettavat oliot ja sen jälkeen suoritetaan haluttu tasauskomento valitsemalla sen tyyppi asianmukaisesta valikosta, paletista tai valintaikkunasta, kutsutaan komentopohjaisiksi tasaustyökaluiksi. Näitä työkaluja käytetään epäsuorasti joko *valikon* (menu), *paletin* (palette) tai *dialogin* (dialog) avulla. Ensimmäiset komentopohjaiset tasaustyökalut ovat yhtä vanhoja kuin kaupalliset graafiset käyttöliittymätkin. Alkuvuodesta 1981 esitelty Xerox 8010-toimistoautomaatitietojärjestelmä, lisänimeltään Star, sisälsi ensimmäisen työkalun piirrosolioiden tasaamiseksi [Bewley *et al.*, 1983]. Laajemmin tasaustyökalut tulivat tutuksi Apple Macintoshin MacDraw-ohjelman esittelemisen yhteydessä vuonna 1984 [Apple, 1984]. Molemmissa järjestelmissä olioiden keskinäisten välien ja äärirajojen tasaaminen tapahtui dialogista valittavien komentojen avulla.

Kuvassa 4.1. on esimerkki tyypillisestä valikkopohjaisesta tasaustyökalusta. Kuvassa on ensin valittu tasattavat kappaleet (harmaat nelikulmiot), minkä jälkeen valikon ”Align and Distribute” -kohdasta voidaan valita kappaleiden yhteiset vaaka- ja pystysuorat tasausreunat sekä mahdollisesti myös valittujen olioiden välien tasaus niin ikään pysty- tai vaakasuoraan.



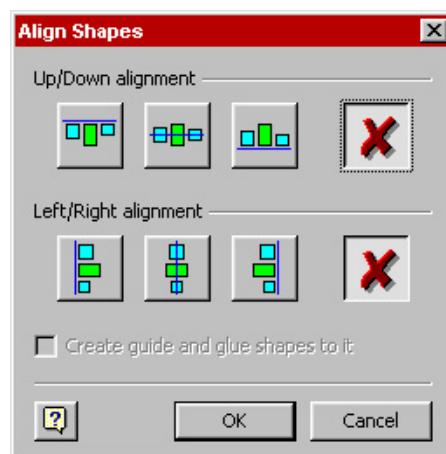
Kuva 4.1. Microsoft PowerPoint -sovelluksen tasausvalikko.
[MS PowerPoint, 1999]

Tasauksen ensimmäinen vaihe, olioiden valitseminen, on komentopohjaisilla työkaluilla suorakäyttöinen ja suoritetaan osoitinlaitteella. Toinen vaihe, komennon valitseminen ja toiminnon hyväksyminen, on kuitenkin epäsuora; käyttäjä ei näe työkalun käytöstä aiheutuvia muutoksia ennen kuin on hyväksynyt valitun tasaustavan. Tämän vuoksi käyttäjän täytyy etukäteen kuvitella haluamansa lopputulos ja yrittää löytää tätä vastaava komento miettimällä miten valitut oliot käyttäytyvät kunkin komentomahdollisuuden kohdalla.

Komentopohjaiset työkalut antavat käyttäjälle mahdollisuuden säätää valittujen olioiden sijoittelua tiettyjen ominaisuuksien perusteella. Ulkoasun säädön vapaus perustuu lähinnä valitun tasausmuodon ja olioiden etäisyyden säätämiseen. Monet kaupalliset tuotteet ovat päätyneet käyttämään dialogipohjaisia tasaustoimintoja piirrettyjen olioiden järjestelemiseen ruudulla. Ratkaisuun on todennäköisesti päädytty pääosin sen vuoksi, että nykyisten graafisten käyttöliittymien käyttäjät ovat lähes poikkeuksetta tottuneet käyttämään näitä käyttöjärjestelmien peruselementtejä, jolloin sovellusten käyttöön on helppo tottua [Raisamo ja Rähä, 1996]. Toinen mahdollinen syy näissä usein kankeissa vanhoissa ratkaisuihin on se, että sovellusten käyttäjät ovat tottuneet tiettyihin ratkaisuihin ja olleet tähänkin asti

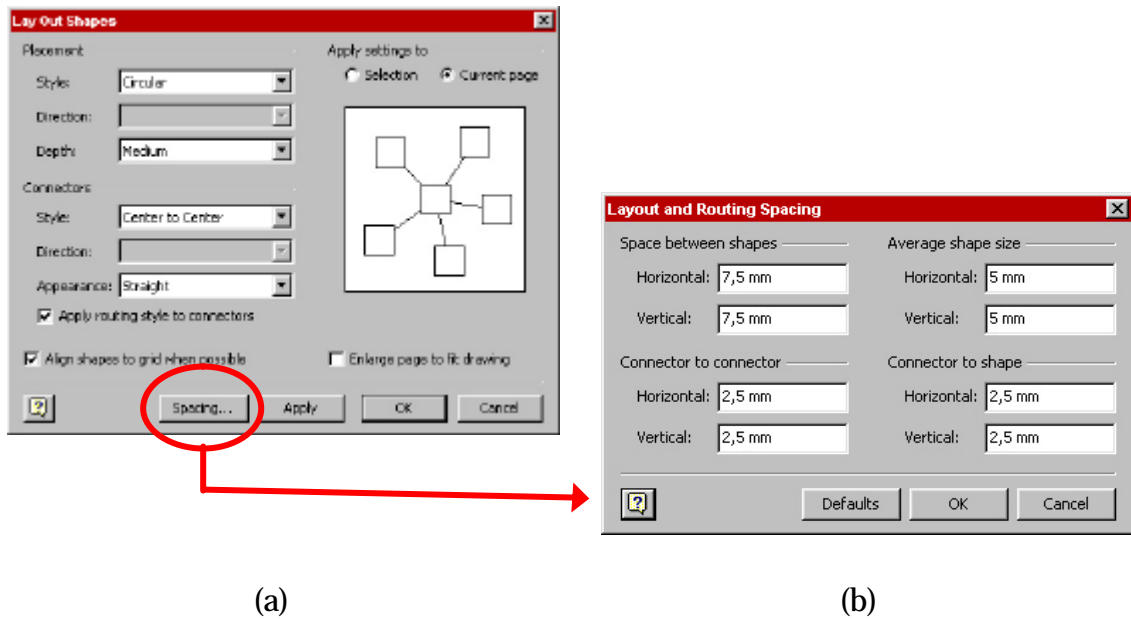
tarpeeksi tyytyväisiä niihin – ohjelmistojen valmistajilla ei siis ole ollut edes syytä ottaa riskejä uusia tuotteita toteuttaessaan.

Ensimmäiset komentopohjaiset tasaustyökalut olivat melko helppoja käyttää ja ymmärtää, tosin niillä pystyikin lähinnä vain tasaamaan valittuja kappaleita halutun reunan tai keskipisteen mukaan vaaka- tai pystysuoraan. Kuvassa 4.2 näkyy Microsoft Visio [2001] -kaavionpiirtosovelluksen tasausdialogi, jossa voi valita tasattavien kappaleiden tasausreunat vaaka- ja pystysuunnassa. Alhaalla aktivoimattomana näkyvästä valintaruudusta on mahdollisuus luoda tasausoperaatiosta *rajoite* (constraint), yleinen tasauslinja, johon voi jälkeempään lisätä uusia kappaleita. Rajoitteista kerrotaan enemmän kohdassa 4.3.



Kuva 4.2. Microsoft Visio -kaavionpiirtosovelluksen olioiden tasaukseen käytettävä dialogi. [MS Visio, 2001]

Microsoft Visio [2001] antaa myös mahdollisuuden muokata dialogipohjaisesti oliojoukon ulkoasua. Käyttäjälle tarjotaan valittavaksi kolme erilaista ulkoasutyylä, *säteittäinen* (radial), *vuokaavio/puu* (flowchart/tree) ja *kehä* (circular). Kuvassa 4.3 on olioryhmien ulkoasun muokkaamiseen käytettävä dialogi (4.3a), josta voidaan valita oliojoukon tasausmuoto, tasauksen suunta ja sen syvyys. Lisäksi dialogin alareunan "Spacing..."-painikkeesta voi avata tasaukseen liittyvien etäisyysarvojen säätämiseen käytettävän sisäkkäisen dialogin (kuva 4.3b), jonka avulla voi säätää tasattavien kappaleiden keskinäistä etäisyyttä, olioiden keskimääräistä kokoa, suhdeviivojen minimietäisyyttä toisiinsa ja suhdeviivojen minimietäisyyttä olioihin.



Kuva 4.3. Microsoft Visio -kaavionpiirtosovelluksen olioryhmien ulkoasun muokkaukseen käytettävät dialogit. [MS Visio, 2001]

4.3. Rajoitteet

Rajoitteet ovat sommittelusääntöjä, jotka määräävät olioiden sijoittelun kaaviossa. Di Battista *et al.* [1999] sanovat, että varsinaisista esteettisistä kriteereistä rajoitteet kuitenkin eroavat siinä, että ne koskevat vain jotakin tiettyä osaa kaaviosta, eivät koko kaaviota. Piirrosohjelmissa rajoitteita on käytetty aina Sutherlandin [1963] Sketchpadista lähtien, joka oli ensimmäinen vuorovaikuttainen piirto-ohjelma.

4.3.1. Snap-Dragging

Bier ja Stone [Bier ja Stone, 1986; Bier, 1989] ovat kehittäneet raahaamiseen ja rajoitteisiin perustuvan vuorovaikutustekniikan nimeltään Snap-Dragging. Snap-Dragging -tekniikassa rajoitteet kohdistuvat ainoastaan piirrettävään kappaleeseen ja katoavat heti, kun piirtotapahtuma lopetetaan. Vuorovaikutustekniikan rajoitteet perustuvat piirrettyjen kappaleiden kärkipisteiden ja niitä yhdistävien viivojen suhteisiin, joita arvioidaan piirron aikana geometrisilla kaavoilla. Tekniikka oli alun perin tarkoitettu täsmällisten kaksiulotteisten viivapiirrosten laatimiseen, mutta myöhemmin Bier [1989] laajensi sen koskemaan myös kolmiulotteista piirtoa.

Gleicher [1992] on esitellyt Snap-Dragging -tekniikkaan perustuvan Briar-nimisen piirto-ohjelman, jossa rajoitteet ovat kuitenkin pysyviä ja siten myös jälkepäin muokattavissa. Gleicher [1994] sanoo suurimmaksi esteeksi rajoitteiden käytön yleistymiselle ongelmat niiden vuorovaikutteisuudessa ja




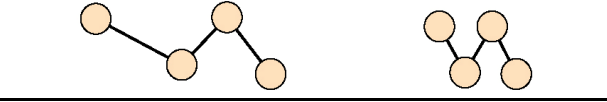

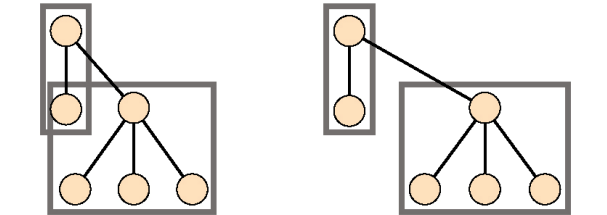
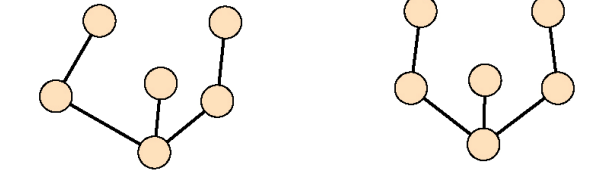
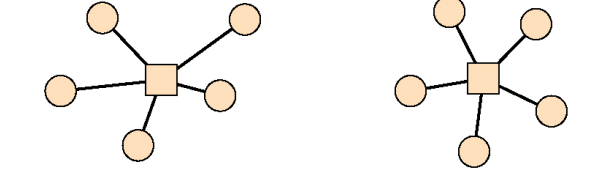
käytettävyydessä. Hänen mukaansa etenkin useampia yhtäaikaisia rajoitteita on usein vaikea visualisoida ja lisäksi niiden päällekkäisyys voi aiheuttaa käyttäjälle yllätyksiä sekä tuottaa vaikeuksia ymmärtää etenkin erityyppisten rajoitteiden yhdistämisen logiikkaa.

Hudson [1990] on puolestaan toteuttanut semantic snapping -nimisen vuorovaikutustekniikan kaavionpiirtoon. Tekniikka pyrkii päättelemään olioiden yhteyksiä ja järjestämään ominaisuuksiltaan toisiaan vastaavia olioita lähekkäin tai kiinnittämään näiden sijainnin. Semantic snapping antaa myös mahdollisuuden määrittellä olioille semanttisia ominaisuuksia, jotka otetaan rajoitteissa huomioon. Lisäksi vuorovaikutustekniikka tulostaa sanallista palautetta näytölle yritettäessä esimerkiksi muodostaa semanttisesti ristiriitaista suhdetta kaavion olioiden välille.

4.3.2. Vuorovaikutteiset rajoitteet

Rajoitteita on käytetty myös tuomaan sommittelualgoritmien käyttöön vuorovaikutteisuuksia sekä kaavion ulkoasuun vakautta. Böhringer ja Newbery Paulisch [1990] ovat esitelleet sovelluksen, jossa automaattisia sommittelualgoritmeja on sovellettu rajoitteiden käsittelemiseen. Tämän ratkaisun avulla käyttäjä voi määrittellä kaaviolle semanttisia ominaisuuksia, rajoitteita, jotka ohjaavat automaattisten sommittelualgoritmien toimintaa. Rajoitteiden avulla käyttäjä kykenee paremmin vaikuttamaan sommittelualgoritmin toimintaan ja siten kaavion ulkoasun lopulliseen muotoon. Böhringerin ja Newbery Paulischin sovellus käyttää Sugiyaman sommittelualgoritmia [Sugiyama *et al.*, 1981], joka on ensimmäisiä kaavion esteettisiä ominaisuuksia kunnioittavia ratkaisuja.

Ryall [1997] sekä Ryall *et al.* [1997] ovat esitelleet vuorovaikutteisen kaavionpiirtosovelluksen nimeltään GLIDE (graph layout interactive diagram editor), joka on tarkoitettu pienten ja keskikokoisten kaavioiden laatimiseen. GLIDE käyttää kaavion sommittelussa apunaan käyttäjän asettamia rajoitteita, joita kirjoittajat kutsuvat *näkyviksi ryhmittelypiirteiksi* (visual organization feature, VOF). Sovelluksessa on kaikkiaan kahdeksan erilaista ryhmittelypiirrettä: *tasaus* (alignment), *tasaväli* (equal spacing), *jakso* (sequence), *ryhmittymä* (cluster), *t-muoto* (t-shape), *vyöhyke* (zone), *symmetria* (symmetry), ja *keskusmuoto* (hub shape). Taulukko 4.1 kuvaa ryhmittelypiirteiden toimintaa esimerkkikuvien avulla.

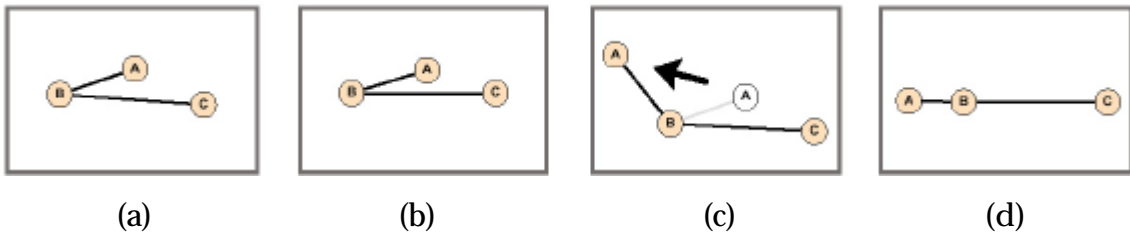
Ryhmittelypiirre	Esimerkkikuva
Tasaus	
Tasaväli	
Jakso	
Ryhmittymä	
T-muoto	
Vyöhyke	
Symmetria	
Keskusmuoto	

Taulukko 4.1. GLIDE-kaavionpiirtosovelluksen näkyvät ryhmittelypiirteet.
[Ryall, 1997]

GLIDEssa käyttäjä laatii kaavion likimääräisen sommittelun ja asettelee olioille haluamansa ryhmittelypiirteet, joiden avulla sovellus hienosäätää kaavion ulkoasuun vastaamaan käyttäjän asettamia vaatimuksia. GLIDE tukee käyttäjää paikallisia kaavion osia järjestävillä rajoitteidentyydytyskaavoilla, joiden kaavion ulkoasuun tekemät muutokset visualisoidaan animoimalla. Järjestelmän ryhmittelypiirteet ovat pysyviä ja niiden avulla käyttäjän tulisi voida vaivattomasti ja vuorovaikutteisesti sommitella kaavio itseään miellyttävään ulkoasuun.

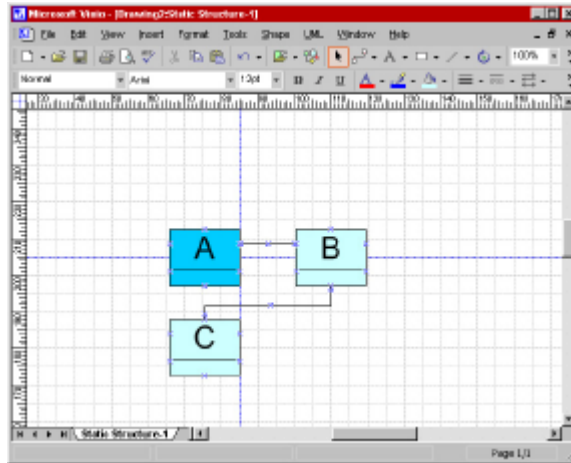
Käytännön testituloksia GLIDEn ja visuaalisten ryhmittelypiirteiden toivuudesta ei kuitenkaan ole julkaistu, joten on kyseenalaista toimivatko ne käytännössä yhtä hienosti ja jouhevasti kuin kirjoittajat esittävät. Lisäksi GLIDE ei ole rajoitteiden luonnissa suoraikäyttöinen, koska sovellus vaatii rajoitteeseen liitettävien olioiden valitsemisen yksitellen ennen kuin haluttu rajoitekomento voidaan antaa. GLIDE on kuitenkin ensimmäinen kaavionpiirto-sovellus, joka mahdollistaa käyttäjän vaikuttamaan ulkoasuun kaavion visuaalisen ryhmittelyn avulla [Ryall, 1997].

