

**Erkki Mäkinen (toim.)**

**Tietojenkäsittelytieteellisiä tutkielmia  
Syksy 2010**



TIETOJENKÄSITTELYTIETEIDEN LAITOS  
TAMPEREEN YLIOPISTO

D-2010-17

TAMPERE 2010

TAMPEREEN YLIOPISTO  
TIETOJENKÄSITTELYTIETEIDEN LAITOS  
JULKAISUSARJA D – VERKKOJULKAISUT  
D-2010-17, JOULUKUU 2010

**Erkki Mäkinen (toim.)**

**Tietojenkäsittelytieteellisiä tutkielmia  
Syksy 2010**

TIETOJENKÄSITTELYTIETEIDEN LAITOS  
33014 TAMPEREEN YLIOPISTO

ISBN 978-951-44-8344-8  
ISSN 1795-4274

## Sisällysluettelo

Kuvien tiivistämisestä.....	1
<i>Jussi Ampuja</i>	
Tactile depth cues .....	20
<i>Kalle Myllymaa</i>	
Lifeloggaus – tausta, hyödyt ja ongelmat.....	31
<i>Saila Oldén</i>	
Polunetsintä tietokonepeleissä.....	45
<i>Ville Rahikainen</i>	
Suosittelujärjestelmät.....	65
<i>Markus Sairanen</i>	
Videopelien vika? – kirjallisuuskatsaus moderniin väkivaltaisten videopelien tutkimukseen.....	79
<i>Auvo Salmenharju</i>	
Tietokoneavusteinen psykoterapia addiktioiden hoidossa.....	102
<i>Hanny Sievänen</i>	

# Kuvien tiivistämisestä

## Jussi Ampuja

### Tiivistelmä

Digitaalisen tiedon tiivistäminen tarkoittaa sen ilmaisemista pienemmällä määrällä bittejä, jolloin se on mahdollista siirtää tai tallentaa tehokkaammin. Tässä tutkielmassa luodaan katsaus tärkeimpiin kuvien tiivistyksessä käytettäviin menetelmiin.

**Avainsanat ja -sanonnat:** kuvanpakkaus, kuvien tiivistäminen

**CR-luokat:** E.4

## 1. Johdanto

Nykyajan informaatioyhteiskunnassa kaikenlaista tietoa halutaan säilöä ja siirtää valtavia määriä, usein digitaalisesti. Huolimatta teknologisesta kehityksestä, informaation siirto- ja tallennuskapasiteetti on aina rajallinen. Tästä muodostuu tarve entistä tehokkaammalle tiedon kuvaukselle.

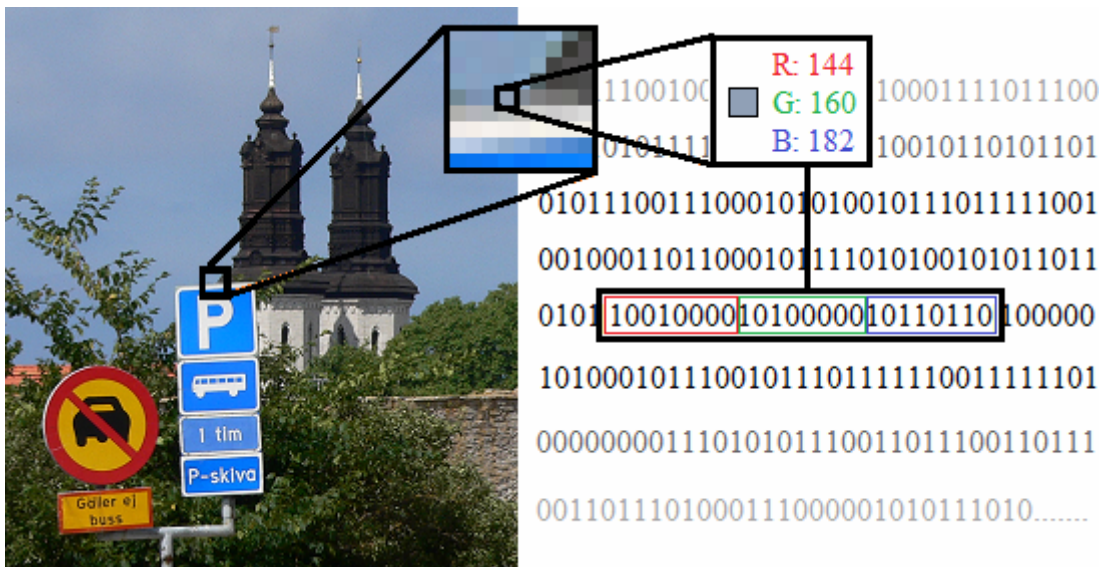
Kaikki digitaalinen informaatio koostuu perustasolla ykkösistä ja nolista, biteistä. Tiedoston esitysmuotoa muuttamalla sen vaatimaa bittimäärää on yleensä mahdollista pienentää, jolloin on kyse tiedon tiivistämisestä tai pakkaamisesta. Tiivistäminen on keskeistä varsinkin "luonnollisen" datan (kuten kuvan ja äänen) tapauksessa, koska niiden kanoninen esitys tyypillisesti tarvitsee epäkäytännöllisen suuren määrän muistia.

Digitaalinen kuva voi olla joko vektori- ja rasterityyppinen, mutta tässä tutkielmassa keskityn ainoastaan jälkimmäiseen kategoriaan. Rasterikuva, toiselta nimeltään bittikartta, koostuu kaksiulotteisesta matriisista kuvapisteitä eli pikseleitä. Termi resoluutio viittaa matriisin kokoon, joka on tapana esittää eriteltynä vaaka- ja pystyakselilla (esim. 1024x768). Mitä suurempi resoluutio, sitä tarkempia yksityiskohtia kuvassa on mahdollista esittää.

Pikseli on käytännössä värinäyte. Värisyvyys tarkoittaa yhden pikselin värin kuvaukseen käytettävien bittien määrää. Yksinkertaisimmillaan kuva koostuu kahdesta väristä, ja pikseliä kohden riittää yksi bitti – väri on joko tai. 8-bitin värisyvyydellä (jota käytetään tyypillisesti esimerkiksi harmaasävykuville) on mahdollista esittää  $2^8$  eri väriä. Useimmat nykyaikaiset näyttölaitteet kykenevät 24-bitin värisyvyyteen, jolla pystytään ilmaisemaan yli 16 miljoonaa eri värisävyä. Tämä on yleensä varsin riittävästi, koska ihmissilmä kykenee erottamaan rohkeimpien arvioiden mukaan noin 10 miljoonaa väriä. [Leong,

2006]

Ihminen muodostaa värihavaintonsa kolmen värin yhdistelmänä [Leong, 2006], joten myös värikuvissa pikselien väri ilmoitetaan kolmen värikanavan avulla. Kuvassa 1 on käytetty RGB-mallia, joka määrittelee värit punaisen (R), vihreän (G) ja sinisen (B) kombinaationa. Muita vaihtoehtoja olisivat esimerkiksi CMY- tai YUV-väriavaruudet. Jos luet tätä paperilta, tulostuksessa on käytetty CMYK-mallia. Kyseinen väriavaruus on ns. subtraktiivinen, joka tarkoittaa, että värejä lisäämällä saadaan aikaan tummempi väri (paperi on valkoista). YUV taas on esimerkiksi analogisissa TV-lähetyksissä käytetty standardi, josta lisää kohdassa 3.1.



Kuva 1: Yksittäinen pikseli kuvassa ja sitä vastaavat bitit binääriesityksessä käytettäessä 24-bitin RGB-väriavaruutta.

Tiivistämättömän kuvan vaatima muistimäärä saadaan suoraan kertomalla sen resoluutio värisyvyydellä. Voimme nopeasti laskea, että esim. "täyden teräväpiirtoresoluution" (1920x1080) kuva 24-bitin värisyvyydellä vaatii noin 6 megatavua tallennustilaa. Kahden tunnin elokuva (30 ruutua sekunnissa) tarvitsee siis noin 1251 gigatavua. Kuitenkin ihmisten ainakin oletetaan haluavan esimerkiksi katsella hyvälaatuista liikkuvaa kuvaa internet-yhteyksien välityksellä. Tässä valossa tiivistäminen ei ole ainoastaan hyödyllistä, vaan jopa jossain määrin välttämätöntä.

Kuvan kanoninen esitysmuoto (kuten kuvassa 1) sisältää onneksi lähes poikkeuksetta runsaasti redundanssia eli toistoa, joka on aina jossain määrin turhaa tietoa. Redundanssi voi olla esimerkiksi koodausredundanssia, spatiaalista eli tilallista redundanssia tai psykovisuaalista redundanssia [Suominen, 2007]. Koodausredundanssi tarkoittaa, että pikselien sisältämä data ilmaistaan jollain alioptimaalisella, tuhlaavaisella tavalla. Spatiaalista redundanssia syntyy siitä, että pikseleiden välillä on alueellista korrelaatiota, joita ei huomioida

kuvan esitysmuodossa. Psykoviisuaalinen redundanssi liittyy ihmisen havainnointikykyyn, eikä siis välttämättä matemaattisessa mielessä ole oikeaa toistuvuutta, vaan paremminkin katsojan kannalta epärelevantin tiedon välitystä.

Tiivistäminen pyrkii poistamaan kuvan binääriesityksessä ilmenevän redundanssin ja näin saavuttamaan pienempiä tiedostokokoja. Erilaisia tiivistysmenetelmiä on useita ja niiden kategorisointi vaikeaa. Yksi selkeä luokittelu-peruste on kuitenkin se, voidaanko tehty tiivistys perua ilman että yhtään informaatiota katoaa. Tällaisen jaottelun perusteella saamme kaksi pääkategoriaa: häviöttömät ja häviölliset menetelmät.