Kuvassa 4.4 näkyy esimerkki GLIDEn vuorovaikutteisten rajoitteiden toiminnasta. Kuvan (4.4a) olioihin on käytetty tasausrajoitetta (kuva 4.4b), jolloin järjestelmä on siirtänyt oliota A alaspäin ja oliota B ja C ylöspäin, jotta rajoite tulisi täytettyä. Särmien AB ja BC välisen kulman vuoksi oliota ei kuitenkaan voida tasata, minkä vuoksi käyttäjä raahaa oliota A (kuva 4.4c) vasemmalle. Tämän jälkeen järjestelmä pystyy tyydyttämään asetetun rajoitteen (kuva 4.4d).



Kuva 4.4. Esimerkki vuorovaikutteisten rajoitteiden toiminnassa GLIDEssä. Kuvan esimerkissä käyttäjä raahaa oliota A vasemmalle, jolloin sovellus voi tasata oliot A, B ja C. [Ryall, 1997]

Myös Microsoft Visio [2001] mahdollistaa kappaleiden vaaka- ja pystysuorien tasauskomentojen muuntamisen rajoitteiksi, joihin voi lisätä uusia oliota ja joiden sijaintia kaaviossa voi muokata vuorovaikutteisesti niitä raahaamalla. Rajoitteiden luonti tapahtuu valitsemalla sovelluksen tasausdialogista (kuva 4.2) kohta "Create guide and glue shapes to it". Kuvan 4.5 esimerkissä on määritelty kaksi tasausrajoitetta, joista toinen liittää oliot A ja B yhteen pystysuunnassa olioiden keskikohtien mukaan ja toinen kappaleet A ja C vaakatasossa oikeista reunoistaan.



Kuva 4.5. Esimerkki rajoitteiden käytöstä Microsoft Visiossa. [MS Visio, 2001]

4.4. Suora-asemointi

Graafisissa käyttöliittymissä suurin osa komentojen suorittamisesta tapahtuu osoitinlaitteen ja sitä näytöllä edustavan osoitinkursorin avulla. Tämän vuoksi onkin luonnollista, että graafiset kaavionkäsittelyohjelmat mahdollistavat olioiden manipuloinnin yksitellen tai ryhmänä. Suora-asemointi on vuorovaiikutustekniikka, joka tarkoittaa osoitinlaitteen kohdistimen siirtoa halutun kohteen päälle, tämän valitsemista (aktivointia) ja sen siirtämistä valittuna uuteen paikkaan. Tätä tapahtumaa kutsutaan yleisesti *raahaamiseksi* (dragging). Kaavioiden käsittelyssä raahaaminen on etenkin piirtovaiheessa kätevä tapa siirtää ja järjestellä etenkin yksittäisiä olioita piirtoalustalla. Kaavion muokkaaminen raahaamalla voidaan tehdä yleensä kahdella eri tapaa:

1. käsiteltävät oliot voi yksitellen valita ja siirtää uuteen järjestykseen, tai
2. oliot voi ensin järjestää haluttuun sommitelmaan, valita ne ryhmäksi ja siirtää ryhmänä haluttuun paikkaan.

Kummassakin edellä mainitussa tavassa olioiden asetteleminen yksittäin haluttuun sommitelmaan vaatii käyttäjältä usein paljon aikaa ja kärsivällisyyttä. Lisäksi ensimmäisessä tavassa olioita voi joutua siirtämään niin kauaksi alkuperäisestä sijaintipaikastaan, että käyttäjä kadottaa siirtämättä olevat oliot ja joutuu etsimään nämä uudelleen. Tämä tapa onkin kohtuullinen vain, jos siirrettävä matka on lyhyt. Toinen tapa soveltuu paremmin olioiden siirtämiseen pidemmälle niiden alkuperäisestä sijaintipaikasta. Huonona puolena on tekniikan vaatima uusi työvaihe, ryhmittäminen, jossa käyttäjä joutuu valitsemaan siirrettävät kappaleet yksitellen, joissakin tapauksissa antamaan vielä ryhmityskomennonkin, ennen kuin olioryhmä voidaan kokonaisuudessaan siirtää.

Olioiden valitseminen yksitellen on hidas ja tarkkuutta vaativa prosessi. Olioiden valitsemista helpottamaan onkin kehitetty erilaisia aputyökaluja, joilla voidaan ympäröidä halutut oliot ympäröimällä ne joko suorakulmiolla tai vapaasti piirrettävällä muodolla, jolloin kaikki alueen sisälle jäävät oliot tulevat valituksi. Mitään ihanteellisia ratkaisuja nämäkään eivät ole, tuovathan ne kuitenkin uuden lisävaiheen tasauseraatioon, jossa siinäkin voi sattua virheitä.

Suora-asemointi on joskus kätevä ja nopeakin tapa muuttaa yksittäisten olioiden sijaintia kaaviossa. Ulkoasun laajempaan sommitteluun ja viimeistelyyn menetelmä on kuitenkin liian hidas, työläs ja epätarkka. Lisäksi vain harvoin saavutettu lopputulos huolellisenkaan työrupeaman jälkeen täysin täyttää kaavion luettavuuden kriteerejä. Sugiyama *et al.* [1981] sanovatkin, että käsin on erittäin vaikea piirtää visuaalisesti ymmärrettäviä hierarkioita, etenkin jos nämä ovat laajoja. Kaavion ulkoasua muokatessa täytyy aina ottaa huomioon, että muokattavan kaavion ulkoasu on vaikea säilyttää tasapainoisena; yhdenkin olion siirtäminen kaaviossa uuteen paikkaan saattaa vaatia kaikkien tähän suhteessa olevien olioiden asettelemista uudelleen.

4.5. Suorakäyttöiset tasaustyökalut

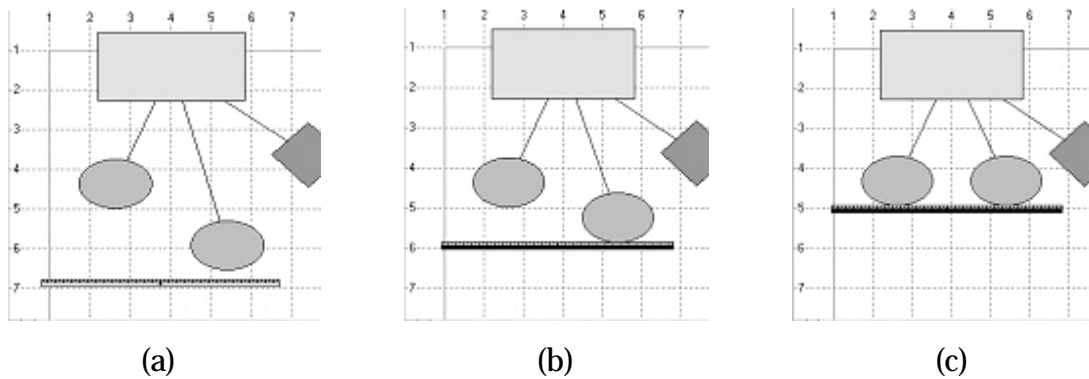
Suorakäyttöiset tasaustyökalut ovat hiirikursorina toteutettuja työkaluja, joilla voi muokata kaaviota valitsemalla suorakäyttöisesti yhden tai useamman olion osallistumaan tasauseraatioon. Tässä kohdassa on esitelty kaksi samankaltaista tasaustyökalua, *tasauspuikko* (alignment stick) ja *tasausluutatyökalu* (broom alignment tool). Kummatkin työkalut perustuvat samaan intuitiiviseen viivainmetaforaan, työkalujen toteutustavat ja toiminnallisuus poikkeavat kuitenkin hieman toisistaan.

4.5.1. Tasauspuikko

Raisamon ja Rähän [Raisamo, 1995; Raisamo ja Rähä, 1996] kehittämä tasauspuikko on suunniteltu piirtokappaleiden helppoon, nopeaan ja vuorovaikutteiseen tasaamiseen. Perinteisestä raahaamisesta menetelmä eroaa eniten siinä, että sillä voidaan käsitellä useita olioita kerrallaan valitsematta niitä erikseen. Tasauspuikko noudattaa viivainmetaforaa, jossa työkalu on visualisoitu hiirellä ohjattavalla viivainkursorilla, jonka pituutta ja suuntaa voi säätää työkalua käytettäessä. Tasauspuikko osoittaa käytännön esimerkkinä, että usean piirtokappaleen yhtäaikainen tasauseraatio on mahdollista visualisoida ruudulla ohjattavaksi työkaluksi.

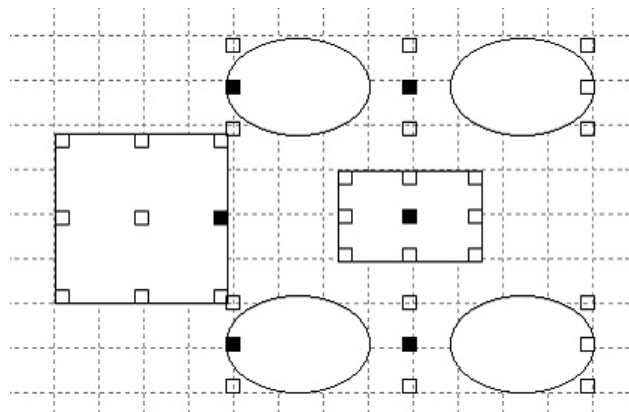
Kuva 4.6 esittelee tasauspuikon toimintaa käytännössä. Kuvan tilanteessa työkalua kuvaava kursori on siirretty tasattavien kappaleiden alle (4.6a). Tämän jälkeen se on aktivoitu painamalla hiiren nappi pohjaan ja kursoria on

liikutettu ylöspäin, jolloin se osuu ensimmäiseen olioon valiten sen mukaan tasausoperaatioon ja alkaa työntää sitä reunaansa vasten (kuva 4.6b). Lopulta työkalukursoria on liikutettu ylöspäin niin paljon, että kummatkin tasattavat oliot ovat tasattuina kursorin reunaan vasten (kuva 4.6c). Tasausoperaatioon valittuja olioita voi työkalun avulla liikuttaa vapaasti piirtoalueella, mikä mahdollistaa helpon korjauksen, jos olioita työntää työkalulla liian pitkälle.



Kuva 4.6. Olioiden tasaaminen tasauspuikon avulla. [Raisamo ja Rähä, 1996]

Perusmuodossaan tasauspuikko tasaat oliot niiden ulkoreunojen sen kohdan mukaan, johon kursori ensiksi niitä koskettaa. *Tasauspistetyökalun* avulla on mahdollista määrittellä kaavion olioille tai olioryhmille yksi tai useampia pisteitä, joiden mukaan kappaleet tasaantuvat työkalukursorin pintaa vasten. Tasauspistetyökalu määrittelee jokaiselle tasattavaksi mahdolliselle kohteelle yhdeksän tasauspistettä, yhden kohdealueen keskipisteeseen, neljä kohdealueen kulmiin ja neljä alueen sivujen puoliväleihin. Lisäksi käyttäjä voi vapaasti lisätä tasauspisteitä mihin tahansa alueen kohtaan ja poistaa niitä.



Kuva 4.7. Tasauspistetyökalun avulla valittavat tasauspisteet. [Raisamo ja Rähä, 2000]

Kuvassa 4.7 on esimerkki tasauspistetyökalun valitsemisen jälkeisestä näkymästä. Tasattavien kohteiden aktivoimattomat tasauspisteet näkyvät kuvassa pieninä läpinäkyvinä neliöinä ja aktivoituneet pisteet puolestaan mustina neliöinä. Määrittämisen jälkeen työkalua vaihdettaessa aktiiviset tasauspisteet jäävät näkyviin kertomaan käyttäjälle, miten eri kohteet käyttäytyvät niitä tasaustyökaluilla käsitellessä.

Tasauspuikko ja siitä johdettu *tasauskaari* (alignment arc) [Raisamo ja Niemi, 2000] ovat alun perin suunniteltuja minkä tahansa oliopohjaisen piirto-ohjelman tasaustyökaluiksi. Työkaluilla voi kätevästi samalla sekä valita, tasata, että liikuttaa tasattavia kohteita piirtoalustalla. Yhtenä ongelmana työkalujen käytössä on niiden liikuttaminen aktiivisena, jolloin jokainen työkalukursoriin osuva kohde otetaan mukaan tasausoperaatioon. Tämä ongelma on kuitenkin hyvin vaikea ratkaista, koska sovelluksen ei ole mahdollista tietää, onko käyttäjä tarkoittanut jonkin kohteen liitettäväksi operaatioon vai ei, ellei tätä sille erikseen ilmoiteta. Lisäksi työkaluilla tasaaminen saattaa hävittää yksittäisten olioiden väliset hierarkiat.

Olioita ryhmittämällä työkalujen avulla voi käsitellä suurempiakin kokonaisuuksia kaavioissa ja samalla tehdä niiden sisäisistä hierarkioista pysyviä. Ryhmittäminen on kuitenkin toteutettu perinteisellä tavalla, joka vaatii jokaisen olion valitsemista yksitellen ja ryhmityskomennon antamista. Eräs mahdollinen ratkaisu ryhmittämisen parantamiseen olisi käyttää *induktiivisia ryhmiä* [Olsen ja Deng, 1996], jotka sovellus muodostaisi automaattisesti käyttäjän aiempien toimintojen perusteella. Näiden avulla käyttäjän ei tarvitsisi enää tehdä vaivalloisia ryhmittelyoperaatioita, vaan sovellus huolehtisi itse niiden muodostamisesta – toki käyttäjällä pitäisi yhä säilyä mahdollisuus vaikuttaa myös itse muodostettujen ryhmien kokoon ja käyttäytymiseen.

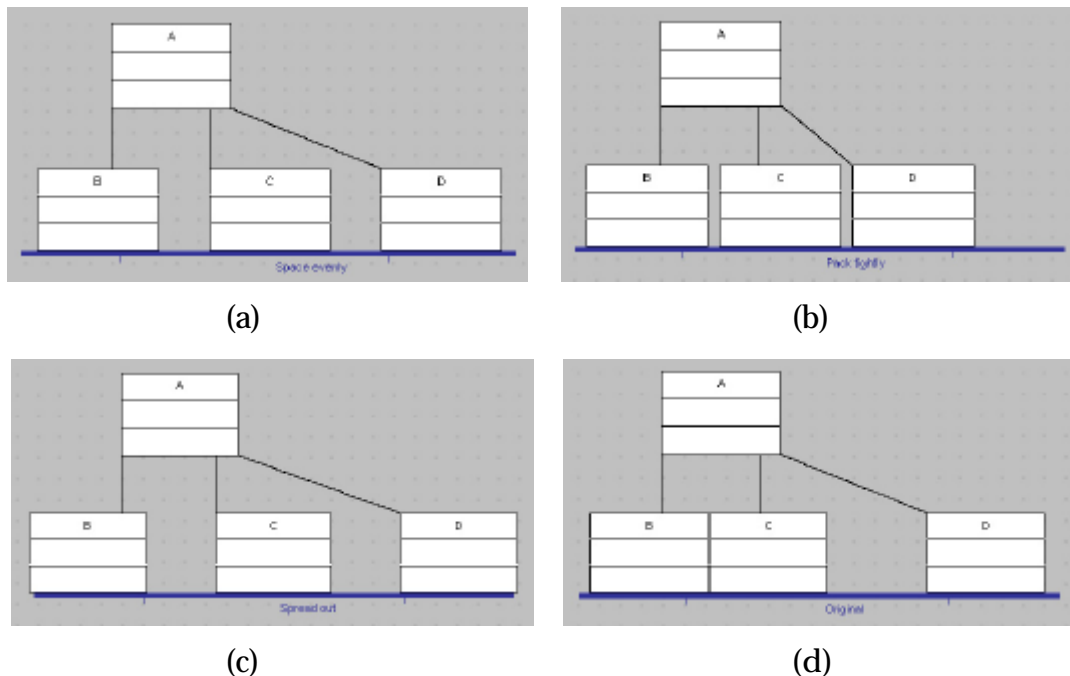
Suurin ongelma Raisamon ja Rähän [1996] tasaustyökaluissa on kuitenkin se, että ne kykenevät ainoastaan tasaamaan kohteita haluttuun linjaan tai kaareen, minkäänlaista tasavälitoimintoa työkaluilla ei ole. Tämän vuoksi esimerkiksi kaavioiden sommittelussa olennaista tasapainoisen ulkoasun kriteeriä on työlästä noudattaa, koska työkalut eivät automaattisesti tasaa valittujen kohteiden välisiä etäisyyksiä, vaan vaativat käyttäjältä tarkkuutta sekä mahdollisesti useamman eri tasausoperaation suorittamista onnistuneen ulkoasun saavuttamiseksi.

4.5.2. Tasausluutatyökalu

Robbins *et al.* [1999] ovat esitelleet tasauspuikkoa muistuttavan tasausluutatyökalun. Työkalu on osa ArgoUML-nimistä kaavionpiirtosovellusta [ArgoUML, 2001; Robbins ja Redmiles, 1999], ja sen toimintaperiaate on

samanlainen kuin tasauspuikolla: tasausluutatyökalua käytetään liikuttamalla työkalua kuvaavaa, pituussuunnassa säädettävää hiirikursoria ruudulla ja työntämällä sitä vasten tasattavaksi haluttuja kappaleita, jotka tasautuvat työkalukursorin reunaa vasten.

Erona luutatyökalussa tasauspuikkoon nähden on mahdollisuus tasata käsiteltävien kappaleiden keskinäiset etäisyydet painamalla tasaustoiminnon aktivoivaa välilyöntiä. Kuva 4.8 havainnollistaa, kuinka useammalla välilyönnin painalluksella on mahdollista valita eri tasausvaihtoehtojen välillä, joita nykyisessä versiossa on kolme erilaista: *tasaväli* (kuva 4.8a, space evenly), *tiivit välit* (kuva 4.8b, pack tightly) ja *hajautus* (kuva 4.8c, spread out). Neljäs välilyönnin painallus palauttaa oliot alkuperäiseen järjestykseen (kuva 4.8d, original).

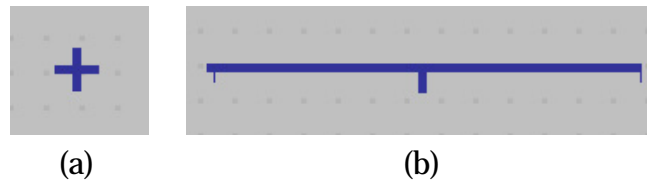


Kuva 4.8. Esimerkkejä tasausluutatyökalun käytöstä; kolme erilaista tasausvaihtoehtoa (a,b,c) ja olioiden alkuperäinen asetteluvaihtoehto (d).