## 2. Häviöttömät tiivistysmenetelmät

Häviötön tiivistys (nimensä mukaisesti) säilyttää kaiken kuvan sisältämän informaation, ja pakattu kuva voidaan haluttaessa palauttaa täsmälleen alkuperäiseen esitysmuotoonsa. Häviöttömien pakkausmenetelmien periaatteet ovat tyypillisesti riippumattomia pakkauksen kohteesta, ja niitä voidaan soveltaa kuvien lisäksi mihin tahansa muuhunkin binääridataan. Jatkossa onkin toistuvasti käytetty havainnollistavina esimerkkeinä merkkijonoja, eikä vaikkapa kuvan pikselien väriä (mikä olisi harhaanjohtavaa, sillä usein ennen tiivistystä monivärikuvat jaetaan väritasoihin jotka pakataan erikseen [Fränti, 1994]).

### 2.1 Tilastolliset menetelmät

Tilastolliset menetelmät pyrkivät ennustamaan pakattavaa dataa kuvaamalla sen ominaispiirteitä jonkin mallin avulla. Pakkausprosessi voidaan jakaa kahteen erilliseen vaiheeseen: mallintamiseen ja koodaukseen. Mallintaminen muodostaa käsillä olevaa tietoa mahdollisimman hyvin vastaavan mallin, jota käyttäen koodaaja pyrkii kuvaamaan tiedon tiiviisti. Näistä vaiheista koodaus nähdään usein helpompana, joten keskitymme ensin siihen.

Tiivistyksen kannalta on mielenkiintoista pystyä arvioimaan pakattavan datan informaatioisisältöä. Kahden megatavun kokoinen kuvatiedosto ei useinkaan sisällä kahta megatavua tietoa – jos sisältäisi, sen häviötön tiivistäminen olisi mahdotonta. Niinkin abstraktin asian kuin tiedon mittaaminen kuulostaa vaikealta, mutta on ainakin jossain mielessä kuitenkin mahdollista.

Tiivistettävä data voidaan nähdä sarjana valintoja, jotka kohdistuvat ääreliseen joukkoon *symboleja*. Kutsumme tätä joukkoa *aakkostoksi*. Jos aakkostona on yhden alkion joukko { a }, ja valitsemme sieltä symbolin 'a', ei valinta ole yllättävä, eikä siksi informatiivinen. Tuhannen symbolin aakkostosta suoritettuna sama valinta kertoo enemmän.

Suomen kielen aakkosto kirjaimin määriteltynä sisältää 29 symbolia, { a, b, c, ..., ö }. Näistä jokaisen binäärinen esitys laajennetulla ASCII-standardilla vaatii 8 bittiä, joten 15-kirjaimisen sanan muistivaatimus tällä standardilla on aina vakio, tasan 15\*8 bittiä. Pakattavan datan symbolien esiintymistaajuudet eivät kuitenkaan juuri koskaan ole yhtä suuret – jotkut symbolit toistuvat

hyvinkin säännöllisesti, ja toiset esiintyvät vain harvoin.

Jos (ja kun) haluamme välittää tietoa mahdollisimman tiiviisti, on järkevää esittää usein esiintyvät symbolit pienemmällä määrällä bittejä kuin harvinaiset. Esimerkiksi merkkijonossa "SAIPPUAKAUPPIAS" symbolin A esiintymistodennäköisyys on 4/15, kun taas symbolin K vain 1/15. Näin ollen A kannattaa ilmaista lyhyemmällä bittijonolla kuin K. Vaikka harvinaiset symbolit saisivat alkuperäistä vakioimittaa pidemmätkin koodit, kokonaisuuden kuvaamiseen tarvittava bittimäärä yleensä vähenee.

### 2.1.1 Tietoa mittaamassa: entropia

Informaatioteoriassa *entropia* (entropy) tarkoittaa ennustamattomuutta tai epäjärjestystä. Mitä suurempi datan entropia on, sitä enemmän se sisältää informaatiota, ja sitä enemmän tarvitaan bittejä sen kuvaamiseen. [Shannon, 1948]

Symbolin todennäköisyyden perusteella sille voidaan laskea suure nimeltä *informaatiosisältö* tai *itseinformaatio* (information content, self-information). Symbolin  $x$  informaatioisisältö  $I$  saadaan kaavasta

$$I(x) = -\log_2(P(x))$$

jossa  $P(x)$  on  $x$ :n esiintymistodennäköisyys. Logaritmi on kaksikantainen, koska käsittelemme binääridataa. Suureen mittayksikkö on näin ollen bitti. Miinusmerkki on mukana vain jotta tuloksesta saadaan positiivinen. Informaatiosisältö on kääntäen verrannollinen symbolin todennäköisyyteen, joten yllättävimmät ja epätodennäköisimmät symbolit sisältävät tämän ajatuksen mukaan eniten tietoa.

Nopanheitto (painottamattomalla nopalla) voi päättyä kuudella eri tavalla. Jokaisen mahdollisen tuloksen informaatioisisältö on noin 2.58 bittiä. Sadan peräkkäisen nopanheiton tulosten kuvaamiseen tarvitsemme siis vähintään 258 bittiä. Jos muokkaamme noppaa siten, että se antaa 90 prosentilla heitoista luvun kuusi, tapahtumasarjamme entropia vähenee. Nyt voimme teoriassa odottaa pystyvämme ilmaisemaan yhden tuloksen keskimäärin 0.81 bitillä.

Mallin entropia  $H(x)$  on satunnaismuuttujan  $x$  informaatioisisällön odotusarvo (huomioitava, että  $x$  ei tässä tarkoita tiettyä symbolia):

$$H(x) = E(I(x))$$

joka voidaan aikaisemmin annetun informaatioisisällön laskukaavan avulla kirjoittaa edelleen muotoon

$$H(x) = -\sum_{i=1}^n p(x_i) \log_2 p(x_i)$$

jossa  $n$  on aakkoston koko. Yksikkö on bittiä per symboli. Mallin entropia määrittää optimaalisen määrän bittejä, jolla yksi valinta voidaan keskimäärin esittää.

Useat kirjoittajat (mukaanlukien Shannon [1948] itse) ovat kutsuneet informaation sisältöä yksinkertaisesti "symbolin entropiaksi" tekemättä käsitteiden välille sen suurempaa eroa. Toiset lähteet kuitenkin korostavat, ettei kyse ole synonyymeistä, joten olen niitä tässä käsitellyt erillisinä suureina. Informaation sisältö on yksittäisille symboleille niiden todennäköisyyden perusteella laskettu painoarvo, kun taas entropia kuvaa symbolijoukkoon kohdistuvan valinnan tuloksen epävarmuutta.

$X$	$P(x)$	$I(x)$
S	0.133	2.906
A	0.267	1.906
I	0.133	2.906
P	0.267	1.906
U	0.133	2.906
K	0.066	3.907

*Taulukko 1: Merkkijonon 'SAIPPUAKAUPPIAS' symbolit, niiden todennäköisyydet ja informaation sisältöt.*

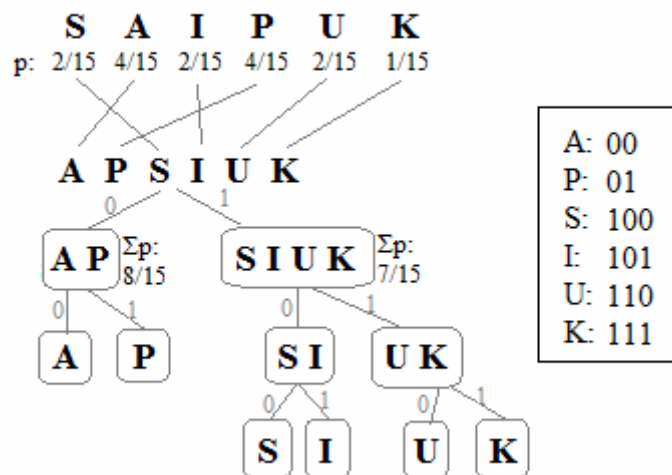
Taulukossa 1 esitetyn mallin entropiaksi saamme likimain 2.44 bittiä per symboli, joten jos merkkijono 'saippuakauppias' voitaisiin tätä mallia käyttäen koodata optimaalisesti, se mahtuisi noin 36.6 bittiin. Pystymme nyt siis laskemaan, kuinka pitkällä bittijonoilla symbolit olisi teoriassa mahdollista ilmaista. Itse koodien muodostamisessa se ei meitä valitettavasti juurikaan auta.

### 2.1.2 Shannon-Fano-koodaus

Shannon-Fano-koodaus on vanhimpia ja yksinkertaisimpia menetelmiä bittikoodien muodostamiselle. Ajatuksena on järjestää symbolit todennäköisyyden mukaan (laskevasti tai nousevasti), ja sitten jakaa ne kahtia sellaisesta kohtaa, että molempien puolien yhteenlaskettu todennäköisyys on mahdollisimman lähellä toisiaan. Muodostuneista osajoukoista toista merkataan bitillä 0 ja toista bitillä 1. Jokainen osajoukko jaetaan edelleen rekursiivisesti saman periaatteen mukaan, kunnes ne sisältävät ainoastaan yhden symbolin, ja symbolit voidaan ilmaista yksiselitteisin bittijonoin. Prosessi esitetty kuvassa 2.

Koodin yksiselitteisyys tässä tarkoittaa sitä, että purkajan on voitava erottaa, mihin symbolia kuvaava bittijono loppuu ja mistä seuraava alkaa. Oletetaan, että erillistä erotinmerkkiä ei käytetä. Tällöin mikään koodi ei saa olla toisen alkuosa. Jos jokin symboli koodataan esimerkiksi bittijonolla 01, minkään muun symbolin koodi ei voi alkaa kyseisillä biteillä ilman ilmeistä ristiriitaa.





Kuva 2: Shannon-Fano -koodien muodostaminen taulukossa 1 kuvatulle mallille.

Shannon-Fano -koodaukselle tuotettujen koodien keskimääräinen pituus on maksimissaan yhden bitin päässä mallin entropiasta. Menetelmän käyttö on kuitenkin harvinaista, koska esimerkiksi seuraavaksi käsiteltävä Huffman-koodaus ei ole juurikaan monimutkaisempi, ja sen tuottamat koodit ovat keskimäärin lyhyempiä. [Bell *et al.*, 1990]

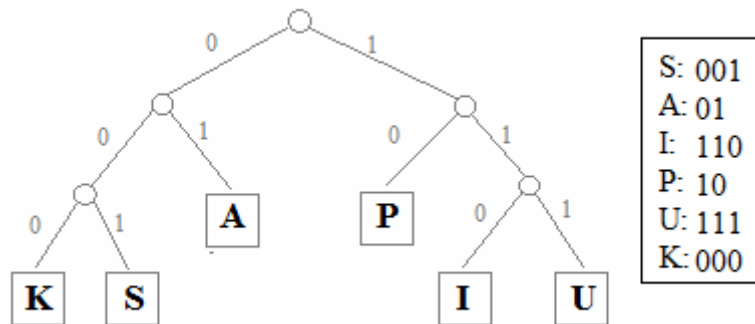
### 2.1.3 Huffman-koodaus

Huffman-koodaus rakentaa aakkostosta tietorakenteen, joka voidaan luontevasti esittää binääripuuna. Puussa symbolit ovat lehtisolmuissa, todennäköiset lähimpänä juurisolmua. Puu voidaan rakentaa esimerkiksi seuraavassa kappalessa kuvatulla tavalla ajassa  $O(n \log n)$ .

Jokaiselle symbolille luodaan solmu. Solmut sijoitetaan todennäköisyyden mukaan järjestyvään prioriteettijonoon, jossa epätodennäköisimmät tulevat ensin. Niin kauan kuin jonossa on ainakin kaksi solmua, luodaan uusi solmu, ja asetetaan sen todennäköisyydeksi kahden jonossa ensimmäisenä olevan solmun yhteenlaskettu todennäköisyys. Nämä kaksi solmua sijoitetaan uuden solmun lapsiksi ja samalla poistetaan ne jonosta. Juuri luotu solmu lisätään jonoon. Näin jonossa olevien solmujen määrä pienenee yksi kerrallaan, kunnes solmuja on enää yksi. Se on puun juurisolmu, jonka jälkeläisistä löytyvät kaikki ensimmäisessä vaiheessa luodut solmut ja niiden sisältämät symbolit.

Näin rakennetusta puusta voidaan symboleille muodostaa yksiselitteiset Huffman-koodit lähtemällä liikkeelle juuresta ja merkkäämällä vasenta kaarta ykkösellä ja oikeaa nollalla tai toisinpäin. Edellä esitetty algoritmi ei ole ainoa (eikä välttämättä edes paras) tapa puun muodostamiseen, mutta se on tässä yksinkertaisuutensa takia. Komulainen [2001] kuvaa prosessin visuaalisesti solmuja yhdistelemällä. Koodien pituuksien kannalta lopputulos on sama.

Kuvassa 3 on sanalle "saippuakauppias" muodostettu Huffman-puu ja sen avulla saadut koodit symboleille, kun bitti 0 merkitsee vasenta kaarta puussa edettäessä.



Kuva 3: Taulukossa 1 kuvatusta mallista muodostettu Huffman-puu.

Kuvia 2 ja 3 vertaamalla voimme todeta, että tässä tapauksessa sekä Huffman- että Shannon-Fano-koodaus tuottivat symboleille täsmälleen samantyyppiset koodit. Yleisesti ottaen Huffman-koodaus kuitenkin suoriutuu paremmin. Menetelmää pidettiin vuosikymmeniä jopa jossain mielessä optimaalisena, vaikka sillä useimmiten ei mallin entropiaa saavutetakaan. [Bookstein and Klein, 1993] Aritmeettisen koodauksen yleistymisen myötä käsitys optimaalisuudesta on muuttunut, mutta Huffman-koodausta käytetään silti edelleen laajalti. Usein (esimerkiksi JPEG- ja MP3-formaateissa) se on pakkausprosessin viimeinen vaihe.

#### 2.1.4 Aritmeettinen koodaus

Molemmat aikaisemmin käsitellyt menetelmät asettavat symboleille koodit, joissa tietty bittijono vastaa aina tiettyä symbolia. Aritmeettisessä koodauksessa lähestymistapa on erilainen: koko data muutetaan yhdeksi reaaliluvuksi, joka "osoittaa" symboleita koodauksen edetessä kapenevalta väliltä. Väli on jaettu symboleja kuvaaviin aliväleihin, joiden koko määräytyy symbolin todennäköisyyden mukaan. Lähtötilanteessa kokonaisväli on tyypillisesti nolasta yhteen (tai tarkemmin nolasta äärettömän lähelle yhtä), ja se kapenee jokaisen koodatun symbolin myötä symbolia kuvaavan alivälin kokoiseksi. Mitä pidempi ja entropialtaan suurempi tietomäärä, sitä tarkempi luku tarvitaan sitä kuvaamaan. Koska todennäköisimmät symbolit saavat lukuväliltä suurimmat lohkot, niiden esiintyessä väliä tarvitsee tarkentaa vain vähän, ja bittejä säästyy. Lopullinen tiiviste on oikeastaan myös väli, mutta purkaja ei tarvitse sen ylä- ja alarajoja, vaan mikä tahansa luku tuolta väliltä riittää. [Bell *et al.*, 1990]

Koodataan merkkijono **'cabaa'**  
 Todennäköisyysjakauma:  
 $p(c):0,2$   $p(a):0,6$   $p(b):0,2$   
 Entropia: 1,37095059445467 bps  
 .....

Koodattu: (ei mitään) Seuraava symboli: **'c'**  
 Väli: 0 - 1  
 Aliväli 'c': [0, 0,2)  
 Aliväli 'a': [0,2, 0,8)  
 Aliväli 'b': [0,8, 1)  
 .....

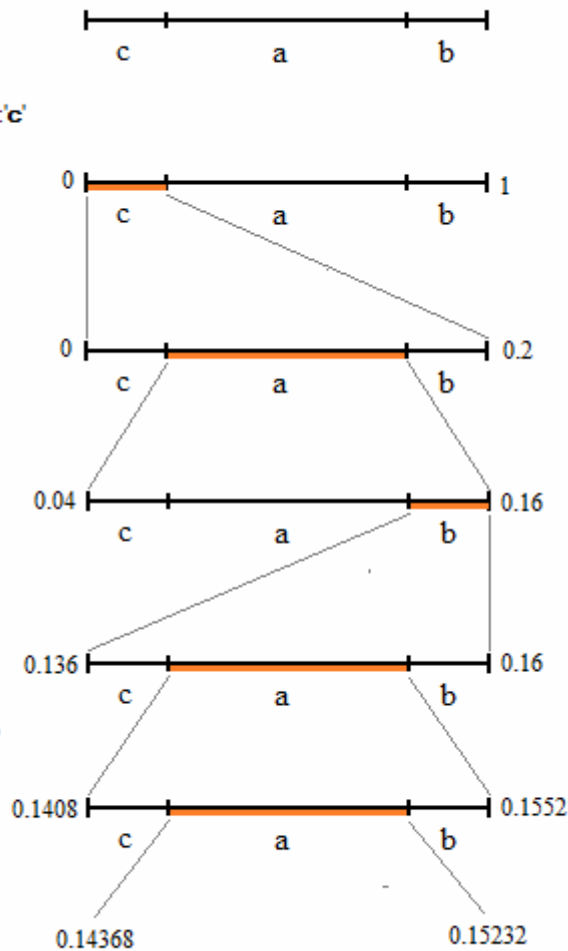
Koodattu: **'c'** Seuraava symboli: **'a'**  
 Väli: 0 - 0,2  
 Aliväli 'c': [0, 0,04)  
 Aliväli 'a': [0,04, 0,16)  
 Aliväli 'b': [0,16, 0,2)  
 .....

Koodattu: **'ca'** Seuraava symboli: **'b'**  
 Väli: 0,04 - 0,16  
 Aliväli 'c': [0,04, 0,064)  
 Aliväli 'a': [0,064, 0,136)  
 Aliväli 'b': [0,136, 0,16)  
 .....

Koodattu: **'cab'** Seuraava symboli: **'a'**  
 Väli: 0,136 - 0,16  
 Aliväli 'c': [0,136, 0,1408)  
 Aliväli 'a': [0,1408, 0,1552)  
 Aliväli 'b': [0,1552, 0,16)  
 .....

Koodattu: **'caba'** Seuraava symboli: **'a'**  
 Väli: 0,1408 - 0,1552  
 Aliväli 'c': [0,1408, 0,14368)  
 Aliväli 'a': [0,14368, 0,15232)  
 Aliväli 'b': [0,15232, 0,1552)  
 .....

Tulos : [0,14368, 0,15232)  
 Binääriesitys 0.0010011 kelpaa.  
 Pituus: 7 bittiä, 1,4bps.



Kuva 4: Aritmeettisen koodauksen eteneminen tiivistettäessä merkkijonoa "cabaa".

Kuvassa 4 vasemmalla on kirjoittamani yksinkertaistetun aritmeettisen merkkijonokoodaajan tuloste ja vieressä koodauksen eteneminen skaalautuvan janan avulla kuvattuna. Vertailun vuoksi aikaisemmin käytetty merkkijono "saippuakauppias" pakkautui 37 bittiin, eli mallin entropian asettama alaraja (pyöristettynä kokonaisiin bitteihin) saavutettiin.