[ArgoUML, 2001]

Tasausluutatyökalun tasaussuuntaa tai -kulmaa ei voi säätää tasausoperaation aikana vaan se valitaan aloitettaessa työkalun käyttö. Kuvassa 4.9 näkyy luutatyökalun hiirikursori alkutilassa (4.9a), josta se hiirtä ylös- tai alaspäin liikuttaessa muuttuu kuvan 4.9b kaltaiseksi työkalukursoriksi ja oikealle tai vasemmalle liikuttaessa vastaavaksi pystysuoraksi cursoriksi. Työkalukursorin pituutta voi kasvattaa liikuttamalla hiirtä cursorin pitkän sivun suuntaisesti, jolloin kuvan 4.9b cursorissa työkalun alareunalla näkyvät pienet hakaset liikkuvat ja levittävät työkalun reunoja. Työkalukursoria ei ole

mahdollista lyhentää, vaan tasausoperaatio on kursorin pidentämisen jälkeen aloitettava uudelleen, mikäli haluaa käyttää lyhyempää työkalua.



Kuva 4.9. Tasausluutat työkalun hiirikursori kuvattuna alkutilassaan sekä vaakasuorassa käyttöasennossaan. [ArgoUML, 2001]

Tasausluutat työkalulla voi myös peruuttaa tehdyt muutokset pitämällä se aktivoituna ja liikuttamalla työkalua takaisin tulosuuntaansa, mikäli yhtäkään neljästä edellä mainitusta tasauskomennosta ei ole vielä valittu. Sen jälkeen kun jokin tasauskomennosta on valittu, eli välilyöntiä on painettu kerrankin, työkalulla ei voi enää liikuttaa olioita vaan ainoastaan valita jokin kuvan 4.8 tasausvaihtoehdoista. Työkalulla voi täten liikuttaa valittuja olioita ainoastaan yhdessä suunnassa kerrallaan, ensin työntämällä niitä hiirikursorilla tasausuuntaan ja sen jälkeen valitsemalla niiden keskinäinen asettelu neljästä valmiista tasausvaihtoehdosta. Tämä ei kuitenkaan mielestäni ole kovinkaan toimiva ratkaisu, koska tällöin olioita ei pysty työkalun avulla siirtämään paikasta toiseen, vaikka toisaalta tällöin olioita voi tasata yhdessä suunnassa kerrallaan, mikä saattaa joissakin tilanteissa olla tarpeellista.

Tasausluutat työkalu on tasauspuikkoa rajoittuneempi kappaleiden tasauskohdan valinnassa ja tasausoperaation joustavuudessa. Lisäksi työkalukursorin koon säätämistä ja työkalun toimintaa on rajoitettu liiaksi toiminnan joustavuuden kustannuksella – käyttäjän mahdollisuus säätää tasattavien olioiden keskinäisiä etäisyyksiä ei ole kovinkaan vuorovaikutteista.

4.6. Yhteenveto

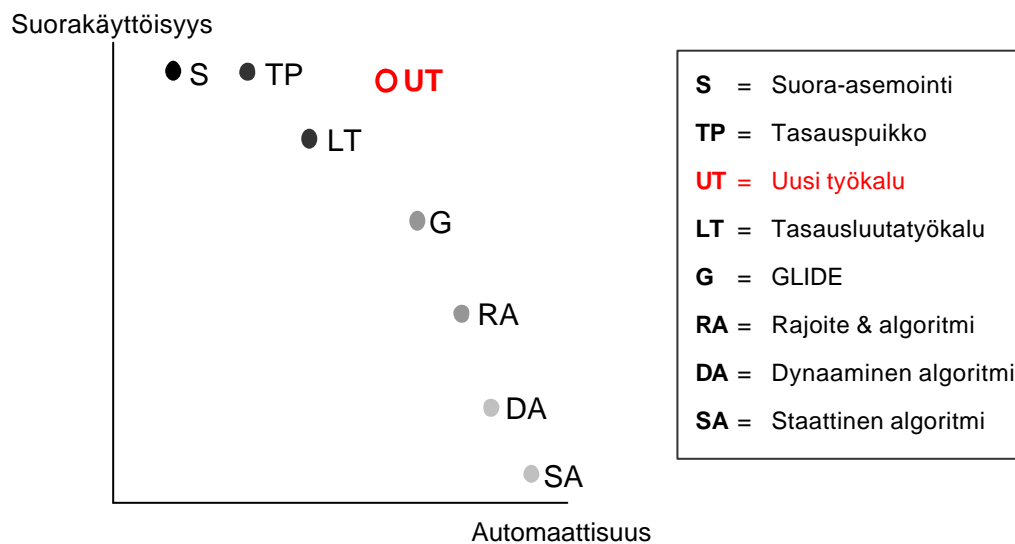
Böhringer ja Newbery Paulisch [1990] jakavat kaavionmuokkaustekniikat kahteen menetelmäperheeseen, automaattisiin ja manuaalisiin tekniikoihin. Automaattisissa tekniikoissa sovellus hoitaa käytännön työn, tai ainakin hyvin suuren osan siitä, kun manuaaliset tekniikat puolestaan vaativat käyttäjän aktiivista osallistumista muokausprosessiin.

Automaattiset sommittelumenetelmät tarjoavat käyttäjälle vain harvoin riittävän mahdollisuuden vaikuttaa kaavion ulkoasuun, esimerkiksi staattiset sommittelualgoritmit järjestävät kaavion aina uudelleen välittämättä ulkoasuun tehtyjen muutosten laajuudesta ja vaikutuksesta kaavion sommitteluun. Dynaamiset sommittelualgoritmit tarjoavat käyttäjälle jonkinasteisen

mahdollisuuden vuorovaikutteisuuden ulkoasun laadinnassa, mutta usein näin saavutettu etu on kuitenkin lähinnä näennäinen, koska nämäkin algoritmit perustuvat ennalta määritettyihin kriteereihin; käyttäjä voi vaikuttaa lähinnä vain siihen, mitä esteettisiä kriteerejä sovellus noudattaa. Kaavion ulkoasun hienosäätö ei kuitenkaan yleensä automaattisilla menetelmillä kovinkaan luontevasti onnistu.

Täysin manuaaliset sommittelumenetelmät puolestaan ovat aidosti vuorovaikutteisia, mutta vaativat käyttäjää keskittymään liiaksi yksittäisiin olioihin ja niiden aseteluun. Kaavion järjestäminen manuaalisesti on työlästä ja vaatii paljon aikaa. Lisäksi saavutettu lopputulos on vain harvoin esteettiset kriteerit täyttävä.

Kuvassa 4.10 on koottu tässä luvussa esitellyt kaavionkäsittelymenetelmät yhteen ja järjestetty ne menetelmän suorakäyttöisyyden ja automaattisuuden mukaan. Kuva on tarkoitettu menetelmien keskinäisten suhteiden vertailuun, joten pisteiden sijoittelu akseleille on ainoastaan viitteellistä. Kuvaan on vertailun vuoksi sijoitettu myös luvussa 6 tarkemmin esiteltävä kappaleiden keskinäiset välit tasaava vuorovaikutteinen kaavionkäsittelytyökalu (kuvassa UT), joka perustuu suorakäyttöiseen tasauspuikkoon lisäten siihen käyttäjää avustavia automaattisia elementtejä.



Kuva 4.10. Eri kaavionkäsittelymenetelmät aseteltuna suorakäyttöisyys-automaattisuus-akseleilla.

Suorakäyttöisyyden ja automaattisuuden yhdistäminen on vaikeaa, koska näiden kahden on kunnolla toimiakseen tuettava saumattomasti toisiaan. Ongelmana on, että kaavionkäsittelymenetelmä on joko liikaa tai liian vähän automatisoitu. Liian automatisoitu menetelmä aiheuttaa sen, että käyttäjä voi vain vähän, jos lainkaan, vaikuttaa kaavion ulkoasuun. Liika manuaalisuus

puolestaan hidastaa kaavion muokkausprosessia merkittävästi ja tekee siistin ja yhtenäisen ulkoasun luomisen vaikeaksi.

Riittävän vuorovaikutteiset, mutta samalla käyttäjää automaattisesti avustavat kaavionkäsittelymenetelmät antaisivat käyttäjälle mahdollisuuden muokata ja hienosäätää kaavion ulkoasua helposti ja tehokkaasti auttaen samalla ylläpitämään käyttäjälle kaaviosta muodostunutta käsitteellistä mallia. Tästä johtuen kaavioiden muokkaamiseen on tarpeen kehittää uusia suorakäyttöisiä työkaluja, jotka on suunniteltu huomioimaan olioiden välinen hierarkia ja kaavion ulkoasun tasapainoisuus.

5. Uudet kaavionkäsittelytyökalut

Tässä luvussa esitellään uudenlainen lähestymistapa, jossa suoraikäyttöisiin kaavionkäsittelytyökaluihin on lisätty käyttäjää helpottavia automaattisia ominaisuuksia. Luvun ensimmäisessä kohdassa kerrotaan tarkemmin tasaus-työkalujen vuorovaikutteisuudesta, minkä jälkeen toisessa kohdassa selvitetään työkalujen toimintaa. Luvun kolmannessa kohdassa asetetaan vaatimuksia uusille työkaluille. Tämän jälkeen neljännessä kohdassa esitellään neljä uutta kaavioiden vuorovaikutteiseen muokkaamiseen kehitettyä suoraikäyttöistä työkalua. Lopuksi viidennen kohdan yhteenvedossa pohditaan esiteltyjen työkalujen käytön mahdollisuuksia ja rajoituksia kaavioiden käsittelyssä.

5.1. Työkalujen vuorovaikutteisuus

Luvuissa 3 ja 4 on jo aiemmin esitelty näkökulma, jonka mukaan kaavioiden muokkaamisen tulisi olla ihmisen ja tietokoneen välinen vuorovaikutteinen prosessi, jossa käyttäjän antamien muokkaukomentojen mukaiset operaatiot päivittyvät näytölle reaaliajassa. Käytännössä tämä tarkoittaa sitä, että kaavionkäsittelyyn tulisi suunnitella suoraikäyttöisiä muokkaustapoja, jotka antavat käyttäjälle enemmän valtaa päättää piirtämänsä kaavion ulkoasusta.

Useimmissa aiemmissa lähestymistavoissa vuorovaikutteisuuden on pyritty lisäämällä automaattisiin menetelmiin käyttäjän itsensä määritettävissä olevia ominaisuuksia. Tämän periaatteen mukaan kehitetyt menetelmät eivät kuitenkaan tähän mennessä ole todistetusti tuoneet merkittäviä parannuksia, jotka tukisivat käyttäjälle kaaviosta muodostunutta käsitteellistä mallia ja auttaisivat käyttäjää muokkaamaan kaavion ulkoasua perinteisiä ratkaisuja merkittävästi vuorovaikutteisemmin.

Nykyään jo monien käyttöliittymien perusominaisuudeksi vakiintuneen suoraikäyttöisyyden avulla on mahdollista kehittää uusia kaavionkäsittelytyökaluja, jotka tekevät kaavioiden muokkaamisesta aiempaa vuorovaikutteisempää. Tehokas ja tarkka kaavioiden muokkaaminen vaatii kuitenkin käyttöliittymissä perinteisesti käytettyä raahaamistekniikkaa tehokkaampia työkaluja, joiden vaikutusalue on laajempi, mutta joiden vuorovaikutteisuudesta ja suoraikäyttöisyydestä ei kuitenkaan ole tingitty.

Tämän tutkielman kohdassa 2.4 esitettiin, että graafisten käyttöliittymien ikkunoihin, ikoneihin, menuihin ja osoittamiseen perustuva WIMP-metaphora ei kykene täysin jäljittelemään todellisen maailman vuorovaikutustapahtumia. Beaudouin-Lafon [2000] ja Beaudouin-Lafon *et al.* [2000] sanovat siirtymisen WIMP-käyttöliittymistä uusia vuorovaikutustekniikkoja hyödyntäviin niin

sanottuihin post-WIMP -käyttöliittymiin vaativan sekä uusia vuorovaikutusmalleja että niitä noudattavia työkaluja.

Beaudouin-Lafon [2000] onkin esittänyt *välineellinen vuorovaikutus* -nimisen vuorovaikutusmallin (instrumental interaction), joka laajentaa ja yleistää suorakäyttöisyyden periaatteita. Välineellinen vuorovaikutus esittää graafiset käyttöliittymät käyttäjän hallitsemina *vuorovaikutusvälineinä* (interaction instruments) sekä niiden avulla käsiteltävinä *kohdeolioina* (domain objects). Mallin mukaan vuorovaikutusvälineiden tulisi toimia samalla tavalla kuin todellisen maailman työkaluesikuviansa, joilla käsitellään fyysisesti olemassa olevia kohteita.

Jo Eades *et al.* [1991] aikanaan osoittivat, että kaavionkäsittelyyn tarvitaan menetelmiä, jotka järjestävät vain tietyn, käyttäjän valitseman alueen kaaviosta tuhoamatta samalla käyttäjän kaaviosta muodostamaa käsitteellistä mallia. Ratkaisuksi käsitteellisen mallin tukemiseen Misue *et al.* [1995] kehittivät kolme matemaattista mallia: kaavion olioiden ortogonaalisen järjestelyn sekä olioiden keskinäisten etäisyyksien ja niiden paikallisten suhteiden huomioon ottamisen. Ortogonaalisessa mallissa oliot asetellaan yksinkertaiseen ortogonaaliseen järjestykseen, joka on helppo hahmottaa. Lähekkäisyysmalli puolestaan noudattaa ohjetta, jonka mukaan toisiaan lähellä alun perin olleiden olioiden tulee myös sommittelun säätämisen jälkeen olla lähellä toisiaan. Topologinen malli ehdottaa, että muunnetun kaavion visuaalisella kuvauksella tulee olla samat paikalliset suhteet kuin alkuperäisellä kaaviolla.

Näkisin kuitenkin, että paras tapa saavuttaa tämä vaatimus on seurata Beaudouin-Lafonin [2000] välineellisen vuorovaikutuksen mallia ja kehittää vuorovaikutteisia työkaluja, joiden toiminta ja ohjaaminen ovat käyttäjän itsensä hallittavissa. Beaudouin-Lafon uskookin, että uusia vuorovaikutustekniikoita ei ole paljoakaan kokeiltu käytännön sovelluksissa, koska sovellusten kehittäjät pitäytyvät mieluummin vanhoissa tutuissa vuorovaikutustekniikoissa kuin ottavat riskin yhdistämällä uusia tekniikoita sovelluksiinsa.

Kohdassa 4.5 esitellyn tasauspuikon [Raisamo ja Räihä, 1996] kaltaiset suorakäyttöiset työkalut sopivat hyvin kaavionkäsittelyyn etenkin, kun kaaviota halutaan sommitella vuorovaikutteisesti ja olioiden semanttiset yhteydet halutaan määrittellä itse. Näitä työkaluja käyttämällä kosketus kaavion sommitteluun säilyy koko muokausprosessin ajan ja käyttäjän on työkalun suoran palautteen ansiosta helppo sommitella kaavio pala palalta häntä itseään miellyttävään muotoon – juuri työkalun käyttäjälle välittyvän informaation vuorovaikutteisuus onkin suorakäyttöisten tekniikoiden suurimpana etuna automaattisiin sommittelualgoritmeihin nähden.

Nykyisin piirto-ohjelmissa laajasti käytettyihin komentopohjaisiin tasaustyökaluihin verrattuna tasauspuikko poistaa tasausoperaatiosta yhden välivaiheen, tasaustavan valitsemisen tehden siten koko tasausoperaation suorakäyttöiseksi. Työkalujen laajan vaikutusalueen ansiosta ne puolestaan tekevät kaavion muokkaamisesta olioiden yksittäin raahaamista nopeampaa.

5.2. Työkalujen toiminta

Buxton [2001] näkee nykyisten tietokonesovellusten ongelmana niiden kattamien ominaisuuksien määrän liiallisen kasvun, jolloin ihmisen oma suorituskyky ei enää yllä tietokoneen tarjoamien ominaisuuksien tasolle. Ratkaisuna liian monimutkaisesti toteutettuun ongelmanratkaisuun Buxton esittää sovelluksen toimintojen hajauttamisen osiin, useisiin erikoistettuihin työkaluihin, joilla kaikilla on oma vastuualueensa. Työkalut eivät kuitenkaan saa olla liian monimutkaisia ja raskaita rakenteeltaan tai toiminnoiltaan vaan niiden vaikutusalue tulee rajata tarpeeksi pieneksi, jotta suorituskykyongelmat eivät häiritse työkalujen käyttöä. Työkalujen toiminnassa on täten Buxtonin [2001] ratkaisua noudattaen tehtävä selkeä ratkaisu työkalun tehokkuuden ja sen vaikutusalueen välillä.

Raisamo ja Räihä [2000] ovat tehneet kattavan empiirisen tutkimuksen, joissa he vertasivat tasauspuikkoa valikko- ja paletti käyttöisiin komentopohjaisiin tasaustyökaluihin. Tutkimus osoitti, että jopa kokemattomimmat tietokoneen käyttäjät suoriutuivat työkalun käytöstä yhtä hyvin kuin perinteisten komentopohjaisten tasaustyökalujen kanssa. Raisamon ja Räihän tulokset osoittivat myös, että tasauspuikko on käyttökelpoinen työkalu piirto-ohjelmien lisäksi myös kaavioiden käsittelyssä, mutta tarvitsee jatkokehittelyä suoriutuakseen kunnolla hieman monimutkaisemmista tasausoperaatioista; esimerkiksi useamman päällekkäisen olioiden käsittelyminen oli tasauspuikolla hankalaa kuten myös tasattavien olioiden välien saaminen yhdenmukaisiksi.

Robbins *et al.* [1999] saivat tasausluutatyökalun käytöstä tekemässään kokeessa selville, että työkalu tarjoaa käyttäjälle luonnollisemman ja suorakäyttöisemmän vuorovaikutustekniikan kuin perinteiset tasaustyökalut. Testin mukaan tasausluuta vaatii vähemmän hiiren liikuttamista ja olioiden raahaamista kuin normaaliensa tasaustyökalujen käyttö. Tulosten mukaan siis myös tasausluudan lähestymistapa näyttäisi olevan kehityskelpoinen kaavioiden vuorovaikutteiseen manipulointiin, mutta kuten kohdassa 4.5 aiemmin perustellusti todettiin, se ei kuitenkaan mahdollista käyttäjälle kovinkaan monipuolista vuorovaikutusta.