Purkaja suorittaa melko samanlaisen operaation. Myös se luo välin, aluksi nolasta yhteen, ja jakaa sen edelleen aliväleihin todennäköisyysjakauman perusteella. Tiivisteestä saatava luku osuu jollekin näistä aliväleistä, josta voidaan päätellä datan ensimmäinen symboli. Väliä kavennetaan ja sama toistetaan.

Taulukosta 1 näemme, että koodien informaatioisisältö on harvoin luonnollinen luku. Aikaisemmin käsitellyt menetelmät voivat asettaa koodeiksi vain kokonaislukuja, kuten 3 tai 4 bittiä, mutta ei mitään siltä väliltä. Esimerkiksi Huffman-koodauksella onkin mahdollista saavuttaa mallin entropia vain, jos symbolien todennäköisyydet sattuvat olemaan  $\frac{1}{2}^n$  potensseja. Aritmeettisen

koodauksen tärkeimpiä vahvuuksia on, että se ei aseta symboleille min-käänlaisia määrättyjä koodeja, eikä siksi välttämättä tarvitse kokonaisia bittejä niiden kuvaamiseen ("puolikkaita bittejä" ei tietenkään ole mahdollista käyttää, mutta yksi bitti voi kuvata kaksikin symbolia jne.) Näin päästään usein hyvin lähelle Shannonin teoreettista alarajaa tarvittavien bittien määrässä.

Kuvan 4 perusteella on nähtävissä (ainakin) kaksi potentiaalista ongelmaa, jotka on kuitenkin suhteellisen helppo välttää. Ensinnäkin, kun tiivistettä puretaan, on mahdotonta tietää, milloin kuuluu lopettaa (kuvan mallilla 'c' ja "cccccccccccccc" voidaan molemmat ilmaista luvulla 0.0). Tämä voidaan korjata joko kirjoittamalla tiivisteeseen erikseen, montako symbolia se sisältää, tai lisäämällä aakkostoon erillinen lopetusmerkki. Toinen ongelma on se, että symbolimäärän ja datan pituuden kasvaessa tarvitaan hyvin nopeasti niin tarkkoja lukuja, että ohjelmointikielten tarkimmatkaan sisäänrakennetut tietotyypit eivät pysty niitä ilmaisemaan. Sekä tiivistäminen että purkaminen voidaan kuitenkin tehdä kokonaan käyttäen rajallisella tarkkuudella toimivia muuttujia. [Langdon, 1984]

Vaikka aritmeettisen koodauksen käyttö on viimeaikoina yleistynyt, ei se ole kuitenkaan täysin syrjäyttänyt Huffman-koodausta. Bookstein ja Klein [1993] löytävät tähän useita syitä: aritmeettisen koodauksen vaatima laskentateho on heidän mukaansa huomattavasti suurempi kuin Huffman-koodauksessa, eikä pakkaussuhde käytännössä useinkaan silti ole merkittävästi parempi. Lisäksi he näkevät aritmeettisen koodauksen periaatteet vaikeatajuisina ja toteutuksenkin hankalana. Käytännössä suurempi este saattavat kuitenkin olla aritmeettista koodausta suojanneet patentit.

### 2.1.5 Mallintamisesta

Usein koodausmenetelmää merkittävämpää on se, millaista mallia datasta käytetään. Käytännössä malli tarkoittaa koodauksessa käytettävää symbolien todennäköisyysjaumaa. Jos tämä jakauma on kovin kaukana todellisuudesta, ei mallin suhteen optimaalinenkaan koodaus tuota hyvää tulosta. Mallintaminen on prosessi, jolla malli luodaan. Pakkaajalle tehtävä on periaatteessa helppo: todennäköisyydethän voi laskea! Käytännössä asia on kuitenkin mutkikkaampi.

Jotta tieto pystytään purkamaan, täytyy purkajalla olla siitä täsmälleen samanlainen malli kuin pakkaajalla. On tietenkin mahdollista sisällyttää symbolitodennäköisyydet tai niistä muodostetut tietorakenteet mukaan tiedostoon, jolloin kyse on ns. *semi-adaptiivisesta* mallintamisesta. Tällaisen menetelmän käyttökelpoisuus riippuu pitkälti em. tietorakenteiden koosta – mallin kuvaus voi helposti viedä enemmän tilaa kuin tiivistys sitä säästää. Lisäksi näin tehtäessä pakkaajan täytyy käydä data lävitse kahteen kertaan (ensimmäinen kerralla lasketaan todennäköisyyksiä), mikä voi joissain käyttökonteksteissa olla ongelmallista.

Jos tarkkoja symbolitodennäköisyyksiä ei tiedetä, täytyy malli muodostaa muulla tavoin. Yksinkertaisin vaihtoehto on, että pakkaaja ja purkaja käyttävät

etukäteen sovittua, *staattista* mallia, jossa todennäköisyydet noudattavat jotain oletettua jakaumaa riippumatta tiivistettävän datan todellisesta sisällöstä. Esimerkiksi tekstejä tiivistettäessä staattinen malli voi olla hyvä ratkaisu – suomen, englannin ja useiden muiden luonnollisten kielten symbolien esiintymistaajuudet on tilastoitu suhteellisen tarkasti, ja lähes kaikki niillä kirjoitetut tekstit noudattavat ainakin likimain näitä todennäköisyysjakaumia. Koska meitä kuitenkin tällä kertaa kiinnostavat kuvat, on staattinen malli usein huono valinta; on vaikea perustellusti väittää, että vaikkapa jokin tietty sinisen sävy olisi aina muita värejä todennäköisempi.

Toinen, yleisesti ottaen staattista parempi, mallintamistapa on *adaptiivinen mallintaminen*. Adaptiivisessa lähestymistavassa dataa ei aluksi edes pyritä ennustamaan kovin tarkasti, vaan siitä kerätään lähinnä tilastoja. Alkutilanteessa käytetään jotakin (mahdollisesti huonoa) staattista mallia, jota sekä pakkaaja että purkaja päivittävät jokaisen kohtaamansa symbolin myötä. Koodauksen edetessä adaptiivinen malli siis "sopeutuu" käsillä olevaan tietoon, ja pakkausteho paranee loppua kohden. Näin sekä pakkaaja että purkaja pystyvät – toisistaan mitään tietämättä – ylläpitämään samassa vaiheessa koodausta täysin samanlaista mallia. Hyvin lyhyitä tietomääriä pakattaessa adaptiivinen mallintaminen ei välttämättä tuota kovin hyviä tuloksia, koska sopeutumista ei juuri ehdi tapahtua. Tieto täytyy kuitenkin käydä läpi vain kerran, eikä todennäköisyyksiä tarvitse kommunikoida datan mukana.

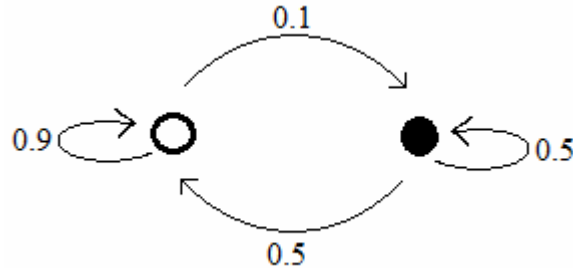
Adaptiivisen mallin käyttö aritmeettisen koodauksen yhteydessä on luontevaa, koska jokaisen esiintyvän symbolin myötä tarvitsee vain päivittää todennäköisyyksiä. Alivälien koot lasketaan joka tapauksessa joka askeleella uudestaan. Huffman-koodauksen adaptiivinen soveltaminen käytännössä vaatii tarkoitusta varten kehitettyä variaatiota algoritmista [Gallager, 1978]. Puu voitaisiin tietenkin rakentaa uudelleen jokaisen symbolin jälkeen, mutta se olisi hidasta.

### 2.1.6 Kontekstit

Adaptiivisessa mallissa käytettyä ajatusta, jonka mukaan todennäköisyysjakautta on mahdollista muuttaa kesken koodauksen, voidaan viedä vielä pidemmälle. Tähän asti symboleja on käsitelty naapureistaan riippumattomina satunnaismuuttujina, ja mallimme on ollut ns. muistiton (discrete memoryless source, DMS). Käytännössä kuvissa on kuitenkin aina vahvoja korrelaatioita lähekkäisten pikseleiden välillä. Sinistä pikseliä seuraa todennäköisesti toinen sininen pikseli, tai merkkijonojen tapauksessa jonoa 'Tamper' merkki 'e'. Tätä tietoa on mahdollista hyväksikäyttää tiivistyksessä muuttamalla symbolien todennäköisyysjakaumaa niitä edeltävien symbolien, *kontekstin*, mukaan. Jos siis kontekstin perusteella on syytä olettaa seuraavan symbolin olevan 'e', asetetaan tässä kohtaa sille normaalia korkeampi todennäköisyys. Tällaisen valistuneen arvauksen osuessa oikeaan, symbolin kuvauksesta selvittää vähemmällä biteillä.

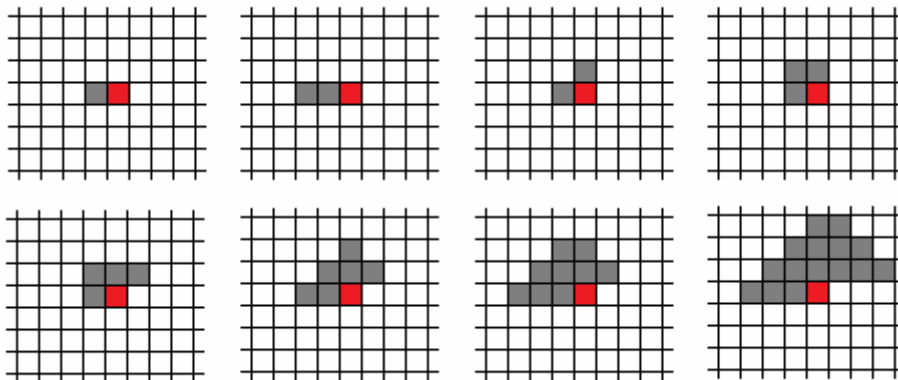
Konteksteja voidaan mallintaa erilaisilla tilakaavioilla, esimerkiksi Marko-

vin ketjuilla. Kuvan 5 esittämässä tilanteessa on mustia (musta ympyrä) ja valkoisia (valkoinen ympyrä) pikseleitä. Valkoista pikseliä seuraa 90% todennäköisyydellä toinen valkoinen pikseli ja 10% todennäköisyydellä musta pikseli. Mustaa seuraavalla pikselillä on yhtä suuri mahdollisuus olla valkoinen tai musta.



*Kuva 5: Valkoisen tai mustan pikselin todennäköisyys perustuen edellisen pikselin väriin.*

Kuvan 5 esittämä malli on varsin yksinkertainen ensimmäisen asteen Markovin ketju, jossa valinta on riippuvainen pelkästään edellisestä valinnasta. Silti, jos tiivistämme 8-bittistä värikanavaa jossa mahdollisia tiloja on 256 kappaletta, saadaan ensimmäisenkin asteen mallista varsin monimutkaisen näköinen.  $N$ :n asteen mallissa, jossa aakkoston koko on  $k$ , tiloja on  $k^N$  kappaletta. Teoreettinen muistivaatimus siis kasvaa eksponentiaalisesti kontekstin pituuteen nähden.



*Kuva 6: Mahdollisia konteksteja punaiselle pikselille, kun kuvaa käydään läpi vasemmalta oikealle ja ylhäältä alas.*

Kuvat ovat kaksiulotteisia, joten niiden käsitteleminen yksiulotteisena sarjana symboleita ei usein ole järkevää. Sen sijaan, että konteksti muodostettaisiin tietystä määrästä edellisiä symboleita, kannattanee siihen valita tilallisesti lähekkäin sijaitsevia pikseleitä. Oletetaan, että kuvaa prosessoidaan vasemmalta oikealle ja ylhäältä alas. Niinpä jokaisen ei-reuna-alueella sijaitsevan pikselin kohdalle tultaessa tunnetaan jo neljä sen kahdeksasta välittömästä

naapuripikselistä, joten niiden tai jonkin niiden kombinaation käyttäminen kontekstina on luontevaa.

## 2.2 Sanakirjamenetelmät

Toisin kuin tilastolliset menetelmät, jotka koodaavat yksittäisiä symboleja, sanakirjamenetelmät (dictionary methods) pyrkivät koodaamaan kokonaisia symbolijonoja. Koodaus voi tarkoittaa esimerkiksi jonon korvaamista indeksiviittauksella jonkinlaiseen usein toistuvia fraaseja sisältävään sanakirjaan (mistä siis nimi "sanakirjamenetelmät").

### 2.2.1 Jakson pituuden koodaus

*Jakson pituuden koodaus* (run length encoding, RLE) on perusajatukseltaan hyvin yksinkertainen tiivistysmenetelmä. Mikäli datassa esiintyy useampi samanlainen symboli peräkkäin, voidaan koko tämä jakso kuvata yhdellä kappaleella tätä symbolia ja jakson pituudella. Esimerkiksi merkkijono 'aaaaaaaaabbbbaab' koodattuna näin voisi näyttää seuraavalta: '9a3b2a1b'. Jos koodattavassa aakostossa on vain kaksi symbolia (kaksivärinen kuva), ei edes symbolia tarvitse koodata, pelkät jakson pituudet riittävät kuvaamaan värien vuorottelua.

Kuten edellisen kappaleen esimerkistä on nähtävissä, kahden symbolin mittaiset jaksot eivät välttämättä lyhene ja yhden mittaiset peräti laajenevat. Tämä on tiivistämisen kannalta valitettavaa, ja voidaan välttää esimerkiksi siten, että alle kolmen symbolin mittaisia jaksoja ei koodata, ja koodatut symboli-lukumäärä -parit merkitään nk. pakokoodilla. Koska usein saman väriset pikselit ovat keskittyneet lähekkäin muutenkin kuin vaakasuoraan, RLE saattaa hyötyä epätavallisesta pikseleiden läpikäyntijärjestyksestä. Yksittäisten pikselien sijaan voidaan myös koodata lohkojen (esim. 2x2 pikseliä) jaksoja. Menetelmästä on kehitetty jopa häviöllinen versio, jossa pikselin väristä jätetään osa vähiten merkitsevistä biteistä huomiotta.

Pelkästään jakson pituuksia koodaamalla tiivistyssuhteet eivät useinkaan yllä hienostuneemmilla metodeilla saavutettuihin lukuihin, mutta toteutus on helppoa ja sekä pakkaaminen että purkaminen nopeaa. Menetelmä tiivistää tehokkaasti kuvia, joissa on paljon laajoja yksivärisiä alueita (ruutukaappaukset, faksit yms.) Luonnollisia valokuvia pelkästään RLE:llä pakattaessa tulokset jäävät usein vaatimattomiksi, koska pikselistä toiseen toistuvia arvoja ei välttämättä juurikaan esiinny.

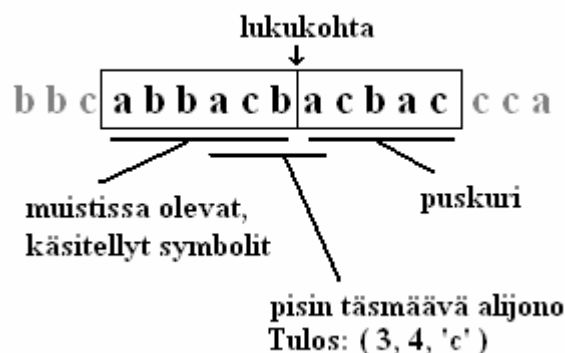
### 2.2.2 Lempel-Ziv

Lempel-Ziv on yleisnimitys suurehkolle joukolle menetelmiä, jotka perustuvat vuosina 1977 ja 1978 julkaistuihin algoritmeihin (LZ77 ja LZ78 em. vuosilukujen mukaan). Yhteistä näille algoritmeille on periaate, että datassa useammin kuin kerran toistuva symbolijono voidaan myöhemmin korvata viittauksella aikaisempaan esiintymään. Sanakirjana siis toimii jo käsitelty data. Suurin osa

Lempel-Ziv -variaatioista pystyy pakkaamaan tietoa melko tehokkaasti. [Bell *et al.*, 1990]

LZ77 on ensimmäinen (1977) Lempel-Ziv algoritmi. Se käy dataa läpi symboli kerrallaan pyrkien löytämään jo prosessoidusta datasta symbolijonoja, joihin viittaamalla voidaan ilmaista mahdollisimman suuri määrä edessä olevia symboleita. Säästääkseen muistia ja laskentaresursseja LZ77 käyttää ns. liukuva ikkuna -ajatusta, jossa dataa pidetään pakkaajan käytettävissä vain tietty määrä lukukohtasta eteen- ja taaksepäin.

Ikkunasta etsitään pisin puskuriin (lukukohtaa seuraaviin symboleihin) täsmäävä, ennen lukukohtaa alkava symbolijono. Tiivisteseen kirjoitetaan kolmiosainen muotoa  $(i, j, k)$  oleva koodi, jossa  $i$  on pisimmän täsmäävän alijonon alkuindeksi (laskien taaksepäin lukukohtasta),  $j$  alijonon pituus ja  $k$  ensimmäisen puskurissa oleva symboli, joka ei täsmännyt. Lukukohtaa siirretään  $j+1$  symbolia eteenpäin ja operaatio toistetaan. Tuloksena on sarja tiukasti vuorottelevia osoitin-pituus-literaali -kolmikoita, joita on useissa kirjoituksissa havainnollistettu kuvaamalla ne käskynä: "mene  $i$  askelta taaksepäin, kopioi sieltä  $j$  kappaletta symboleja ja lisää perään symboli  $k$ ". [Bell *et al.*, 1990]



Kuva 8: LZ77:n "liukuva ikkuna".

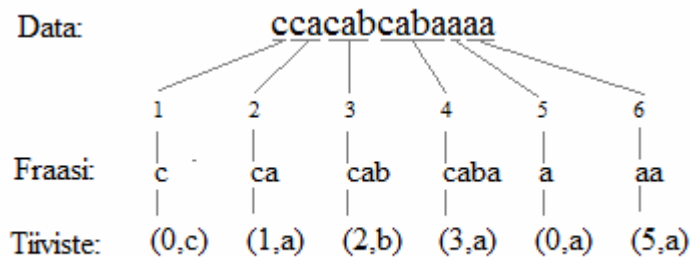
Kuvan 8 tilanteessa on käsitelty symbolijono "abbacb", ja puskurissa on jono "acb". Puskurin alkuun täsmäävät jo käsitellyn datan kolme viimeistä merkkiä, "acb". Kuvassa pisimmän täsmäävän alijonon pituudeksi on kuitenkin saatu neljä. Tämä johtuu siitä, että kun nuo kolme ensimmäistä merkkiä on purettu, on käsitelty data kasvanut kolme merkkiä pidemmäksi, ja sattuu nyt täsmäämään myös puskurin seuraavaan symboliin 'a', joten se voidaan sisällyttää tulokseen.

Käytetyn ikkunan koko on kompromissi tiivistystehon ja suoritusajan välillä. Suuri määrä muistettavia symboleja nostaa pisimmän alijonon etsimiseen tarvittavien symbolivertailujen määrää ja hidastaa prosessia. Toisaalta pienempi ikkuna tarkoittaa pienempää sanakirjaa ja siten myös huonompaa pakkaus-tehoa.