Raisamo ja Niemi [2000] ehdottavat, että kaavion karkea sommittelu voitaisiin ensin tehdä automaattisella algoritmilla, minkä jälkeen kaavion ulkoasu olisi kätevä hienosäätää suorakäyttöisellä työkalulla. Vaihtoehtoisesti kaavion

voi muokata haluamaansa muotoon alusta asti käyttämällä apunaan pelkästään työkalua, mutta tämä on ainakin suuria kaavioita uudelleen järjestettäessä työhönsä kuuloista. Suljettaessa sommittelualgoritmien käyttö pois näyttäisi siltä, että olisi järkevää yrittää sisällyttää suorakäyttöisiin tasaustyökaluihin yksinkertaisia algoritmeja, jotka helpottaisivat käyttäjän työ määrää tukien tätä valittujen olioiden järjestämisessä. Tällöin koko muokkausprosessi säilyisi vuorovaikutteisena ja olisi siten käyttäjän suoraan hallittavissa koko ajan.

5.3. Vaatimuksia uusille työkaluille

Uusien tasauspuikkoon perustuvien työkalujen erikoistaminen kaavionkäsittelyyn edellyttää niiden toiminnan ja siten niiltä vaadittavien ominaisuuksien pohtimista. Coleman ja Parker [1996] esittävät neljä ominaisuutta, jotka hyvän sommittelualgoritmin tulisi täyttää. Nämä ovat toiminnan yleisyys, käytön joustavuus, toiminnan läpinäkyvyys, niiden sopivuus käyttötarkoitukseensa ja suorituskyky. Samat ominaisuudet voidaan tietyin edellytyksin asettaa myös suorakäyttöisille kaavionkäsittelytyökaluille.

Peruslähdekohdaksi uusille työkaluille on tasauspuikon tapaan valittu pyrkimys mahdollisimman suureen suorakäyttöisyyteen ja vuorovaikutteisuuteen. Tasauspuikon yksinkertaistetusta toiminnasta poiketen työkaluihin on katsottu tarpeelliseksi lisätä myös automaattisia ominaisuuksia, joiden avulla käyttäjän työtaakkaa voidaan pienentää ja lopputuloksen laatua parantaa. Automaattisuus ei kuitenkaan saisi huonontaa työkalujen käytön intuitiivisuutta, niiden toiminnan loogisuutta ja helppoutta, vaan sen tulisi yksinomaan helpottaa esteettisesti miellyttävän ja tasapainoisen ulkoasun laatimista. Lisäksi käyttäjän tulisi tietyissä rajoissa pystyä vaikuttamaan myös työkalujen automaattisiin ominaisuuksiin toisin kuin tasausluutatyökalussa, jossa käyttäjän vuorovaikutus eri tasausvaihtoehtoihin on minimoitu.

Jotta kaavioita voidaan käsitellä tehokkaasti, täytyy kaikkien tarvittavien muokkaustapojen olla helposti saatavilla. Uusien työkalujen kehittämisessä on seurattu Buxtonin [2001] esittämää ajatusta, jonka mukaisesti kaavionkäsittelysovellukselta vaadittuja ominaisuudet on jaettu usean erikoistetun työkalun vastuuksi. Tällä tavoin on vältetty ongelmatilanne, jossa yhden monitoimisen työkalun käytön oppiminen vaatisi pitkäjänteistä perehtymistä sen taustalla piilevään monimutkaiseen toiminnallisuuteen. Täten on saatu monipuolinen valikoima erikoistettuja kaavionmuokkaustyökaluja, joiden toiminta on yksinkertaista ja keskenään samankaltaista, minkä ansiosta käyttäjän on huomattavasti helpompaa ymmärtää ja oppia soveltamaan niitä kaavion muokkaamiseen. Tällöin myös Buxtonin [2001] mainitsema tietokonesovelluksille yleinen kuilu fyysisen ja virtuaalisen maailman välillä on saatu minimoitua.

Yksinkertaisen toiminnan lisäksi myös työkalujen onnistunut visualisointi on oleellinen seikka. Käyttäjän tulisikin jo pelkästä työkalukursorin ulkonäöstä tietää, mitä käytössä olevalla työkalulla pystyy tekemään. Myös esimerkiksi työkalun eri tilat ja koon muutos tulisi olla johdonmukaisesti esitetty.

Kaavioiden muokkauksessa on otettava mahdollisimman hyvin huomioon myös kohdassa 3.2 esitellyt kaavion luettavuuteen vaikuttavat esteettiset kriteerit. Luvussa neljä esiteltyssä GLIDE-systeemissä [Ryall *et al.*, 1997] on eroteltu kahdeksan erilaista olioiden ryhmittelypiirrettä, joiden avulla kaavioiden kokonaisvaltainen muokkaaminen on mahdollista. GLIDE perustuu vuorovaikutteisiin rajoitteisiin, joita olioryhmille määrittämällä käyttäjä saa soveluksen automaattisesti järjestämään kaaviota rajoitteiden määräämien muotojen ja tärkeimpien esteettisten kriteerien mukaisesti. GLIDEn ryhmittelypiirteiden ominaisuuksia ei menetelmien luonteiden erilaisuuden vuoksi voi suoraan hyödyntää suorakäyttöisten työkalujen malliksi vaan uusille työkaluille on valittava näille paremmin sopiva lähestymistapa.

5.4. Työkalujen esittely

Tässä kohdassa esiteltävät uudet tasaustyökalut yhdistävät olioiden valitsemisen ja tasaamiseen yhtäläisin välein työkalukursorin reunaa vasten, lisäksi tasaustyökaluilla on mahdollista säätää valittujen olioiden keskinäisiä etäisyyksiä. Näiden lisäksi mukana on myös työkalu, jonka on tarkoitus helpottaa olioryhmien valitsemista ja siirtämistä piirtoalustalla. Muokkausoperaation ajaksi työkalut tavallaan ryhmittävät vaikutuspiirissään olevat oliot, mutta näin muodostetut ryhmät eivät kuitenkaan jää pysyviksi.

Työkaluista *kaappausellipsi* helpottaa toisiaan lähelläolevien olioiden siirtämistä piirtoalustalla, *tasaussellipsi* järjestää oliot symmetriseen ellipsimuotoon ja kaksi erilaista versiota *tasavälipuikosta* tasaavat olioiden välisiä etäisyyksiä joko olioiden keskipisteiden tai niiden reunojen perusteella. Seuraavissa alakohdissa esitellään jokainen työkalu tarkemmin sekä suhteutetaan niiden ominaisuudet kaavioiden muokkaamiseen.

5.4.1. Kaappausellipsi

Moni oliopohjaisten piirto-ohjelmien käyttäjä tai kaaviota tietokoneella käsitellyt on varmasti kiinnittänyt huomiota siihen, kuinka hankalaa on valita piirtoalustalta useampi kohde ja liikuttaa niitä ryhmänä. Perinteisesti tähän on vaadittu liikuteltavien kohteiden valitsemista yksitellen ja niiden ryhmittämistä kokonaisuudeksi. Tämän operaation helpottamiseksi tuntui luontevalta suunnitella työkalu, kaappausellipsi, jolla valitun kokoisen joukon

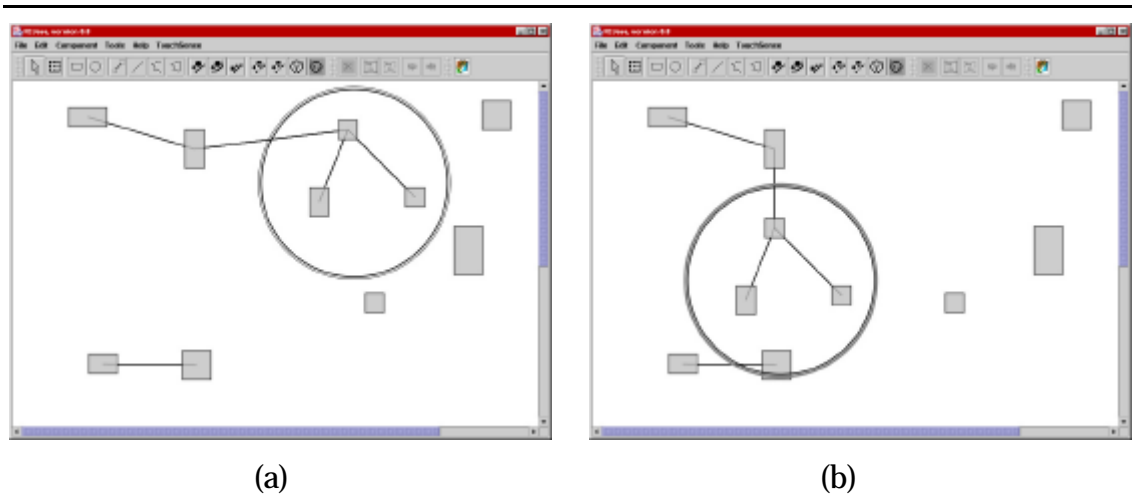
piirtoalustan olioista voisi valita kerralla ja liikuttaa kokonaisuutena, ikään kuin kaapata tai lapioida osa olioista toiseen paikkaan.

Luonnollinen kuvaus työkalun lapiometaforalle on soikea hiirikursori, jota liikutetaan näytöllä. Ellipsin muotoista työkalukursoria ei käytännön vaatimuksista johtuen kuitenkaan kannattanut toteuttaa todellisen maailman esikuvansa tapaan kiinteäksi, vaan metaforaa oli syytä laajentaa työkalun koon muokkausmahdollisuudella, kuten tasauspuikossa [Raisamo ja Rähkä, 1996] on tehty.

Työkalun toiminnallisuutta mietittäessä vaatimuksina oli luoda nopea tapa valita siirrettävä oliojoukko, työkalun ja sen vaikutusalueen selkeä visualisointi sekä työkalun vaikutusalueen yksinkertainen säätäminen. Oliojoukon valinnassa oli loogisinta päätyä valitsemaan kaikki ne oliot, jotka ovat soikean työkalukursorin sisäpuolella tai leikkaavat sen reunoja työkalu aktivoitaessa. Valitun oliojoukon keskinäiset suhteet päätettiin siirron aikana säilyttää ennallaan, koska olioiden liikuttaminen haluttiin erottaa selvästi niiden sommittelun muokkaamisesta. Oliojoukon ulkopuoliset suhteet puolestaan päivittyvät dynaamisesti joukkoa siirrettäessä.

Piirtoalustalla liikuteltavana hiirikursorina kuvatun työkaluellipsin vaikutusalueen luulisi olevan käyttäjän helposti ymmärrettävissä. Työkalun vaikutusalueen eli hiirikursorin muodon säätäminen piti suunnitella yksinkertaiseksi ja nopeaksi toimenpiteeksi, jotta käyttäjä voisi keskittyä itse asiaan, kaavion ulkoasun muokkaamiseen. Työkalukursorin muodon säätäminen on mahdollista sekä pysty- että vaakasuunnassa ja se tapahtuu *ei-dominoivalla* (non-dominant) kädellä, jolloin *dominoivaa* (dominant) kättä ei rasiteta liialla toiminnallisuudella. Myös työkalukursorin kiertäminen on yhtäläillä mahdollista, jotta siirrettäväksi halutut oliot olisi mahdollista valita riittävän tarkasti.

Peruslähtökohtana kaappausellipsin suunnittelussa oli, että liikutettavien olioiden valitseminen työkalulla on nopeampaa kuin niiden perinteinen yksitellen valitseminen ja ryhmittäminen kokonaisuudeksi. Lisäksi operaation loputtua ryhmäksi valittu oliojoukko vapautuu automaattisesti ilman erillistä purkukäskyä. Työkalun vaikutusalueen (eli kursorin koon) säädöllä ja hiiren painalluksella tehdään sama asia, mihin ennen tarvittiin lukuisia valintaosoituksia ja ryhmityskomento. Perinteiseen raahausneliöön nähden kaappausellipsi puolestaan poistaakin olioiden ryhmittämiseen ja ryhmän purkamiseen vaaditut välivaiheet.



Kuvat 5.1. Siirrettävät oliot valitaan kaappausellipsillä (a) ja oliojoukko siirretään uuteen paikkaan (b).

Kuvassa 5.1 valitaan kaappausellipsin hiirikursorilla piirtoalustalta siirrettäväksi haluttu joukko olioita (5.1a), minkä jälkeen aktiivisella työkalulla siirretään valittu oliojoukko uuteen paikkaan piirtoalustalla (kuva 5.1b). Siirron aikana työkalu säilyttää siirrettävän oliojoukon keskinäiset suhteet. Lisäksi työkalun toiminnan loputtua siirron ajaksi muodostettu ryhmä purkautuu ja siirretyt olioita voi jälleen käsitellä yksittäin.

5.4.2. Tasausellipsi

Usein kaavion sommittelun muokkaamisessa on olennaisen tärkeää pystyä järjestämään vain pieniä osia koko kaaviosta ilman, että halutun vaikutusalueen ulkopuolelle jääviin olioihin vaikutetaan. Tasausellipsi mahdollistaa valittuun olioon suorassa yhteydessä olevien olioiden tasaamisen työkalukursorin reunoille, jolloin saadaan aikaan symmetrisiä olioryhmiä.

Työkalun metafora on melko irrallaan todellisuudesta: ellipsi, jonka reunoille oliot tasataan. Työkalun suorittamaa operaatiota ei meitä ympäröivästä todellisuudesta suoraan löydy, joten reaali maailman esikuvaa on sille hankala keksiä. Työkalun elliptiseen muotoon päädyttiin lopulta toiminnallisuuden asettamilla ehdoilla, ja siksi uskonkin käyttäjien sisäistävän työkalun toiminnan sitä kokeiltuaan.

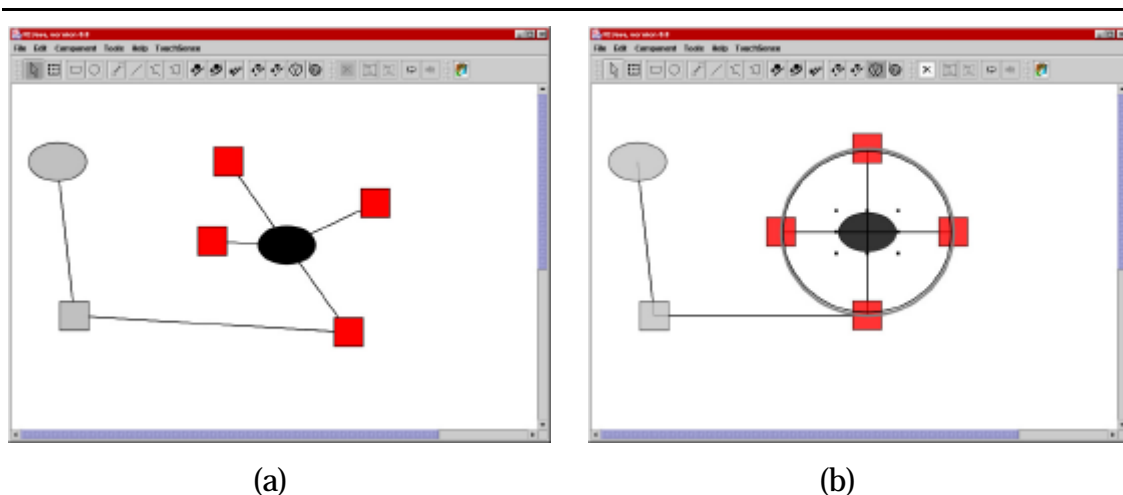
Tasausellipsin vaikutusalue päätettiin rajata valittuun olioon suoraan suhteessa oleviin olioihin, koska tällöin kaavion sommittelun muokkaaminen ja muutosten huomaaminen pysyy käyttäjän hallinnassa – liian laaja vaikutusalue saattaisi vain hämmentää käyttäjää ja tehdä työkalun käytöstä liian vaativaa.

Alun perin työkalun toiminnallisuus oli toteutettu siten, että tasaus oli yksiulotteista: tasauksessa oliot vain siirrettiin työkalun reunoille huomioi-

matta niiden keskinäisiä etäisyyksiä. Tasapainoisempaan ulkoasuun pyrittäessä on kuitenkin tärkeää, että sommittelun muutokset tekevät ulkoasusta mahdollisimman tasapainoisen kokonaisuuden. Tässä tutkielmassa valittiinkin sellainen ratkaisu, että työkalu tasaa oliot tasavälein työkalukursorin kehälle, jolloin sommittelusta tulee aiempaa tasapainoisempi ja visuaalisesti miellyttävämpi. Tasaus tapahtuu nyt siis kaksiulotteisesti, koska tasausmuodon lisäksi huomioon otetaan myös tasattavien olioiden keskinäiset etäisyydet.

Työkalun toiminnalle olennaista on, että tasausoperaation aikana voidaan työkalun muotoa muuttaa, jolloin oliot tasataan dynaamisesti työkalun muotoon mukautuen. Tämä toteutettiin mahdollisuudella säätää työkalukursorin kokoa tasausoperaation aikana. Työkalukursoria tasausoperaation aikana siirrettäessä tasattavat kappaleet liikkuvat työkalukursorin mukana, joten erillistä operaatiota tasatun olioryhmän siirtämiseen ei tarvita.

Kuva 5.2 esittää kaavioiden piirroksessa yleisen tilanteen, jossa halutaan muodostaa symmetrinen ryhmä samaan kokonaisuuteen kuuluvista olioista. Kuvan 5.2a tapaan keskeiseen olioon suorassa yhteydessä olevat oliot voidaan tasausellipsillä tasata tasavälein keskusolion ympärille symmetriseksi kehäksi (kuva 5.2b). Olioiden tasausoperaatio suoritetaan asettamalla työkalukursorin keskellä oleva risti halutun keskuskappaleen päälle ja aktivoimalla työkalu. Työkalun ollessa aktivoituna sen kursorin korkeutta ja leveyttä voidaan säätää portaattomasti, jolloin työkalu tasaa oliot dynaamisesti työkalukursorin uuden muodon mukaiseksi kokonaisuudeksi.



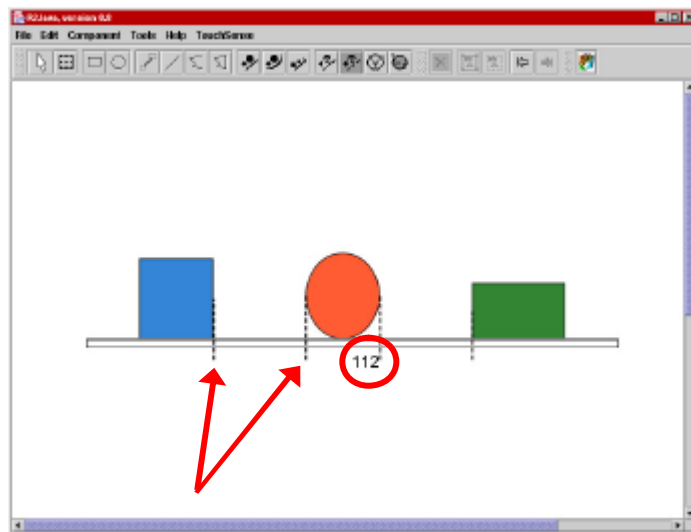
Kuva 5.2. Käsiniirretty kaavio (a), jota muokataan tasausellipsillä (b).