LZ78 on toinen Lempel-Ziv -algoritmien päähaara. LZ78 ei käytä liukuvaa



ikkunaa, vaan data jäsennetään fraaseiksi, jotka tallennetaan algoritmin ylläpitämään erilliseen fraasisanakirjaan. Symboleita luetaan lukukohdasta eteenpäin tasan niin pitkälle, että yksikään sanakirjan fraasi ei enää täsmää pusku-roituun dataan. Tiivisteeseen (ja sanakirjaan) lisätään muotoa  $(i, j)$  oleva merkintä, kun  $i$  on viimeisen täsmänneen fraasin indeksi sanakirjassa ja  $j$  viimeinen datasta luettu symboli, joka ei enää täsmännyt. Jos yhtään täsmäävää fraasia ei löytynyt,  $i$ :ksi kirjoitetaan nolla. Lukukohtaa siirtyy puskurin luetun symbolijonon pituuden verran eteenpäin.



Kuva 9: Merkkijonon "ccacabcabaaaa" tiivistäminen LZ78:lla.

Erillinen sanakirja poistaa tarpeen ikkunalle, jolloin osoittimet voivat viitata datassa kuinka kauas taaksepäin tahansa (eivät kuitenkaan mihin kohtaan tahansa, ainoastaan fraasien alkuun). Sama symbolijono voidaan jäsentää fraaseiksi useammalla eri tavalla. Edellä esitetty vaihtoehto on ahne (greedy) lähestymistapa, joka ei pakkaustehon kannalta välttämättä ole paras. Optimaalisen jäsenysstrategian löytäminen olisi kuitenkin hidasta, ja käytännössä kaikki algoritmin sovellukset käyttävät ahnetta jäsenystä [Bell *et al.* 1990].

LZ78 pyrkii vähentämään LZ77:n suorituskykyä haittaavien symbolijonovertailujen määrää. Fraaseille on silti syytä luoda jonkinlainen tietorakenne, josta täsmäävät jonot löytyvät nopeasti. Kokeellisesti toteuttamani LZ78 pakkaaja nimittäin prosessoi 60 kilotavun tiedostoa lähes kolme minuuttia. Ajasta suurin osa kului täsmäävien fraasien etsimiseen järjestämättömästä listasta.

### 3. Häviölliset tiivistysmenetelmät

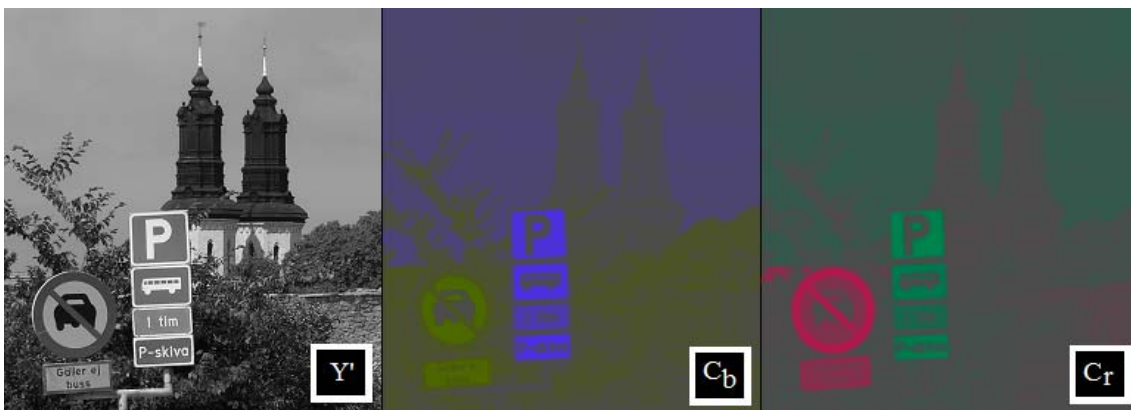
Tähän mennessä käsitellyt menetelmät sisällyttävät tarkasti kaiken kuvan sisältämän informaation tiivisteeseen. Katsojan kannalta on kuitenkin harvoin oleellista, onko vaikkapa valokuvan jokainen pikseli täsmälleen oikean värinen. Häviöllinen tiivistys pyrkii poistamaan kuvasta myös psykovisuaalisen redundanssin, säilyttäen vain katsojalle relevantin aineksen. Häviöllisellä tiivistyksellä saavutetut pakkaussuhteet näyttäisivät tyypillisesti olevan moninkertaisesti parempia kuin häviöttömällä.

Osa tiedosta siis hukataan peruuttamattomasti. Tärkeä kysymys onkin: mikä osa? Häviöllisiä menetelmiä arvioitaessa on otettava huomioon myös aina jossain määrin subjektiivinen mittari, tiivistetyn kuvan laatu. Ongelmaksi muo-

dostuu lähinnä mahdollisimman huomaamattoman (ja siksi tarpeettoman) informaation tunnistaminen.

### 3.1 Krominanssikanavien alinäytteistäminen

Tapaa, jolla ihminen havainnoi kuvia, on käytetty hyväksi kuvasignaalin pakkauksessa jo kauan ennen digitaalisen tekniikan yleistymistä. Yksi RGB-mallille vaihtoehtoinen tapa värin kuvaamiseen on Y'CbCr-malli, joka on digitaalinen vastine analogiselle YUV-mallille. Kirjainyhdistelmässä Y' tarkoittaa luminanssia eli valoisuutta ja C krominanssia eli väri vaihteluita, Cb sinisessä ja Cr punaisessa värissä.



Kuva 10: Y'CbCr-värikanavat eroteltuna.

Koska ihmisen silmä on huomattavasti herkempi vaihteluille kirkkaudessa kuin väreissä, voidaan kuvan krominanssi-informaatiosta poistaa melko suuri osa (esim. puolet) ilman minkäänlaista havaittavaa eroa kuvanlaadussa. Käytännössä alinäytteistys tarkoittaa krominanssikanavan resoluution laskua, eli useammalle luminanssikanavan pikselille käytetään samaa krominanssiarvoa.

Kuvassa 11 on krominanssikanavan resoluutiota laskettu. Toisessa näytteessä (keskellä ylhäällä) krominanssikanava sisältää enää neljäsosan luminanssikanavan pikseleistä, mutta eroa alkuperäiseen (vasemmalla ylhäällä) on vaikea havaita. Pelkästään tämän operaation avulla kuvan esittämiseen tarvittavat bitit vähenevät noin puoleen ilman merkittävää eroa kuvanlaadussa. Sarjan loput kuvat sisältävät jo huomattavia vääristymiä, varsinkin alareunan liikennemerkkien alueella.



Kuva 11: Kuvan krominanssikanavien alinäytteistystä. Ylhäällä vasemmalla alkuperäinen kuva. Alhaalla oikealla krominanssi-informaatiosta hävitetty 99.9%.

Krominanssikanavan alinäytteistämisen pääasiallinen haittavaikutus on värien “vuotaminen”, jota syntyy varsinkin terävien reunojen läheisyyteen. Laajamittaista vuotamista nähtävissä esimerkiksi kuvan 11 viimeisessä kohdassa ajokieltomerkin läheisyydessä. Näyte on otettu merkin punaisesta reunuksesta ja levitetty liian laajalle. Toinen mahdollinen haittavaikutus on krominanssin siirtyminen, jota voi tapahtua, jos kuvaa pakataan ja puretaan toistuvasti erilaisilla asetuksilla.

### 3.2 Fraktaalitiivistys

Fraktaalitiivistys on Barnsleyn ja Sloanin [1988] kehittämä varsin omaperäinen näkemys kuvan tiivistämisestä. Fraktaalipakkaaja etsii kuvasta sisäkkäisiä samankaltaisuuksia, eli alueita, jotka voidaan kuvata soveltamalla jonkinlaista muunnosta johonkin toiseen alueeseen. Yksinkertaisimmillaan tällainen muunnosoperaatio voi olla esimerkiksi koon muuttaminen, kääntäminen tai peilaus. Näistä operaatioista muodostetaan ns. fraktaalikoodeja, joiden avulla kuva on lopulta määritetty.

Fraktaalipakkauksella on mahdollista saavuttaa hyviä tiivistyssuhteita ja (varsinkin niihin nähden) laadukkaita kuvia. Koska fraktaaliesitys ei varsinaisesti määrittele yksittäisiä pikseleitä, kuva kärsii koon muuttamisesta huomattavasti vähemmän kuin normaali bittikartta, eikä esimerkiksi pikselöitymistä esiinny ylöspäin skaalauksen yhteydessä lainkaan. Tämä ominaisuus

tekee menetelmästä kuvankäsittelyn kannalta mielenkiintoisen muussakin kuin tilansäästötarkoituksessa.

Pakkausprosessin kesto on fraktaalitiivistyksen suurin kompastuskivi. Koska transformaatioissa käytettävien samankaltaisuuksien etsiminen on laskennallisesti erittäin vaativaa, ovat menetelmän käytännön sovellukset olleet harvinaisia [Blake and Wein, 1996]. Fraktaalitiivistyksen mahdollisuuksia kuitenkin tutkitaan yhä aktiivisesti, varsinkin kun laskentaresursseja vaativa pakkausvaihe ei ole nykyaikaisilla laitteilla enää yhtä suuri este kuin aikaisemmin. Esimerkiksi Erra [2005] on tutkinut mahdollisuutta suorittaa pakkaus käyttäen yksinomaan grafiikkapiirin suoritinta. Hän raportoi sen suoriutuneen tiivistyksen vaativimmasta vaiheesta noin 280 kertaa nopeammin kuin koneen keskusprosessori.

### 3.3 Taajuusmuunnokset

Sen sijaan, että kuvaa käsiteltäisiin sarjana vierekkäisiä värinäytteitä, se voidaan nähdä pikselistä toiseen jatkuvana signaalina. Taajuusesityksessä pikselien arvot kuvaavat vaihteluita kuvasignaalin amplitudissa – käytännössä värikanavan voimakkuudessa.

Taajuusmuunnokset (tarpeeksi suurella tarkkuudella suoritettuna) eivät sinänsä ole häviöllisiä operaatiota, eivätkä oikeastaan edes tiivistä kuvaa – päinvastoin. Siirtymä tila-avaruudesta taajuusavaruuteen on kuitenkin tiivistyksen kannalta hyödyllistä, koska taajuusesityksestä on helpompi poistaa psykoviuaalista redundanssia. [Clarke, 1985]

Diskreetin Fourier'n muunnoksen avulla signaali voidaan esittää siniaal-toisten funktioiden summana. Esimerkiksi JPEG-formaatti soveltaa diskreettiä kaksiulotteista kosinimuunnosta 8x8 pikselin osioille, jolloin suurin osa signaalin "energiasta" keskittyy osioiden vasempaan yläkulmaan. Ihmissilmä havaitsee huonosti nopeita taajuusvaihteluita suhteellisen pienellä alueella. JPEG hyödyntää tätä pyöristämällä muunnoksella saadut arvot nk. kvantisatiomatriisin avulla, jolloin valtaosa pikseleistä saa hyvin pieniä kokonaislukuarvoja, (usein esim. 0) ja ne on helppo lopuksi pakata tiiviisti jollakin häviöttömällä menetelmällä. Erilaisten Fourier'n muunnoksesta johdettujen tranformaatioiden lisäksi siirtymä voidaan toteuttaa esimerkiksi aallokemuunnoksella (wavelet transform) tai Karhunen-Lóeve -muunnoksella. [Wallace, 1992]

## 4. Yhteenveto

Kuvia on montaa eri tyyppiä, ja parhaan tiivistysstrategian valinta riippuu vahvasti kuvan ominaispiirteistä ja käyttötarkoituksesta. Yksinkertainen kaksivärikuva saadaan pakattua nopeasti ja tiiviisti pelkästään jakson pituuden koodauksella. Luonnolliset värivalokuvat toisaalta vaativat hyvin erilaista lähestymistapaa, ja jonkinlainen häviöllisiä vaiheita sisältävä menetelmien yhdistelmä lienee usein varteenotettavin vaihtoehto.

Yleisessä mielessä hyvät tiivistysmenetelmät, kuten erilaiset Lempel-Ziv-johdannaiset jotka pakkaavat suhteellisen tehokkaasti mitä tahansa tietoa, eivät välttämättä ole kuvien kannalta erityisen käyttökelpoisia. Ne eivät millään tavalla huomioi kuvien kaksiulotteista luonnetta [Fränti 1994], eikä niitä nähdäkseni ole kovin helppoa muokata tähän tarkoitukseen sopivammiksi. Toisaalta, jos häviöttömyys on ehdoton vaatimus, on parempaakaan vaihtoehtoa vaikea keksiä.

Kiinnostavana, mahdollisesti laajemmankin tutkimisen arvoisena menetelmänä haluaisin mainita fraktaalitiivistyksen. Tiivistyssuhteet verrattuna kuvan laatuun vaikuttavat erityisen hyviltä, ja reaaliaikainen pakkaus alkaa olla mahdollista tietokoneiden laskentatehon kasvaessa.

## Viitteet

- [Barnsley and Sloan, 1988] Michael F. Barnsley and Alan D. Sloan, Better way to compress images. *Byte* (Jan. 1988), 215–223.
- [Bell *et al.*, 1990] Timothy C. Bell, John G. Cleary and Ian H. Witten, *Text Compression*. Prentice-Hall, 1990.
- [Blake and Wein, 1996] I. F. Blake and C. J. Wein, On the performance of fractal compression with clustering. *IEEE Transactions on Image Processing* **5**, 522–526.
- [Bookstein and Klein, 1993] A. Bookstein and S. T. Klein, Is huffman-coding dead?. *Computing* **50** (1993), 279–296.
- [Clarke, 1985] R. J. Clarke, *Transform Coding of Images*. Academic press, 1985.
- [Erra, 2005] Ugo Erra, Toward real time fractal image compression using graphics hardware. In: G. Bebis, R. Boyle, D. Koracin and B. Parvin (eds.), *Advances in Visual Computing 2005*. Springer-Verlag, 2005, 723–728.
- [Fränti, 1994] Pasi Fränti, *Block Coding in Image Compression*. Ph.D. Thesis, University of Turku, 1994.
- [Gallager, 1978] Robert G. Gallager, Variations on a theme by Huffman. *IEEE Transactions on Information Theory* **24** (Oct. 1978), 668–674.
- [Huffman, 1952] David Huffman, A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* **40**, 1098–1101.
- [Komulainen, 2001] Kaisa Komulainen, *Harmaasävykuvien häviötön tiivistäminen*. Pro gradu -tutkielma, Joensuun yliopisto, Tietojenkäsittelytieteiden laitos, 2001.
- [Langdon, 1984] Glen G. Langdon, An introduction to arithmetic coding, *IBM Journal of Research and Development* **28**, 135–149.
- [Leong, 2006] Jennifer Leong, Number of colors distinguishable by human eye. In: Glenn Elert (ed.), *The Physics Factbook*, 2006. URL: <http://hypertextbook.com/facts/2006/> (v. 20.12.2010).
- [Shannon, 1948] Claude Elwood Shannon, A mathematical theory of commu-

nication. *The Bell System Technical Journal* **27**, 379–423, 623–656.

[Suominen, 2007] Svante Suominen, *Kuvatekniikan perusteet*. Luentokalvoja, Teknillinen korkeakoulu, 2007.

[Wallace, 1992] Gregory K. Wallace, The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* **38**, 18–34.

# Tactile depth cues

Kalle Myllymaa

## Abstract.

This paper presents a technique for rendering depth information on touchscreens using tactile feedback. Because of recent advances in technology, mobile devices are used more and more often to view images. Humans cannot always rely on the depth information received from pictures due to optical illusions. The proposed technique provides additional depth cues for users allowing them to perceive depth more accurately.

**Keywords:** Haptics, rendering.

**CR-classification:** H5.2

## 1. Introduction

Touchscreens are used widely in smartphones and other mobile devices. Because of built-in cameras and good quality displays such devices are often used for viewing pictures. Still, two-dimensional pictures combined with relatively small displays have their limitations. Sometimes optical illusions make it extremely difficult to judge distances in pictures.

Vibration motors or other tactile feedback methods are often used in mobile devices to enhance the user experience and to give feedback in such situations where audio or visual feedback is not sufficient. This tactile feedback is also used to give a more natural feeling when using touchscreen keyboards [Hoggan et al., 2008] and presenting six-dot-braille [Rantala et al., 2009].

This paper presents a technique for utilizing these tactile feedback methods in depth perception. The second chapter takes a look into the previous work done with tactile feedback on touchscreens and other devices along with a brief outline of some other haptic methods for rendering depth. Depth perception and haptic interaction is presented in Chapters 3 and 4. The Tactile Depth Generator is introduced in Chapter 5 with a focus on rendering depth using tactile feedback. Lastly, Chapter 6 gathers conclusions from the work done and presents ideas to utilise the Tactile Depth Generator.

## 2. Related work

The effectiveness of tactile feedback on touch screens has been investigated using graphical user interface elements. Hoggan et al. [2008] investigated the text entry on a touch screen device with and without tactile feedback. An ex-

periment was conducted where a phrase was shown to participants and asked to memorize it. The experiment consisted of 12 participants. After memorizing the phrase the task was to enter the text as quickly and as accurately as possible using the keyboard for each condition.

The first experiment was done using Palm Treo 750 and Samsung i718 touch screen devices utilizing the devices' internal actuators. The second study used a Dell Axim PDA with two more expensive, specialised actuators providing localization. The methodology of the second experiment was similar to the first but this time the device had two C2 actuators attached to the back of the device to see if they could increase performance and to get closer to that of real physical keyboard.

Results showed that touchscreen keyboards with tactile feedback produce fewer errors and greater speed of text entry compared to standard touchscreen keyboards without tactile feedback in laboratory and mobile contexts. Hoggan et al. [2008] also found, using multiple, specialised actuators providing localised feedback, improved text entry as opposed to a single standard actuator.

A six-dot Braille character consists of a rectangular array of two columns and three rows where individual dots are either raised or lowered which transliterates traditional written letters. These tactile characters can be read by gliding the fingers over the dots. Rantala et al. [2009] used a prototype device based on the Nokia 770 Internet Tablet to present Braille. A prototype equipped with a piezoelectric actuator solution under the touchscreen of the device was used in an experiment where the task was to recognize single Braille characters. Experiments were done using both stylus and index finger. Results showed that single Braille characters can be presented in a mobile device using spatiotemporal and temporal tactile feedback.

Haptic rendering research usually focuses on full force feedback with well established object data. However, it is not likely that all the objects require full haptic feedback. Kim and Ryu [2004] proposed a novel 2.5D haptic rendering algorithm for the background scene which doesn't provide the tactual feedback on the reverse or hidden surfaces like full 3D haptic feedback. The algorithm performs collision detection and responds using depth values in graphic rendering hardware. Kim and Ryu used a localized occupancy map instance to determine the rendering force in the collision response. These architectural solutions makes the algorithm efficient and memory effective. This is why the algorithm is suitable for rendering both static and dynamically changing background scenes.



Robles-De-La-Torre and Hayward [2001] ran two experiments where human subjects explored surfaces by touch. They used an apparatus that allowed them to separate force cues from surface geometry. Subjects used their index fingers to press down on the plate of the manipulandum while smoothly rolling it on three interchangeable physical surfaces. The interface's force was called a 'virtual surface' because of its capability to provide the same horizontal force component that an equivalent physical surface would return, regardless of the manipulandum's vertical position. The force applied by the user and the force from the physical surface were used to calculate the horizontal force for the haptic interface through a frictionless point. The experiments consisted tasks where participants were asked to locate and identify physical and virtual features of the surfaces.