5.4.3. Tasavälipuikot

Olioiden tasaaminen vaaka- tai pystysuoraan tasoon on kaavioniirtosovelluksissa perusvaatimus, jonka jo kohdassa 4.5 tarkemmin esitelty R2-Draw-

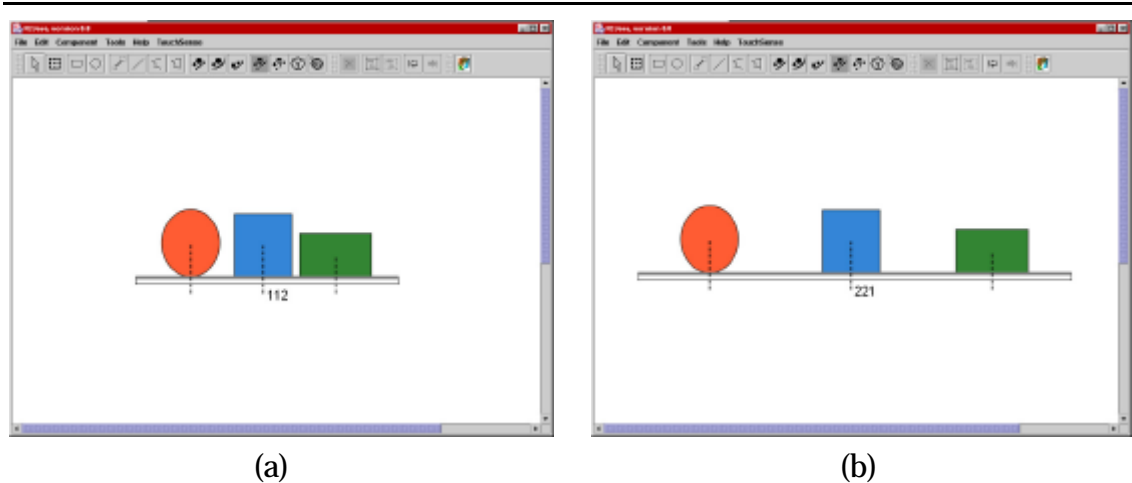
sovelluksen tasauspuikko täytti [Raisamo ja Rähä, 1996]. Tasavälipuikko onkin tasauspuikon suora jälkeläinen, jolla voi olioiden pelkän tasaamisen ja liikuttamisen lisäksi myös säätää tasattavien olioiden keskinäiset etäisyydet yhtäläisiksi.

Usein ei kuitenkaan ole lainkaan yksiselitteistä, pitäisikö olioiden etäisyydet tasata näiden keskipisteiden vai reunojen mukaan, joten tässä kohdassa on katsottu tarpeelliseksi kehittää kaksi eri versiota tasavälipuikosta, joiden toiminnallisuudessa ei ole välientasausmetodin lisäksi mitään eroa. Kuvassa 5.3 on olioiden reunojen etäisyydet tasaava tasavälityökalu. Kuvaan on ympyröity olioiden välin pikseleinä kertova lukema (kuvassa 112) sekä osoitettu nuolilla olioiden tasaussuunnassa uloimmat pisteet näyttävät apuviivat. Suurimpina eroina aiemmin kohdan 4.5 kuvassa 4.9 esitettyihin tasausluutatyökalun erilaisiin tasausvaihtoehtoihin tasavälipuikoissa on mahdollisuudet portaat-tomasti säätää olioiden välisiä etäisyyksiä ja liikuttaa valittuja olioita koko tasausprosessin ajan. Tämä tekee työkalun käytön selvästi joustavammaksi ja monipuolisemmaksi.



Kuva 5.3. Oliot reunojen mukaan tasaava tasavälipuikko.

Tasavälipuikon käyttäminen on helppoa: työkalulla valitaan halutut oliot koskettamalla niitä työkalukursorilla, aivan kuten tasauspuikossakin, jolloin työkalu automaattisesti tasaa valittujen olioiden väliset etäisyydet. Tämän jälkeen olioiden etäisyyksiä voi säätää säätämällä työkalukursorin pituutta. Kuvassa 5.4 on esimerkki oliojoukon tasaamisesta keskikohdat huomioivalla tasavälipuikolla ja olioiden välisten etäisyyksien säätämisestä käyttäjälle mieluisaksi. Kuvan 5.4a tilanteessa olioiden keskustojen väliset etäisyydet ovat 112 pikseliä, jonka jälkeen ne on kasvatettu 221 pikseliin kuvassa 5.4b.



Kuva 5.4. Oliojoukon tasaaminen ja olioiden keskinäisten välien säätäminen tasavälipuikon avulla.

5.5. Yhteenveto

Tässä luvussa esiteltyjen uusien kaavionkäsittelytyökalujen sekä aiemmin kohdassa 4.5 esiteltyjen tasauspuikon ja -kaaren avulla käyttäjän tulisi kyetä laatimaan haluamansa ulkoasu kaaviolle. Yhdessä nämä työkalut muodostavat suorakäyttöisen työkalujoukon, joka kattaa kaikki yleisimmät kaavionkäsittelyn tarpeet: olioiden tasaamisen ja niiden välien tasaamisen joko olioiden reunojen tai niiden keskikohtien mukaan, kaarevamuotoisten olioryhmien koostamisen sekä olioiden järjestämisen tasavälein keskuskappaleen ympärille. Näiden lisäksi erikseen määriteltävissä olevien tasauspisteiden avulla voidaan edelleen monipuolistaan yksittäisten olioiden ja olioryhmien järjestämistä.

Esimerkiksi ryhmän Ryall *et al.* [1997] GLIDE-sovelluksen vuorovaikutteisiin rajoitteisiin verrattuna suorakäyttöisellä työkaluvalikoimalla on mahdollista suoriutua yhtäläillä kaikista vastaavista muokkaustoimenpiteistä, joskin työkalujen perusluonteelle ominaisesti käyttäjän itsensä opastamana. Tasausellipsi sekä tasavälipuikot noudattavat automaattisesti symmetrisyyden, ryhmittelyn ja olioiden keskinäisten välien esteettisiä kriteerejä, minkä lisäksi näillä työkaluilla käyttäjä voi helposti noudattaa myös kaavion alan kriteeriä. Joidenkin esteettisten kriteerien, kuten olioiden etäisyyden, suhdeviivojen pituuden ja olioiden sekä suhdeviivojen päällekkäisyyden noudattaminen on jätetty käyttäjän itsensä vastuulle, koska niiden lisääminen täysin suorakäyttöiseen työkaluun ei ole ollut mahdollista.

Tässä luvussa esitellyt työkalut eivät ole vielä täysin valmiita, esimerkiksi olioiden välit tasaaviin työkaluihin pitäisi toteuttaa nykyistä onnistuneempi animointi, jotta käyttäjän olisi helpompi pysyä mukana työkalujen kaavioon

tekemissä muutoksissa. Eräs mahdollinen ja hyödyllinen lisäpiirre olisi esittää aina kulloinkin työkalun vaikutusalueessa oleva oliojoukko, tasauksen aikainen ryhmä, samanvärisenä ja siten helposti kaavion muista olioista erottuvana kokonaisuutena. Myös työkalukursorien kursoreissa on vielä kehittymahdollisuuksia, jotta ne esittäisivät nykyistä havainnollisemmin toiminta-ajatuksensa jo ennen työkalun kokeilemistä ja kuvaisivat paremmin esimerkiksi olioiden välisiä etäisyyksiä työkalun pinnalla.

Eräs näkökulma työkalujen käytön ja toiminnan parantamiseen on tuntopalautteen lisääminen sovellukseen. Tällä tavoin esimerkiksi työkalujen toimintaa voisi kuvata visuaalista palautetta tukevan tuntopalautteen avulla ja esteettisten kriteerien loukkaamisesta voitaisiin huomauttaa käyttäjää antamalla tästä tuntopalautetta, joka ei lainkaan rajoittaisi sovelluksen suorakäyttöisyyttä.

6. Tuntopalauteen lisääminen kaavionpiirto-sovellukseen

Kuten jo luvussa 2 esitettiin, tuntopalautea ei edelleenkään ole erilaisia pelien tehosteita lukuun ottamatta käytetty hyväksi normaaleille kotitietokoneille tarkoitetuissa sovelluksissa, vaikka tämän mahdollistavia edullisia tuntopalaute-laitteita on ollut markkinoilla jo vuosia. Tässä luvussa perehdytään tuntopalauteen lisäämiseen kaavionpiirto-ohjelman suorakäyttöisiin työkaluihin.

Todisteita edullisten voimapalaute-laitteiden käytön puolesta ovat esittäneet muun muassa Yu ja Brewster [2002], jotka osoittivat, että voimapalautehiirtä voidaan tehokkaasti käyttää hyväksi kaksiulotteisten graafien esittämisessä. Johansson ja Linde [1999] ovat puolestaan tutkineet voimapalauteohjaussauvan käyttöä visualisointivälineenä ja todenneet, että edullisia voimapalaute-laitteita voidaan onnistuneesti käyttää tiedon visualisointiin vaikka niillä onkin tiettyjä rajoitteita verrattuna kalliisiin tutkimuskäyttöön tarkoitettuihin tuntopalaute-laitteisiin.

Luvun ensimmäisessä kohdassa esitellään yksityiskohtaisesti tutkielmassa käytetty toteutusympäristö, minkä jälkeen tutustutaan tarkemmin käytetyn tuntopalaute-standardin ominaisuuksiin. Kolmannessa kohdassa selvitetään tuntopalaute-tehosteiden suunnitteluun liittyviä seikkoja sekä kerrotaan tuntopalauteen käytöstä kaavionpiirto-sovelluksen eri työkalutyypeissä. Tämän jälkeen luvun neljännessä kohdassa arvioidaan tuntopalauteen käytön mahdollisesti mukanaan tuomia etuja sekä seikkoja, joihin täytyy jatkossa kiinnittää enemmän huomiota. Lopuksi yhteenvedossa esitetään mahdollisia aiheita tuntopalauteen jatkokehitykselle.

6.1. Toteutusympäristö

Sovellusta on kehitetty 800 megahertsin Pentium III -pöytätietokoneella, jossa on Windows 2000 -käyttöjärjestelmä. Pieniä ongelmia voimapalaute-tehosteiden toteuttamiselle aiheutti se, että voimapalautehiiren ajuri ei aina toiminut, joten osa näiden tehosteiden testauksesta ja kehittämisestä on täytynyt tehdä Windows Me -ympäristössä. Sovelluksen toteutus-alustana oli Java 2, josta käytettiin Java Development Kitin versiota 1.4.

Tuntopalaute-tehosteiden lisäämiseen käytetty TouchSense-standardi mahdollistaa tehosteiden laatimisen ja testaamisen erityisellä Immersion Studio -ohjelmalla, jossa on graafinen käyttöliittymä tuntopalaute-tehosteiden muokkaamiseen ja äänitehosteiden liittämiseen osaksi näitä. Tässä tutkielmassa käytettiin kuitenkin tehosteiden liittämässä vaihtoehtoista toteutustapaa, jossa tuntopalaute-tehosteet luodaan yksitellen ohjelmallisesti. Tämän hieman

työlämmän tavan etuna on se, että tällöin käytettyjä tehosteita on mahdollista muokata dynaamisesti ohjelman suorittamisen aikana.

Tuntopalautelaitteina tässä tutkielmassa esitetyssä toteutuksessa käytettiin Logitechin iFeel Mouse -tärinäpalautehiirtä ja Wingman Force Feedback Mouse -voimapalautehiirtä, jotka molemmat tukevat Immersionin TouchSense-tuntopalautestandardia. Seuraavaksi tutustutaan hieman tarkemmin TouchSensestandardin sisältämiin tuntopalautehosteisiin. Käytetyt tuntopalautehiiret on esitelty tarkemmin kohdassa 2.5.

6.2. TouchSense-tehostetyypit






TouchSense-tuntopalautestandardi [Immersion, 2002] tukee yhteensä yhtätoista erityyppistä tuntopalautehostetta, jotka vaihtelevat tärinäpalautehiiren yksinkertaisesta tekstuurista ja ääri viivoista aina voimapalautehiiren painovoimaa tai joustaa jäljitteleviin tehosteisiin. Taulukossa 6.1 näkyy ensin, mitä tehostetyyppejä tämän tutkielman toteutuksessa käytetyt tuntopalautelaitteet tukevat. Tämän jälkeen erilaisten tehostetyyppien ominaisuudet esitellään tarkemmin myöhemmin tässä luvussa kuvattavien suunnitteluratkaisujen pohjaksi.

Tehostetyyppi	Tärinäpalautehiiri	Voimapalautehiiri
Jaksottainen	X	X
Tekstuuri	X	X
Kehys	X	X
Ellipsi	X	X
Jousi		X
Ruudukko	(X)	X
Vakio		X
Ramppi		X
Vaimennin		X
Kitka		X
Vastustus		X

Taulukko 6.1. Tuntopalautelaitteiden tukemat TouchSense-tehosteet.
[Immersion, 2002].

Jaksoittainen (periodic) tehoste on värähtelyä, jossa tietynlainen jakso toistuu niin kauan kuin tehostetta käytetään. Jaksottaisilla tehosteilla voidaan kuvata niin tuulettimen pyörimistä kuin parranajokoneen käyttämistä, mutta sitä voi käyttää myös yksittäisten tärähdysten tekemiseen. Tehosteen aaltomuoto voidaan valita viidestä eri vaihtoehdosta, neliöaallosta, siniaallosta, kolmioaallosta ja sahanteräaallosta ylös tai alas. Erilaiset aaltomuodot

esimerkkikuvineen on koottu taulukkoon 6.2. Tutkielmassa käytetyt tuntopalautehiiret eivät tue jaksottaisen tehosteen sahalaita- eivätkä kolmioaaltomuotoja, vaan ne tuntuvat näillä laitteilla yksinkertaisempaan neliöaaltomuotona.

Aaltomuoto	Esimerkkikuva
Neliö	
Sini	
Kolmio	
Sahalaita ylös	
Sahalaita alas	

Taulukko 6.2. Jaksottaisen tehosteen eri aaltomuodot. [Immersion, 2002].

Tekstuuritehosteen (texture) avulla voidaan kuvata erilaisia homogeenisiä pintarakenteita, kuten esimerkiksi hiekkapaperia tai tennismailan jänteitä. Tehosteen tuntee vain hiirtä liikuttaessa, ja se koostuu esteestä, jonka toistumisen välimatka, leveys, törmäyksen voimakkuus ja suunta vaaka- tai pysty akselille voidaan vapaasti määrittellä.

Kehyksellä (enclosure) voidaan määrittellä neliönmuotoiselle alueelle ulkoseinä, joka voi olla joko hidasteena liikkeelle tai toimia magneetin tavoin vetäen hiirtä sisäpuolelleen. Ulkoseinämän sisälle voidaan määrittellä muita tehosteita.

Ellipsitehoste (ellipse) on elliptistä muotoaan lukuun ottamatta täysin samanlainen kuin kehystehostekin. Kehys ja ellipsi toimivat kumpikin aidosti vain voimapalautehiirellä, mutta myös tärinäpalautehiiri tukee sitä ilmaisemalla kehukseen osumisen yksinkertaisella jaksottaistehosteella.

Jousitehosteella (spring) voidaan nimensä mukaisesti jäljitellä voimia, jotka syntyvät, kun jotakin takaisin lepotilaansa pyrkivää esinettä venyttää tai puristaa kokoon. Lepotila riippuu tehosteelle määritetystä keskipisteestä, joka määrää ohjaimen pyrkimisen joko keskipistettä kohti tai siitä poispäin. Jousta voidaan käyttää esimerkiksi ohjauksen apuvälineenä määrittelemällä puoleensa vetävä keskipiste ja muuttamalla tämän keskipisteen sijaintia paikasta toiseen, jolloin hiiri hakeutuu kohti haluttua kohdetta. Tärinäpalautehiiri ei tue jousitehostetta.

Ruudukkotehosteella (grid) voidaan luoda ruudukko, jonka viivoihin hiiri joko tarrautuu tai joita se hylkii. Ruudukon vastustus- tai vetovoima voidaan määrittellä erikseen vaaka- ja pystyakselille. Mikäli molemmat akselit vetävät hiirtä puoleensa, laite pyrkii ohjautumaan viivojen lähimpään risteyskohtaan. Tärinäpalautehiiri ei suoraan tue ruudukkotehostetta vaan korvaa sen tekstuurina.

Vakiotehoste (constant) kuvaa tietystä suunnasta tulevaa tasaista työntöä, joka työntää hiirtä vastakkaiseen suuntaan. Tehosteella voidaan jäljitellä esimerkiksi koillisesta puhaltavaa tuulta, joka työntää kohdetta lounaaseen päin. Tärinäpalautehiiri ei tue vakiotehostetta.

Ramppitehoste (ramp) muistuttaa tasaista voimaa sikäli, että myös tässä hiireen kohdistuu työntö tietystä suunnasta. Ramppitehosteessa työntävä voima kuitenkin joko kasvaa tai heikkenee lineaarisesti, aivan kuten painovoima tekee luonnossa mäkeä ylös- tai alaspäin kulkiessa. Tärinäpalautehiiri ei tue ramppitehostetta.

Vaimennintehosteella (damper) voi kuvata mitä tahansa liikettä vastustavaa ainetta, kuten vaikkapa vettä tai paksumpaa liejua. Vaimennin toimii kuten liike oikeassa nesteessä: mitä nopeammin tehoston vaikutuksessa yritetään liikkua, sitä suurempi tunnettu vastus on. Vaimentimen toiminnan voi muuttaa myös päinvastaiseksi, jolloin sillä pystyy voimistamaan liikkumista tiettyyn suuntaan siten, että hiiren liikuttaminen nopeasti vasemmalle aiheuttaakin liikkeen jatkumisen edelleen vasemmalle. Tällä tavoin voidaan siis kuvata myös liikkumista liukkaalla pinnalla. Tärinäpalautehiiri ei tue vaimennintehostetta.

Kitkatehosten (friction) avulla voi kuvata kahden toisiaan koskettavan pinnan liikkumista vastakkain. Tehosteella voi vaimennintehosten tavoin joko vastustaa käyttäjän liikkeitä tai myötäillä niitä jäljitellen liukasta pintaa. Tärinäpalautehiiri ei tue myöskään kitkatehostetta.

Vastustustehoste (inertia) vastustaa hiiren liikkeen muutoksia. Tehoste toimii dynaamisesti vastustaen sitä voimakkaammin mitä nopeammin hiirtä yritetään liikuttaa. Tärinäpalautehiiri ei tue vastustustehostetta.

6.3. Tuntopalautetehosteiden suunnittelu

Alettaessa suunnitella tuntopalautetehosteita on tarkkaan otettava huomioon käytettyjen tuntopalautelaitteiden ominaisuudet ja rajoitteet.

Tärinäpalautehiirien kohdalla on erityisesti huomioitava, että niiden yksinkertaiset ja matalataajuiset tärinä- ja värinätehosteet eivät kykene käytännössä muuhun kuin kertomaan käyttäjälle työkalujen eri tiloista ja sovelluksen toiminnassa tapahtuvista muutoksista. Esimerkiksi erilaisia pintamateriaaleja

tai olioiden muita fyysisiä ominaisuuksia niillä ei kovinkaan onnistuneesti pysty kuvaamaan.

Voimapalautehiiren värinätehosteet eivät ole näinkään onnistuneita, mutta voimapalauteominaisuuksiensa ansiosta hiirellä voi melko onnistuneesti ottaa käyttöön erilaisia todellisia tilanteita jäljitteleviä painovoima- ja liikkeen vastustustehosteita, joilla on hyvin suunniteltuina mahdollisuus parantaa merkittävästi koko sovelluksen todentuntuisuutta.

6.3.1. Taustaa

Erilaisia tuntopalautehosteita suunniteltaessa tulee käytetyn tuntopalautelaitteen asettamien rajojen lisäksi ottaa mahdollisuuksien mukaan huomioon myös todellisessa maailmassa vastaavissa tilanteissa saadut aistimukset. Kysymys on siis siitä, kuinka hyvin käytetyllä palautelaitteella voi jäljitellä todellisen maailman piirteitä.

Tässä tutkielmassa käytetyistä tuntopalautelaitteista voimapalautehiiri on selkeästi parempi, koska sen avulla voi paljon värinäpalautehiirtä paremmin jäljitellä kappaleiden fyysisiä ominaisuuksia, kuten kappaleen painoa, sen ääri rajoja ja liikkuttamisesta aiheutuvaa kitkaa, ja siten tehdä sovelluksesta intuitiivisemmän käyttäjä. Vaikka värinäpalautehiiren mahdollistamat tuntopalautehosteet rajoittuvatkin ainoastaan eri taajuuksiin ja eri aaltomuotoisiin värinöihin ja värinöihin, on senkin palautetta oikein käyttämällä mahdollisuus parantaa sovelluksen käytettävyyttä. Sillä ei kuitenkaan ole mahdollista toteuttaa voimapalautehiiren tasoista tuntomaailmaa.

Kohdassa 2.1 tuntoaistilla todettiin olevan melko huono ärsykemuisti, minkä vuoksi ihminen ei kykene helposti erottamaan eriaikaan saatuja tuntoaistimuksia toisistaan. Tuntokäyttöliittymiä suunniteltaessa tämä on huomionarvoinen seikka, sillä sekä käyttäjän suorituskyvyn että sovelluksen käyttömukavuuden kannalta olennaista on, että käytetyt palautetehosteet erottuvat toisistaan riittävästi ja ovat etenkin helposti erotettavissa toisistaan käyttäjän totuttua niihin. Tuntopalauteen erottamisen tukemiseksi käytettyjen tehosteiden täytyy tukea ihmisen tuntoaistin erityisominaisuuksia, mutta samalla myös pyrkiä mahdollisimman hyvin kuvaamaan työkalun toimintaa.

Tuntoaistin ominaisuuksia selvitettyä kohdassa 2.2 osoitettiin, että ihminen tunnistaa ja aistii oikeassa elämässä erityyppisiä ja eritaajuuksisia ärsykeitä eri aistinreseptorien välityksellä. Esimerkiksi tunnusteltavien kohteiden pintarakenteiden yksityiskohtia aistitaan pääasiassa matalataajuuksista informaatiota vastaanottavilla Merkelin kiekkoilla, kun Pacinianin keräset puolestaan välittävät tietoa erilaisten työkalujen käytöstä syntyvästä korkeataajuuksisesta värinästä.

Ongelmaksi tärinätehosteiden tarkemmassa suunnittelussa suorakäyttöisten työkalujen tueksi osoittautuivat hiirten tehosteita tuottavien moottorien taajuusalueet, joita Logitech ei tuotesalaisuuksiin vedoten suostunut luovuttamaan edes tutkimuskäyttöön. Tämän vuoksi on mahdotonta kohdistaa hiirten tehosteita juuri halutuille taajuusalueille ja siten myös aistinreseptoreille. Ongelma on sivuutettu käyttämällä eri tehosteiden suunnittelussa mahdollisimman erityyppisiä tehosteita. Lisäksi etenkin dynaamisten tehosteiden kohdalla on otettu huomioon myös kohdassa 2.2 mainittu tuntoaistin erottelukynnyks, jonka mukaan erottaakseen kaksi rinnakkaista ärsykettä toisistaan ihmiset tarvitsevat keskimäärin viiden prosentin eron näiden ominaisuuksien välillä.

6.3.2. Vaatimuksia

Tuntopalautteen käytölle on eri yhteyksissä laadittu monenlaisia vaatimuksia ja näitä tavoittelevia kriteeristöjä. Usein nämä on kuitenkin suunniteltu pääasiassa joko tuntopalautehiiriä monipuolisempaa haptista palautetta tuottaville laitteille tai täysin virtuaalisiin kolmiulotteisiin keinotodellisuuksiin, joten niitä ei voi suoraan soveltaa tässä tutkielmassa käsiteltyyn tapaukseen.

Esimerkiksi Massie ja Salisbury [1994] esittävät kolme välttämätöntä kriteeriä hyvälle voimapalautesovellukselle. Heidän mukaansa vapaan tilan täytyy tuntua vapaalta, kiinteiden virtuaalisten kohteiden täytyy tuntua kovilta ja virtuaalisia rajoitteita ei pidä voida kumota helposti. Nämä kriteerit pätevät hyvin etenkin PHANToMin kaltaisille korkeatasoisille tuntopalautelaitteille ja jotenkuten myös voimapalautehiirelle, mutta ei lainkaan tärinäpalautehiirelle. Lisäksi ensimmäistä lukuun ottamatta kriteerit eivät sovellu kovin hyvin esimerkkisovelluksen olosuhteisiin, jossa tuntopalautetta ei käytetä käyttäjän toiminnan rajoittamiseen.

Miller ja Zeleznik [1998] puolestaan ovat lisänneet voimapalautte-elementtejä X-Window-ympäristöön. Heidän tutkimuksessaan on eritelty kolme periaatetta tuntokäyttöliittymän suunnitteluun: virheiden vähentäminen käyttäjän liikettä ohjaavien voimien avulla, dynaamisen palautteen käyttäminen sekä käyttäjää ohjaavat rajoitteet, jotka on tarpeen tullen mahdollista kiertää.

Sjöström [1999] puolestaan sanoo, että tuntopalautteen käyttöpotentiaali riippuu siitä kuinka fysiikan lait on ohjelmoitu virtuaaliseen maailmaan. Sjöström on laatinut ominaisuuspareja, jotka on hyvä ottaa huomioon tuntopalautetta suunniteltaessa. Esitän tässä pareista neljä tähän tutkielmaan parhaiten sopivaa, jotka ovat:

- muuttumattomat oliot – dynaamiset oliot,
- konkreettiset esineet – abstraktit asiat,
- säädelyt liikkeet – vapaat liikkeet, ja

- hallinta – tunteminen ja koskettaminen.

Ensimmäinen pari on erityisen tärkeä pyrittäessä virtuaalimaailmassa jäljittelemään todellista maailmaa, suuri osa käyttäjän huomiosta kiinnittyy toimimiseen, erilaisten dion käsittelemiseen ja niiden piirteiden hahmottamiseen. Tällöin on olennaista kiinnittää suunnittelussa huomiota siihen, mitkä oliot ja niiden ominaisuudet tulee säilyttää muuttumattomina ja mitkä muuntuvat dynaamisesti tilanteesta riippuen. Tässä kohtaa omat ongelmansa suunnittelulle aiheuttaa myös toinen esitetty vertailupari: kuinka kuvata aidon ja havainnollisen tuntuisesti sekä olemassa olevia esineitä että ainoastaan virtuaalisessa maailmassa esiintyviä abstrakteja asioita.

Kaikenlaisessa käyttäjän toiminnan ja liikkeiden pakotetussa ohjaamisessa täytyy olla tarkkana, jottei se loukkaa käyttäjän vapautta. Ohjaaminen on hyvä asia niin kauan kun se auttaa käyttäjää toimimaan tarkemmin ja tehokkaammin virtuaalimaailmassa, mutta väärin käytettynä se voi helposti myös tuhota käyttäjän todellisuudentunteen. Lisäksi virtuaalimaailman hallinnan ja sen tuntemisen ja koskettamisen tulisi olla mahdollisimman todenmukaista aina, kun se on mahdollista – käyttäjän täytyy olla mahdollista vapaasti toimia virtuaalimaailmassa ja tuntea sen yksityiskohtia sekä pystyä hallitsemaan ja olemaan vuorovaikutuksessa ympäristön kanssa.

Oakley *et al.* [2001] tekevät hyvän huomion sanoessaan, että tuntopalaute-tehosteita suunnitellessa täytyy ennenkaikkea ottaa huomioon käytetyn tuntopalaute-laitteen ominaisuudet sekä mahdollisuudet – yritys ylittää käytetyn laitteen rajat johtaa lähes poikkeuksetta epäonnistumiseen. Tämän vuoksi käytetyt tuntopalaute-tehosteet täytyy suunnitella hyvin huolellisesti, jotta ne eivät ole pelkästään uusi jännittävä piirre sovelluksessa vaan oikeasti myös parantavat sen käytettävyyttä.

Kaksiulotteisen työpöytäympäristön sekä käytettyjen tuntopalaute-laitteiden erityisominaisuudet huomioon ottaen olen muodostanut joukon tuntopalauteen suunnittelun apuna käytettäviä periaatteita. Nämä periaatteet ovat nähdäkseni yleisesti sovellettavissa, kun halutaan suunnitella toimivia ja havainnollisia tuntopalautesovelluksia kaksiulotteiseen työpöytäympäristöön.

- Suunnittelijan täytyy olla selvillä käytetyn tuntopalaute-laitteen ominaisuuksista ja mahdollisuuksista. Esimerkiksi voimapalaute- ja värinäpalaute-laitteilla on molemmilla omat vahvuutensa ja heikkoutensa, jotka tulee ottaa tehosteiden suunnittelussa huomioon.
- Todellisen maailman ominaisuuksia ja lakeja tulisi noudattaa mahdollisuuksien mukaan myös virtuaalimaailmassa, jotta käyttäjä voi intuitiivisesti ymmärtää annetun palautteen ja siten todella myös

hyötyä siitä. Dynaamisen tuntopalautteen käyttäminen on eräs luonnollinen tapa kertoa käyttäjälle jonkin tietyn tehtävän etenemisestä.

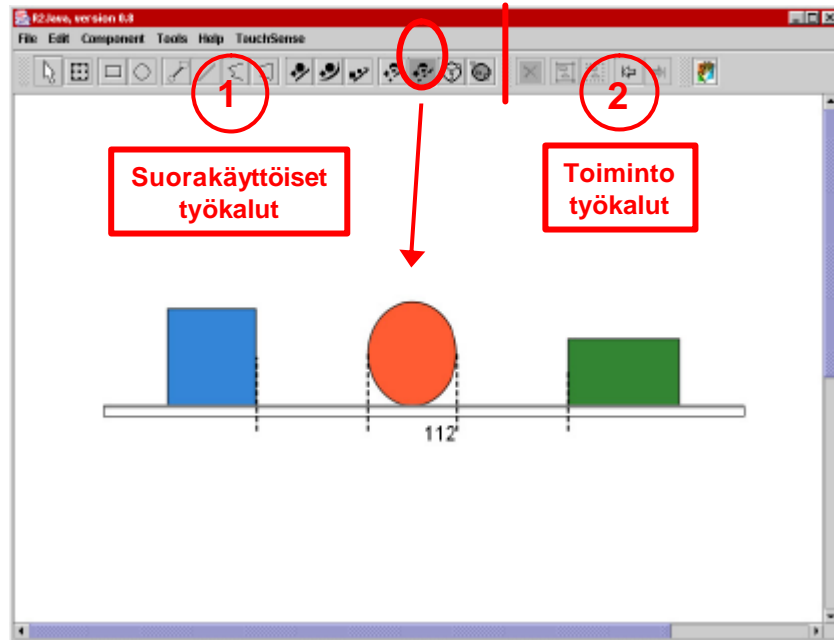
- Johdonmukaisuus on tärkeä osa tuntopalautetehosteiden suunnittelua. Tietystä tilanteesta tai toiminnosta saadun palautteen on tärkeää käyttäytyä joka kerralla samalla tavalla, jotta käyttäjä voi olla varman merkityksestä.
- Tuntoaistin ominaisuuksien tunteminen auttaa suunnittelemaan hyvin toisistaan erottuvia ja mahdollisimman havainnollisia tehosteita.
- Työpöytäympäristössä tuntopalautetta tulee aina käyttää järkevästi. Riittävän paljon, jotta käyttäjä todella hyötyy siitä, mutta ei kuitenkaan kokoajan, jottei käyttäjä ärsyynny.
- Tuntopalautteen tulee aina olla synkronoitu visuaaliseen palautteeseen, ellei se liity johonkin näkymättömään toimintoon tai ominaisuuteen.
- Koska jokainen käyttäjä on erilainen, täytyy myös tuntopalautteen ominaisuuksien ja voimakkuuden olla käyttäjän täydessä hallinnassa. Lisäksi käyttäjän toimintaa sovelluksen sisällä ei saa ilman erityistä syytä hallita tai rajoittaa.

Yllä olevat suunnitteluperiaatteet on sovellettu edullisille tuntopalaute-laitteille aiemmista kehittyneemmille tuntopalautelaitteille tehdyistä tutkimuksista. Voidaan kuitenkin olettaa, että esitettyjen periaatteiden avulla on mahdollista suunnitella ja kehittää sovelluksia, joissa tuntopalaute on luonnollinen sovelluksen käyttöä ja toimintaa parantava osatekijä. Kuten suunnitteluperiaatteiden määrästä ja luonteesta on hyvin nähtävissä, tuntopalautteen käyttäminen sovelluksessa ei suoraan johda parantuneeseen käytettävyyteen, vaan tuntoaistinmodaliteetin onnistunut hyödyntäminen vaatii osakseen perusteellista suunnittelua, onhan huolellisesti suunniteltu tuntopalaute omiaan lisäämään myös käyttäjän tyytyväisyyttä sovellusta kohtaan. Seuraavassa alakohdassa kerrotaan tarkemmin, miten tuntopalautetta on tässä tutkielmassa käytetty tukemaan kaavionkäsittelysovelluksen käyttöä.

6.3.3. Tuntopalaute työkalujen tukena

Käytetyssä kaavionkäsittelysovelluksessa on kahden tyyppisiä työkaluja, toimintotyökaluja ja suorakäyttöisiä työkaluja. Toiminto käynnistetään valitsemalla työkalu, jolloin se suorittaa toimintonsa ja jää odottamaan uutta käynnistystä. Suorakäyttöisellä työkalulla puolestaan tarkoitetaan työkalua, joka ensin valitaan ja jota sitten käytetään piirtoalueella. Kuvassa 6.1 on työkalujen

valintapalkista eroteltu toisistaan sovelluksen suoraikäyttöiset työkalut (1) sekä toimintotyökalut (2). Valintapalkissa tummempana näkyvä ja kuvaan korostetusti ympyröity työkalu on sovelluksessa käytössä oleva olioiden reunat tasaava tasavälipuikko.



Kuva 6.1. Työkalujen jaottelu suoraikäyttöisiin ja toimintotyökaluihin R2Javassa.

Tuntopalautteen tarkoitus on tehdä sovelluksen käytöstä intuitiivisempaa ja todenmukaisempaa välittämällä käyttäjälle informaatiota sovelluksen toiminnasta myös tuntoaistin kautta. Tähän pyrittäessä sovelluksen eri toiminto- ja työkalutyypeille on suunniteltu toisistaan erottuvat ja työkalujen käyttöä kuvaavat tehosteet, jotka ovat joko muuttumattomia ilmaisten jonkin yksinkertaisen asian tapahtumisen tai dynaamisia kuvaten työkalun käyttöä sen hetkisessä tilanteessa.

Toimintotyökaluissa käytetyt tehosteet ovat puhtaasti toiminnasta ja tilanmuutoksesta kertovia, esimerkkeinä voisi mainita vaikkapa työkalujen valinta- ja peruutustoiminnot. Näissä tuntopalautetehosteiden käyttö rajoittuu käytännössä pieneen hiirentärykseen tai -nytkähdykseen, joka tukee näytöllä tapahtuvaa visuaalista muutosta vahvistaen toiminnon suorittamisen.

Suoraikäyttöisissä työkaluissa, kuten esimerkiksi luvussa 6 esitellyissä tasaustyökaluissa, tuntopalautetta voi hyödyntää paljon monipuolisemmin. Näissä työkaluissa etenkin voimapalautehiiri antaa laajemmat mahdollisuudet kiinnittää huomiota sekä työkalujen että käsiteltävien olioiden fyysisiin ominaisuuksiin. Suoraikäyttöisten työkalujen ominaisuuksissa ja toiminnassa

voidaan täten dynaamisesti jäljitellä reaali maailman vastaavia tilanteita, joissa ihminen tuntee kädessään olevien työkalujen fyysiset muutokset tehokkaasti.

Taulukossa 6.3 on esitetty kaavionpiirto-sovelluksen tapahtumat, joissa tuntopalautetehosteita käytetään. Lisäksi taulukossa on kunkin tapahtuman kohdalla eritelty, millä tehosteilla tärinä- ja voimapalautelaitteet niitä kuvaavat sekä kerrottu tehosteen dynaamisuudesta.

Toiminto	Tärinäpalautte	Voimapalautte	Dynaamisuus
Työkalun valinta	Jaksottainen	Vakio	Ei
Työkalun aktivointi	Jaksottainen	Jaksottainen	Ei
Työkalun deaktivointi	Jaksottainen	Jaksottainen	Ei
Olioiden valinta	Jaksottainen	Vaimennin	Kyllä, olioiden tai olioryhmän koon mukaan
Työkalun raahaaminen	Tekstuuri	Kitka	Kyllä, olioiden määrän mukaan
Työkalun säätö	Tekstuuri	Jousi	Kyllä, työkalun koon (ja olioiden välien) mukaan
Varoitus	Tekstuuri	Vakio	Ei

Taulukko 6.3. Eri toimintoja kuvaavien tehosteiden tyypit ja dynaamisuus tärinä- ja voimapalauttehiirillä.

Sovelluksessa työkalun voi valita joko suoraan kuvassa 6.1 näkyvän sovellusikkunan yläreunan valintapainikkeista tai Tools-valikosta. Tärinäpalautelaitetta käytettäessä työkalun valinnasta kertoo käyttäjälle pieni täräys, joka on toteutettu jaksottaisen tehosteen avulla. Käytetty voimapalautetehoste on puolestaan vakiovoima, joka nykäisee käyttäjän kättä hieman alaspäin kuvaten, että jotakin on otettu käteen. Kaikille suorakäyttöisille työkaluille työkalun valinnasta kertova tehoste on aina sama, mutta toimintotyökaluilla, esimerkiksi poista-, ryhmitä- ja kumoa-työkaluilla, on kaikilla omat kyseisen työkalun toimintaa kuvaamaan pyrkivät tehosteensa.

Monet suorakäyttöisistä työkaluista voidaan erikseen aktivoida tai deaktivoida. Kumpaakin toimintoa kuvaa sekä tärinä- että voimapalautelaitteilla sama muuttumaton jaksottainen tehoste. Aktivointitehoste on lyhyt tärähdys, joka pyrkii kuvaamaan toiminnan alkamista ja deaktivointitehoste puolestaan on ominaisuuksiltaan aktivointitehosteen vastakohta antaen tällä tavoin ymmärtää, että suoritettu toiminta loppuun.

Olioiden valinnasta suorakäyttöisillä työkaluilla kertoo tärinäpalautelaitteilla tärähtävä jaksottainen tehoste ja voimapalautelaitteilla puolestaan vaimennintehoste, joka tuntuu käyttäjälle kevyenä töytäisynä. Kumpikin toiminto kuvaava tehoste on dynaaminen, mikä tässä tapauksessa tarkoittaa, että tehosteen voimakkuus on sitä suurempi mitä suurempi olion pinta-ala on. Tehosteen dynaamisuudella on siis pyritty kuvaamaan erikokoisten olioiden fyysisiä kokoeroja.

Työkalun raahaaminen piirtoalueella tuntuu tärinäpalautelaitteella tekstuurina ja voimapalautelaitteella kitkatehosteena kummankin pyrkien kuvaamaan työkalun käytön vaatimaa voimaa. Toiminnon tehosteet ovat dynaamisia ja niiden voimakkuudet riippuvat siitä, kuinka montaa oliota työkalulla käsitellään.

Joidenkin suorakäyttöisten työkalujen, kuten tasauspuiikon sekä tässä tutkielmassa esiteltyjen uusien suorakäyttöisten kaavionkäsittelytyökalujen kokoa voi säätää. Säättötoimintoa kuvaa tärinäpalautelaitteilla tekstuuri-tehoste ja voimapalautelaitteilla jousitehoste, joka joko auttaa käyttäjää pidentämään tai lyhentämään työkalua tai sitten vastustaa käyttäjän liikettä, jos esimerkiksi tasavälipuikolla tasattavat oliot menevät päällekkäin. Työkalun säätöä kuvaavat tehosteet ovat dynaamisia ja riippuvat työkalun koosta sekä joissakin tapauksissa edellä mainitulla tavalla myös olioiden keskinäisistä etäisyyksistä.

Varoitustoiminto on tarkoitettu käytettäväksi lähinnä tärinäpalautelaitteilla tilanteissa, joissa esimerkiksi liian vakavasti rikotaan jotakin luettavuuden kriteeriä. Eräs tilanne, jossa varoitustoimintoa käytetään, on kertoa käyttäjälle tasavälipuikkoa käytettäessä olioiden päällekkäisyydestä. Käyttäjän varoittamiseen käytetty tehostetyyppi on tärinäpalautelaitteilla voimakas negatiivinen tekstuuri, joka tuntuu käyttäjälle terävinä tärähdyksinä säätöliikkeen varoitettavaan suuntaan. Voimapalautelaitteilla tätä toimintoa ei ainakaan vielä tällä hetkellä ole käytössä, mutta tarvittaessa sitä kuvataan vakiotehosteella, joka työntää käyttäjää pois varoitettavasta suunnasta. Voimapalautteen avulla olisi täten mahdollista auttaa käyttäjää noudattamaan esteettisiä kriteerejä ohjaamalla käyttäjän liikkeitä.

6.4. Tuntopalautteen arviointi

Ilman testejä oikeissa käyttötilanteissa on mahdotonta pitävästi arvioida sovelluksen tuntopalautetehosteiden suunnittelun onnistumista sekä etenkin niiden toimintaa käytännössä. Tämän vuoksi tässä kohdassa pitäydytäänkin pääasiassa vertaamaan sovelluksen tuntopalauteominaisuuksia alakohdassa 6.3.2 esitettyihin vaatimuksiin ja suunnitteluperiaatteisiin. On kuitenkin selvää, että tuntopalaute tuo uusia mahdollisuuksia ja piirteitä sovelluksen käyttämiseen, joita ei pelkän visuaalisen palautteen avulla ole mahdollista

ilmaista käyttäjälle; esimerkkinä tästä erilaisten esteettisten kriteerien noudattaminen, mitä voi etenkin voimapalautelaitteiden avulla helpottaa.

Käytettyjen tuntopalautelaitteiden erityisominaisuudet on pyritty ottamaan mahdollisimman hyvin huomioon. Esimerkiksi kuhunkin toimintoon käytetyt palautetyypit riippuvat käytössä olevasta tuntopalautelaitteesta, jonka sovellus tunnistaa käynnistyessään. Lisäksi dynaamisilla ominaisuuksilla on pyritty tuomaan lisää syvyyttä käytettyihin tehosteisiin tilanteissa, joissa se on nähty tarpeelliseksi. Näin tuntopalautteen käytöllä on pyritty tekemään sovelluksen käytöstä todenmukaisempaa ja samalla myös miellyttävämpää.

Tuntoaistin ominaisuuksia on sovelluksessa yritetty huomioida muuttumattomien tehosteiden kohdalla käytettyjen tuntopalautetehosteiden erilaisuutena ja pyrkimyksenä kuvata niiden avulla mahdollisimman hyvin kutakin tapahtumaa. Dynaamisten tehosteiden kohdalla käytettyjen tehosteiden voimakkuuden muutoksissa on lisäksi pyritty mahdollisimman hyvin ottamaan huomioon tuntoaistin erottelukynnys, jotta käytettyjen tehosteiden muutokset ovat käyttäjän havaittavissa.

Ylipäättään tuntopalautetta on sovelluksessa pyritty käyttämään järkevästi, esimerkiksi pelkästä työkalukursorin liikuttamisesta piirtoalustalla ei anneta lainkaan palautetta käyttäjälle. Lisäksi käytetyt palautetehosteet ovat aina samoissa toiminnoissa ja tilanteissa samanlaisia, ainoastaan dynaamisten tehosteiden kohdalla niiden ominaisuudet riippuvat työkalun käytön kuvaamiselle olennaisista seikoista.

Sovelluksen tuntopalaute on pyritty synkronoimaan mahdollisimman hyvin visuaaliseen palautteeseen, eikä ainakaan alustavissa käyttäjätesteissä tämän suhteen ole ilmennyt ongelmia. Toinen asia kuitenkin on, kuinka hyvin synkronointi toimii, kun käsitellään useita kymmeniä olioita yhtäaikaisesti. Tämä siksi, että etenkin dynaamiset tehosteet vaativat säännöllistä päivittämistä, jolloin mahdolliset viiveet alkavat helposti kertautua ja saattavat aiheuttaa käyttäjälle epävarmuutta palautteen merkityksestä.

Sovelluksessa ei paria poikkeusta lukuun ottamatta yritetä ohjata käyttäjän toimintaa. Luettavuuden kriteerien parempi tukeminen vaatisi sovellukselle lisää valtaa vaikuttaa käyttäjän toimintaan, mutta vaarana kaikessa tämän kaltaisessa ohjailussa on aina ristiriita käyttäjän pyrkimysten ja sovelluksen toiminnan välillä. Tämän vuoksi on ehkä parempi, että käyttäjä saa tehdä mitä haluaa ja sovellus enintään varoittaa, jos jotakin asiaa ei ole huomioitu tarpeeksi.

Todellisia pintapuolisia epäkohtia liittyen sovelluksen tuntopalautteeseen on havaittavissa vain muutamia. Ensinnäkin käyttäjällä ei ole mahdollisuutta vaikuttaa sovelluksessa käytettävien tehosteiden ominaisuuksiin, ei edes

tehosteiden yleiseen voimakkuuteen. Käyttäjä voi ainoastaan päättää, onko tuntopalauteominaisuus käytössä vai ei.

Toinen epäkohta on, että käytettyä tuntopalaute laitetta ei ole mahdollista vaihtaa kesken sovelluksen toiminnan vaan se on aina käynnistettävä uudelleen, jotta uusi laite voidaan tunnistaa. Lisäksi käytetyn ohjelmointirajapinnan ominaisuuksien vuoksi sovellus tunnistaa käytettäväksi aina ensimmäisen havaitsemansa tuntopalaute laitteen. Nämä eivät kuitenkaan ole merkittäviä ongelmia tuntopalauteen hyödyntämisen kannalta.

Lisäksi huomionarvoinen seikka sovelluksen tuntopalauteessa on, että mitkään käytetyistä tuntopalauteominaisuuksista eivät ole lähtöisin käsiteltävistä olioista vaan ne kaikki on toteutettu kulloinkin käytössä olevan työkalun kautta. Tämä johtaa siihen, että oliot eivät itsessään tunnu miltään, vaan ne ovat käyttäjälle fyysisesti olemassa ainoastaan, kun niitä käsitellään jollakin sovelluksen työkaluista. Saattaisi olla hyödyllistä lisätä myös olioihin tuntoominaisuuksia, jotka olisivat koko ajan voimassa ja tunnusteltavissa riippumatta siitä, onko käytössä oleva työkalu aktivoitu vai ei. Käytetyn ohjelmointirajapinnan avulla olisikin helppo korostaa olioiden fyysisiä ominaisuuksia liittämällä niihin esimerkiksi niiden muodosta tai väristä kertovia tehosteita, jolloin olioiden olemassaolo tulisi todellisemmaksi.

6.5. Yhteenveto

Tässä luvussa pyrittiin kertomaan tuntopalauteen lisäämiseen liittyvistä seikoista. Sovelluksessa käytetyn tuntopalauteen on tarkoitus parantaa kaavionpiirtotyökalujen käytön luonnollisuutta kaventamalla Buxtonin [2001] mainitsemaa kuilua fyysisen ja virtuaalisen maailman välillä. Käytännössä tuntopalauteen avulla on mahdollista elävöittää sovellusta ja työkalujen metaforia tehden näistä tällä tavoin entistäkin suorakäyttöisempiä ja lyhentää Hutchins *et al.* [1986] mainitsemia etäisyyksiä päämääristä järjestelmään ja järjestelmästä päämääriin. Luvussa esitetyn kaavionpiirtosovelluksen tapauksessa tämä tarkoittaa työkalujen toiminnan ja käsiteltävien olioiden fyysisten ominaisuuksien kuvaamista tärinä- tai voimapalaute tehosteilla.

Eräs ongelma tuntopalauteen lisäämisessä sovellukseen on käytettyjen tuntopalaute laitteiden erilaiset ominaisuudet: tärinäpalautehiirellä ei voida lainkaan tuottaa voimapalaute tehosteita, kun voimapalautehiiren tärinätehosteet puolestaan ovat heikkotasoisempia kuin tärinäpalautehiiren vastaavat. Tällä hetkellä sovellus tunnistaa käynnistettäessä käytössä olevan tuntopalaute laitteen tyyppin ja luo käytettävät tehosteet sen mukaan. Yksi ratkaisu erityyppisten tehosteiden yhdistämiselle olisi kummankin tuntopalaute laitteen käyttäminen rinnakkain, jolloin niiden kummankin parhaat ominaisuudet saataisiin käyttöön. Erityisen mielenkiintoinen ajatus

olisikin yhdistää tärinäpalauteohjauspallo heikomman käden syöttölaitteeksi voimapalautehiiren rinnalle, mutta ainakaan tutkielmassa käytetty versio Immersionin TouchSense API:sta ei tue kahden tuntopalaute laitteen yhtäaikaista käyttöä.

On todennäköistä, että tuntopalautteen tehokas hyväksikäyttö vaatii käyttäjältä aikaa tottua uudenlaiseen palautemuotoon ja herkistyä sen havaitsemiseen. Tämä seuraa jo siitä, etteivät läheskään kaikki työpöytäympäristöön sisältyvät kohteet, työkalut ja tehtävät ole käyttäjille todellisuudesta ennestään tuttuja. Niitä kuvaavasta palautteesta on siten käytännössä mahdotonta tehdä niin intuitiivista, että kaikki käyttäjät ymmärtävät sen ilman opettelua.

Tulevissa käyttäjätesteissä tuleekin keskittyä tutkimaan, millaisia etuja tärinä- ja voimapalaute tuovat sovelluksen käyttöön verrattuna pelkän visuaalisen palautteen käyttämiseen. Lisäksi lienee tarpeellista suorittaa myös pitkäaikaistutkimus sovelluksen käytöstä, jotta tuntopalautteen oppimisvaikutusta voidaan selvittää.

7. Lopuksi

Tässä tutkielmassa esittelin joukon toteuttamiani suorakäyttöisiä kaavionkäsittelytyökaluja, joita käyttämällä kaavioiden muokkaamisesta on mahdollista tehdä täysin vuorovaikutteinen ja käyttäjän hallittavissa oleva prosessi. Lisäksi esiteltyjen työkalujen ja koko kehyksenä olleen kaavionpiirto-sovelluksen käyttöä tukemaan olen suunnitellut ja toteuttanut tuntopalautteen, jonka avulla käyttäjän on mahdollista saada monipuolisempaa palautetta sovelluksen toiminnasta kuin pelkkää visuaalista palautetta käytettäessä.

Työkalujen toimintaa käytännössä ja tuntopalautteen todellisia hyötyjä ei kuitenkaan voida todistaa ennen niiden testaamista. Aiempien tutkimusten ja muun kirjallisuuden pohjalta voi kuitenkin perustellusti päätellä, että tässä tutkielmassa esitetyt kaavionkäsittelytyökalut kaventavat kuilua, joka automaattisten ja manuaalisten kaavionmuokkausmenetelmien väliin on jäänyt.

Ensimmäiseksi suunnitelmissa on testata ja vertailla tutkielmassa esiteltyjä vuorovaikutteisia kaavionmuokkaustyökalun etuja perinteisiin käyttäjän ohjaamiin komentopohjaisiin kaavionkäsittelytyökaluihin nähden. Koehenkilöinä testeissä on suunniteltu käytettävien kokeneita tietokoneen käyttäjiä, jotka käyttävät töissään kaavionpiirto-sovelluksia. Testin käyttötilanteiden tulisi tulosten luotettavuuden vuoksi olla mahdollisimman luonnollisia, joten muokattavien kaavioiden on ajateltu noudattavan *UML-standardia* (Unified Modeling Language). Tuntopalautteen toimintaa testattaessa pyritään puolestaan selvittämään, millaisia hyötyjä edullisten tuntopalautelaitteiden avulla on mahdollista saavuttaa ilman tuntopalautetta käytettyihin työkaluihin verrattuna sekä mihin tarkoituksiin eri tuntopalautetyypit soveltuvat käytettäväksi. Tämän lisäksi testeillä pyritään kokeilemaan tässä tutkielmassa esitettyjen suunnitteluperiaatteiden toimivuutta käytännössä.

Jatkossa päämääräksi tulisi ensin ottaa työkalujen toiminnan ja visualisoinnin kehittäminen edelleen. Näiden lisäksi myös induktiivisten ryhmien toteutuksessa riittää pohtimista. Esimerkiksi muokkausoperaatioiden ajaksi työkalujen vaikutuspiirissä olevia olioita voisi korostaa värikoodauksella, jolloin ne hahmottuisivat yhtenä kokonaisuutena ja siten erottautuisivat paremmin kaavion muista olioista. Myös työkaluilla muokattujen olioiden ryhmittämistä pysyvämmiin kannattaa harkita, jotta kertaalleen tasattuja olioita voisi käsitellä nykyistä helpommin kokonaisuutena.

Esitellyssä kaavionpiirto-sovelluksessa tuntopalautteen on tarkoitus tukea visuaalista palautetta ja siten parantaa sovelluksen käytön tasoa. On kuitenkin eri asia aistia enemmän kuin pystyä suoriutumaan tehtävistä tehokkaammin –

tämän vuoksi sovelluksen kaksikäätistä käyttöä tulisi pyrkiä parantamaan, jotta myös tekemisen taso nousisi entisestään. Erityisen mielenkiintoista olisi kokeilla kaksikäätistä tuntopalautetta valitsemalla dominoivalle kädelle voimapalautehiiri ja ei-dominoivalle kädelle tärinäpalauteohjauspallo, jolloin molempien laitetyyppien vahvuudet olisivat hyödynnettävissä. Lisäksi käyttäjän kannalta myös mahdollisuus itse hallita sovelluksen tuntopalautteen voimakkuutta ja ehkä myös käytettyjen tehosteiden tarkempia ominaisuuksia olisi tervetullut parannus tulevaisuudessa.

Tutkielmassa esitetty tapa muokata kaavioita tuo aiempia menetelmiä enemmän vuorovaikutteisuutta kaavionkäsittelyyn auttaen siten käyttäjää suoriutumaan paremmin tehtävästään. Tämän ansiosta käyttäjä voi itse laatia mieleisensä ulkoasun muokkaamalleen kaaviolle tehokkaasti ja tarkasti. Lisäksi visuaalisen palautteen tukena käytetty tuntopalaute tekee sovelluksesta entistäkin suorakäyttöisemmän, mikä omalta osaltaan lisää käyttäjän tyytyväisyyttä sovellusta kohtaan sekä tekee siitä todentuntuisemman käyttöä. Uskon, että edellä mainitut ominaisuudet tekevät tutkielmassa esitetyistä työkaluista aiempia kaavionkäsittelymenetelmiä monipuolisemman ja käyttäjäystävällisemmän tavan muokata kaavioita.

Viiteluettelo

- [Akamatsu ja MacKenzie, 1996] Motoyuki Akamatsu ja Scott MacKenzie, Movement characteristics using a mouse with tactile and force feedback. *International Journal of Human Computer Studies* **45** (1996), 483-493.
- [Apple, 1984] Apple Computer Inc., *MacDraw Manual*, 1984.
- [ArgoUML, 2001] ArgoUML version 0.9.5, <http://argouml.tigris.org/>. (Tarkastettu 27.5.2002)
- [Beaudouin-Lafon, 2000] Michel Beaudouin-Lafon, Instrumental interaction: An interaction model for designing post-WIMP interfaces. In: *Proceedings of ACM CHI 2000 Human Factors in Computing Systems*, ACM Press, 2000, 446-453.
- [Beaudouin-Lafon et al., 2000] Michel Beaudouin-Lafon, Wendy E. Mackay, Peter Andersen, Paul Janecek, Mads Jensen, Michael Lassen, Kasper Lund, Kjeld Mortensen, Stephanie Munck, Anne Ratzner, Katrine Ravn, Søren Christensen ja Kurt Jensen, CPN/Tools: A post-WIMP interface for editing and simulating coloured Petri Nets. In: K.H. Mortensen (ed.), *Tool Demonstrations Collection, 21st International Conference on Application and Theory of Petri Nets*, Aarhus, Denmark, 2000, 19-28. Available also as <http://www.daimi.au.dk/CPnets/CPN2000/download/CPNTools.pdf>. (Tarkastettu 27.5.2002)
- [Bewley et al., 1983] William Bewley, Teresa Roberts, David Schroit ja William L. Verplank, Human factors testing in the design of Xerox's 8010 "Star" office workstation. In: *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, ACM Press, 1983, 72-77.
- [Bier, 1989] Eric A. Bier, *Snap-Dragging: Interactive Geometric Design in Two and Three Dimensions*. Xerox Palo Alto Research Center, *Technical Report EDL-89-2-09*, 1989.
- [Bier ja Stone, 1986] Eric A. Bier ja Maureen C. Stone, Snap-Dragging. *ACM Computer Graphics* **20**, 4 (1986), *proceedings of SIGGRAPH'86*, 233-240.
- [Burdea, 1996] Grigore C. Burdea, *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, 1996, ISBN 0-471-02141-5.
- [Buxton, 2001] William Buxton, Less is More (More or Less): Speculations about the design of computers. In: P. Denning (ed.), *The Invisible Future: The seamless integration of technology in everyday life*, McGraw Hill, New York, 2001, 145-179.
- [Böhringer ja Newbery Paulisch, 1990] Karl-Friedrich Böhringer ja Frances Newbery Paulisch, Using constraints to achieve stability in automatic graph layout algorithms. In: *Proceedings of the ACM CHI'90, Conference*

proceedings on Empowering people: Human factors in computing system: special issue of the SIGCHI Bulletin, ACM Press, 1990, 43-51.

- [Christian, 2000] Kevin Christian, Design of haptic and tactile interfaces for blind users. In: *UUGuide: Practical Design Guidelines for Universal Usability*, University of Maryland, College Park, USA, 2000. Available as <http://www.otal.umd.edu/UUGuide/>. (Tarkastettu 27.5.2002)
- [Cohen *et al.*, 1992] Robert F. Cohen, Giuseppe Di Battista, Roberto Tamassia, Ioannis G. Tollis ja Paola Bertolazzi, A framework for dynamic graph drawing. In: *Proceedings of 8th Symp. Computational Geometry*, ACM Press, 1992, 261-270.
- [Coleman ja Parker, 1996] Michael K. Coleman ja D. Stott Parker, Aesthetic-based graph layout for human consumption. *Software - Practice and Experience* **26**, 12 (Dec. 1996), 1415-1438.
- [Dengler ja Cowan, 1998] Edmund Dengler ja William Cowan, Human perception of laid-out graphs. In: S.H. Whitesides (ed.), *Lecture Notes in Computer Science 1547, proceedings of Graph Drawing '98*. Springer-Verlag, 1998, 441-443.
- [Dennerlein *et al.*, 2000] Jack T. Dennerlein, David B. Martin ja Christopher Hasser, Force-feedback improves performance for steering and combined steering-targeting tasks. In: *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, ACM Press, 2000, 423-429.
- [Di Battista *et al.*, 1994] Giuseppe Di Battista, Peter Eades, Roberto Tamassia ja Ioannis G. Tollis, Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry Theory and Applications* **4** (1994) 235-282. Available as <ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.gz>. (Tarkastettu 27.5.2002)
- [Di Battista *et al.*, 1999] Giuseppe Di Battista, Peter Eades, Roberto Tamassia ja Ioannis G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 1999, ISBN: 0-13-301615-3.
- [Ding ja Mateti, 1990] Chen Ding ja Prabhaker Mateti, A framework for the automated drawing of data structure diagrams. *IEEE Transactions on Software Engineering* **16**, 5 (1990), 543-557.
- [Eades *et al.*, 1991] Peter Eades, Wei Lai, Kazuo Misue ja Kozo Sugiyama, Preserving the mental map of a diagram. In: *Proceedings of the International Conference on Computational Graphics and Visualization Techniques, COMPUGRAPHICS'91*, Portugal, September 1991, 34-43.
- [Gleicher, 1992] Michael Gleicher, Briar: A constraint-based drawing program. In: *Proceedings of the ACM CHI'92*, ACM Press, 1990, 661-662.

- [Gleicher, 1994] Michael Gleicher, Practical issues in graphical constraints. In: V. Saraswat and P. Van Hentenryck (eds.), *Principles and Practice of Constraint Programming*. MIT Press, 1994.
- [Goldstein, 1999] E. Bruce Goldstein, *Sensation & Perception, Fifth Edition*. Brooks/Cole Publishing Company, 1999, ISBN: 0-534-34680-4.
- [Halskov Madsen, 1994] Kim Halskov Madsen, A guide to metaphorical design. *Communications of the ACM* **37**, 12 (1994), 57-62.
- [Halskov Madsen, 2000] Kim Halskov Madsen, Magic by metaphors. In: *Proceedings of DARE 2000 conference on Designing augmented reality environments*, Elsinore, Denmark, ACM Press, 2000, 167-169.
- [Herman *et al.*, 2000] Ivan Herman, Guy Melan?on ja M. Scott Marshall, Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics* **6**, 1 (Jan.-Mar. 2000), 24-43.
- [Hudson, 1990] Scott Hudson, Semantic snapping: A technique for semantic feedback at the lexical level. In: *Proceedings of the 1990 SIGCHI Conference*, ACM Press, April 1990, 65-70.
- [Hutchins *et al.*, 1986] Edwin L. Hutchins, James D. Hollan, Donald A. Norman, Direct manipulation interfaces. In: D.A. Norman and S.W. Draper (eds.), *User Centered System Design*. Lawrence Erlbaum, 1986, 87-124.
- [Immersion, 2002] Immersion Corporation, <http://www.immersion.com/>. (Tarkastettu 27.5.2002)
- [Jansson ja Billberger, 1999] Gunnar Jansson ja Katarina Billberger, The PHANToM used without visual guidance. In: *Proceedings of the First PHANToM Users Research Symposium (PURS'99)*, 1999, Deutsches Krebsforschungszentrum, Heidelberg, Germany.
- [Johansson ja Linde, 1999] Anders J. Johansson ja Joakim Linde, Using simple force feedback mechanisms as haptic visualization tools. In: *The 16th IEEE Instrumentation and Measurement Technology Conference (IMTC'99)*, 1999.
- [Johnson, 2001] Kenneth O. Johnson, The roles and functions of cutaneous mechanoreceptors. *Current Opinion in Neurobiology* **11** (2001), 455-461.
- [Katz, 1948] David Katz, *Hahmopsykologia*. Suom. Martti Takala. Otava, Helsinki, Kauppalehti Oy:n Kirjapaino, 1948.
- [Knuth, 1963] Donald E. Knuth, Computer-drawn flowcharts. *Communications of the ACM* **6**, 9 (1963), 555-563.
- [Lederman ja Klatzky, 1993] Susan J. Lederman ja Roberta L. Klatzky, Extracting object properties through haptic exploration. *Acta Psychologica* **84** (1993), 29-40.

- [Logitech, 2002] Logitech Inc., <http://www.logitech.com/>. (Tarkastettu 27.5.2002)
- [MacLean, 2000] Karon E. MacLean, Designing with haptic feedback. In: *Proceedings of IEEE Robotics and Automation, ICRA'2000*, San Francisco, USA, 2000.
- [Massie, 1996] Thomas H. Massie, *Initial haptic explorations with the Phantom: Virtual touch through point interaction*. Massachusetts Institute of Technology, MS Thesis, January 1996.
- [Massie ja Salisbury, 1994] Thomas H. Massie ja J. Kenneth Salisbury, The PHANTOM haptic interface: A device for probing virtual objects. In: *Proceedings of ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, USA, 1994.
- [Masui, 1994] Toshiyuki Masui, Evolutionary learning of graph layout constraints from examples. In: *Proceedings of the ACM UIST '94 Symposium on User Interface Software and Technology*, ACM Press, 1994, 103-108.
- [McGee *et al.*, 2000] Marilyn Rose McGee, Phil Gray ja Stephen Brewster, The effective combination of haptic and auditory textural information. In: *Proceedings of the First Workshop on Haptic Human-Computer Interaction*, University of Glasgow, 2000, 33-38.
- [McGee *et al.*, 2001] Marilyn Rose McGee, Phil Gray ja Stephen Brewster, Haptic perception of virtual roughness. In: *Extended Abstracts of ACM CHI 2001*, ACM Press Addison-Wesley, 2001, 155-156.
- [Miller ja Zeleznik, 1998] Timothy Miller ja Robert Zeleznik, An insidious haptic invasion: Adding force feedback to the X desktop. In: *Proceedings of ACM UIST'98*, ACM Press, 1998, 59-64.
- [Minsky *et al.*, 1990] Margaret Minsky, Ming Ouh-young, Oliver Steele, Frederick P. Brooks Jr. ja Max Behensky, Feeling and seeing: Issues in force display. In: *Computer Graphics* **24**, 2 (1990), *Proceedings of SIGGRAPH Symposium on 3D RealTime Interactive Environments*, 235-243.
- [Misue *et al.*, 1995] Kazuo Misue, Peter Eades, Wei Lai ja Kozo Sugiyama, Layout adjustment and the mental map. *Journal of Visual Languages and Computing* **6**, 2 (1995), 195-209.
- [MIT, 2002] MIT Touch Lab, <http://touchlab.mit.edu/>. (Tarkastettu 27.5.2002)
- [MS PowerPoint, 1999] *Microsoft® PowerPoint® 2000 SR-1 (9.0.4527)*, Microsoft Corporation (1987-1999).
- [MS Visio, 2001] *Microsoft® Visio® Professional 2002 (10.0.525)*, Microsoft Corporation (1991-2001).

- [Münch ja Dillman, 1997] Stefan Münch ja Rüdiger Dillmann, Haptic output in multimodal user interfaces. In: *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, 1997, 105-112.
- [Newbery Paulisch, 1993] Frances Newbery Paulisch, The design of an extensible graph editor. In: G. Goos and J. Hartmanis (eds.), *Volume 704 of Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, 1993.
- [Oakley et al., 2000] Ian Oakley, Marilyn Rose McGee, Stephen Brewster ja Philip Gray, Putting the feel in "look and feel". In: *Proceedings of ACM CHI 2000*, ACM Press, Addison-Wesley, 2000, 415-422.
- [Oakley et al., 2001] Ian Oakley, Stephen Brewster ja Philip Gray, Solving multi-target haptic problems in menu interaction, In: *Proceedings of the ACM CHI'2001*, ACM Press, 2001, 357-358.
- [Olsen ja Deng, 1996] Dan R. Olsen Jr. ja Xinyu Deng, Inductive groups. In: *ACM UIST'96 Symposium on User Interface Software and Technology*, ACM Press, 1996, 193-199.
- [Purchase, 1997] Helen C. Purchase, Which aesthetic has the greatest effect on human understanding? In: Giuseppe Di Battista (ed.), *Proceedings of the Graph Drawing '97, volume 1353 of Lecture Notes in Computer Science*, Springer-Verlag, 1997, 248-261.
- [Purchase et al., 1997] Helen C. Purchase, Robert F. Cohen ja Murray I. James, An experimental study of the basis for graph drawing algorithms. *ACM Journal of Experimental Algorithms* **2**, 4 (1997). Available also as <http://www.acm.org/jea/ARTICLES/Vol2Nbr4.pdf>. (Tarkastettu 27.5.2002)
- [Raisamo, 1995] Roope Raisamo, *Kaksi kättä käyttöliittymässä*. Tampereen yliopisto, Tietojenkäsittelyopin laitos, Pro gradu -tutkielma, toukokuu 1995.
- [Raisamo, 1999] Roope Raisamo, An empirical study on how to use input devices to control tools in drawing programs. In: *Proceedings of OzCHI'99, 1999 Conference of the Computer Human Interaction Special Interest Group of the Ergonomics Society of Australia*, Charles Sturt University, 1999, 71-77.
- [Raisamo ja Niemi, 2000] Roope Raisamo ja Tapio Niemi, Interactive layout techniques for conceptual schema editors. In: *CD Proceedings of NordiCHI 2000*, The Swedish Interdisciplinary Interest group for Human-Computer Interaction (STIMDI), 2000.
- [Raisamo ja Rähä, 1996] Roope Raisamo ja Kari-Jouko Rähä, A new direct manipulation technique for aligning objects in drawing programs. In: *Proceedings of ACM UIST'96, Symposium on User Interface Software and Technology*, ACM Press, 1996, 157-164.

- [Raisamo ja Rähä, 2000] Roope Raisamo ja Kari-Jouko Rähä, Design and evaluation of the alignment stick. *Interacting with computers* **12** (2000), 483-506.
- [Ramloll ja Brewster, 2002] Rameshsharma Ramloll ja Stephen A. Brewster, A generic approach for augmenting tactile diagrams with spatial non-speech sounds. To appear in: *Vol. II of proceedings of ACM CHI'2002*, ACM Press, 2002.
- [Ramstein, 1995] Christophe Ramstein, Multimodal user interface system with force feedback and physical models. In: *Proceedings of IFIP International Conference Interact'95*, Lillehammer, Norway, 1995, 157-162.
- [Robbins et al., 1999] Jason Robbins, Michael Kantor ja David Redmiles, Sweeping away disorder with the broom alignment tool. In: *Proceedings of ACM CHI'99, The Human Factors in Computing Systems*, ACM Press, 1999, 250-251.
- [Robbins ja Redmiles, 1999] Jason Robbins ja David Redmiles, Cognitive support, UML adherence, and XMI interchange in Argo/UML. In: *Proceedings of the Construction of Software Engineering Tools (CoSET'99)*, 1999, 79-89.
- [Rosenberg ja Brave, 1996] Louis Rosenberg ja Scott Brave, Using force feedback to enhance human performance in graphical user interfaces. In: *Proceedings of the ACM CHI'96*, ACM Press, 1996, 291-292.
- [Ryall, 1997] Kathleen Ryall, *Computer-Human Collaboration in the Design of Graphics*. Division of Engineering and Applied Sciences, Harvard University, Cambridge, Doctoral dissertation, August 1997.
- [Ryall et al., 1997] Kathy Ryall, Joe Marks ja Stuart Shieber, An interactive constraint-based system for drawing graphs. In: *Proceedings of ACM UIST'97, Symposium on User Interface Software and Technology*, ACM Press, 1997, 97-104.
- [SensAble, 2002] SensAble Technologies Inc., <http://www.sensable.com/>. (Tarkastettu 27.5.2002)
- [Shneiderman, 1982] Ben Shneiderman. The future of interactive systems and the emergency of direct manipulation. *Behaviour and Information Technology* **1** (1982), 237-256.
- [Sjöström, 1999] Calle Sjöström, The potential of haptics: Touch access for people with disabilities. Certec, Lunds Tekniska Högskola, licentiate thesis, December 1999. Available as <http://www.certec.lth.se/doc/touchaccess/TouchAccess.pdf>. (Tarkastettu 27.5.2002)

- [Smith, 1997] Christopher M. Smith. Human factors in haptic interfaces. *ACM Crossroads Student Magazine* **3.3** (Spring 1997). Available as <http://www.acm.org/crossroads/xrds3-3>. (Tarkastettu 27.5.2002)
- [Sugiyama *et al.*, 1981] Kozo Sugiyama, Shojiro Tagawa ja Mitsuhiko Toda, Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics* **11**, 2 (Feb. 1981), 109-125.
- [Sutherland, 1963] Ivan Sutherland, Sketchpad: A man-machine graphical communication system, In: *Proceedings of the IFIP Spring Joint Conference*, 1963.
- [Szirmay-Kalos, 1994] Laszlo Szirmay-Kalos, Dynamic layout algorithm to display general graphs. *Graphics Gems IV*, Academic Press, 1994, 505-517.
- [Tamassia, 1997] Roberto Tamassia, Graph drawing. In: J. E. Goodman and J. O'Rourke (eds.), *Handbook of Discrete and Computational Geometry*. CRC Press LLC, Boca Raton, 1997, 815-832.
- [Vilis, 2000] Tutis Vilis, Touch. In: *The Physiology of the Senses Transformations for Perception and Action, Session 6: Touch*, University of Western Ontario, Canada, 2000. Available as <http://www.med.uwo.ca/physiology/courses/sensesweb/>. (Tarkastettu 27.5.2002)
- [Wall ja Harwin, 2001] Steven A. Wall ja William S. Harwin, A high bandwidth interface for haptic human computer interaction. *Mechatronics, The Science of Intelligent Machines, An International Journal* **11**, 4 (2001), 371-387.
- [Yu ja Brewster, 2002] 24. Wai Yu ja Stephen Brewster, Comparing two haptic interfaces for multimodal graph rendering. In: *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (Haptics 2002)*, IEEE Computer Society, 2002, 3-9.
- [Ziegler ja Fährnich, 1988] Jürgen E. Ziegler ja Klaus-Peter Fährnich: Direct manipulation. In: M. Helander (ed.), *Handbook of Human-Computer Interaction*, Elsevier Science Publishers B.V., 1988, 123-133.