### 3. Depth perception

People with normal vision have the ability to see three-dimensional space and to judge distances accurately. This ability is called *depth perception*. Without depth perception you would not be able to, for example, throw darts or navigate around room.

We use multiple *depth cues* to construct our perception of three-dimensional spaces. Depth cues are formed using environmental features and body messages. Depth perception has been found in newborns by some psychologists, while others view it as learned, but most likely it is partly learned and partly instinctive [Coon and Mitterer, 2008].

#### 3.1. Visual depth cues

According to the theory of depth perception the third dimension of space is lost in the two-dimensional retinal image. Nevertheless, the third dimension can be added to the flat visual field if depth cues are utilized. Depth cues are divided into monocular cues (linear perspective, apparent size, superposition, light and shade, relative motion, aerial perspective and accommodation) and binocular cues (binocular disparity and convergence).

Gibson [1986] suggested a new theory because he suspected that the traditional list of cues for depth was inadequate and the whole theory of depth perception to be false unless perception begins with a flat picture. Gibson doubted the theory of depth perception because studies made with pilots failed to make learning to fly any easier. Gibson's new theory is called ground theory and it is based on the idea that the world consists of a basic surface with adjoining surfaces, not of bodies in empty air.

### 3.2. Pictorial depth cues

Two-dimensional pictures such as paintings and photographs have monocular depth cues called pictorial depth cues giving information about space, depth and distance. These features combined can create powerful illusion of depth.

Linear perspective-cues are based on the apparent convergence of parallel lines in the environment. Their convergence implies great distance because you know they are parallel. Artists can make the more distant of two objects of the same size smaller by giving visible size-cues. When an object partially blocks another object superposition-cues make it easy to see which object is closer. Two-dimensional design can achieve a three-dimensional appearance by lighting it in ways that create clear patterns of light and shadow. Haze, washed out color and objects that are lacking in detail are recognised as more distant than clear object because of aerial perspective. Figure 1 represents these pictorial depth cues [Coon and Mitterer, 2008].

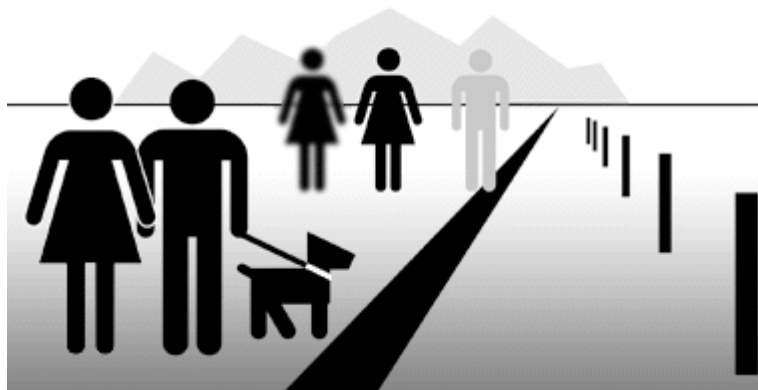


Figure 1. Pictorial depth cues presented

### 3.3. Computer vision depth perception

The production of reliable and accurate three-dimensional depth data measurements is now possible because of recent technological advances in camera optics, CCD cameras and laser rangefinders, for example. Images containing this explicit three-dimensional information have many advantages over their two-dimensional counterparts. Only limited information can be deduced about physical shape and size of an object in a scene from a two-dimensional image but, for example, measurements of the size of an object in a scene can be straightforwardly computed from its three-dimensional coordinates.

Marshal and Martin [1992] describe *depth maps* as data structures which store  $x$  and  $y$  distance information corresponding to the rows and columns of an array in a two dimensional array much like an ordinary image. The corres-

ponding depth readings (z-values) are stored in the array's elements (pixels). A depth map is similar to a grey scale image except that the intensity is replaced with depth information. It is the simplest and most convenient way of representing and storing the depth measurements taken from a scene. Figure 2 presents a standard two-dimensional depth image.

Depth information can be obtained using two cameras in stereo or with a single camera. Depth information can be combined, for example, with structured lighting and related methods involving a single camera and a fixed light source, the positions, orientations and physical properties of which are known.

Depth maps can be produced with two cameras passively by extracting certain features from two separate two-dimensional images and matching the corresponding features between images. This method can have serious problems in finding and accurately locating features in each image and active stereoscopic methods may overcome these problems. Active stereoscopic methods illuminate the scene by sweeping the strong light source across the whole scene. Both cameras can observe the light source to provide known corresponding points in each image to produce depth maps.

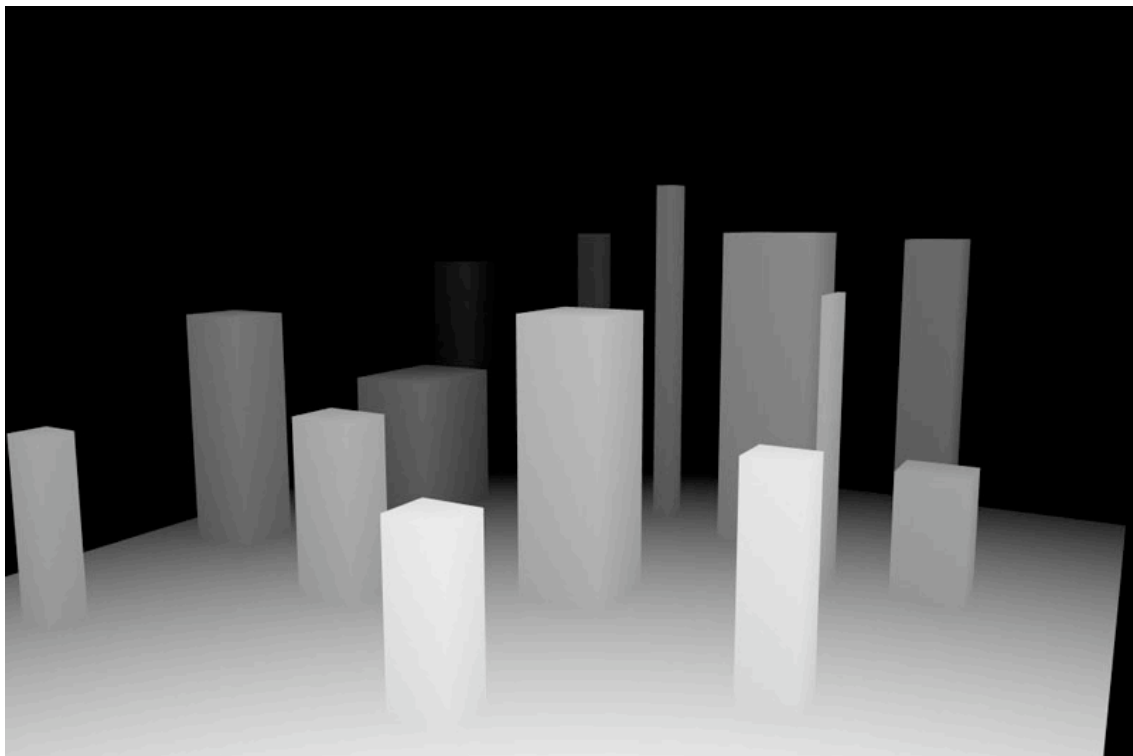


Figure 2. Depth map

#### 4. Haptic interaction

We perceive different mechanical signals such as contact forces, object movements and the geometry of objects from our everyday environment. Haptic per-

ception relies on sensory signals arising from these mechanical signals. *Haptic interfaces* produce computer-controlled mechanical signals in virtual environments which are normally experienced when haptically exploring real, everyday environments. Because haptic interfaces are programmable systems, they can create combinations of mechanical signals that do not have counterparts in real environments [Grünwald, 2008].

Psycho-physicists started using the term haptics as early as in the early 20th century when studying human touch-based perception and manipulation. Salisbury et al. [2004] state that a new usage of the word haptics began to emerge in the early 1990's when a number of emergent technologies made virtualized haptics or "computer haptics" possible. Computer haptics display simulated objects to humans in an interactive manner but unlike computer graphics, they use a haptic interface by exerting controlled forces, for example, on the human hand. Haptic interaction was demonstrated at least as early as the 1960's but haptic interaction with complex computer simulated objects was not possible until recently due to insufficient technology.

Haptic interface devices have a physical connection between the operator and device. This physical connection is used to exchange mechanical energy with a user. Salisbury et al. categorise haptic interfaces by the number of degrees of (DOF). DOF represents the number of dimensions, or in other words, the possible movements/forces exchanged between the device and operator. Opening a door using a knob can be considered a single-degree-of-freedom interaction because of the capability of measuring the position and applying forces to the operator along a single spatial dimension. Two DOF interaction like using a mouse to interact with computers has been shown to be an effective way to interact with simpler three dimensional virtual environments thus limiting the costs and complexity of the haptic devices. Force interaction with 3D virtual objects can be done with three or more degrees of freedom through devices such as the PHANTOM and 6-DOF PHANTOM [SensAble, 2010]. This interaction paradigm is based on the idea of interacting with the virtual world with a point probe.

#### **4.1. Haptic rendering**

Haptic rendering is a process in which sensory stimuli are presented to the user in order to convey information about a virtual haptic object [Salisbury et al., 2004]. The representation of the physical attributes of the virtual object is contained in this information at the simplest level of haptic rendering.

Haptic interfaces use haptic rendering algorithms to make sure that the correct interaction forces are correctly rendered on the human operator from the

virtual environment in which human operator interacts with the haptic interface. Force response algorithms compute interaction forces between the human operator avatar and virtual objects when collisions are detected in the virtual environment by collision detection algorithms. The signals produced by the force response algorithms are processed by the control algorithms. These control algorithms command the haptic device in such a way that minimizes the error between ideal and applicable force. There are no firm rules on how frequently this haptic loop needs to be repeated but 1 kHz is common in today's applications. Typical haptic rendering architecture is presented in Figure 3.

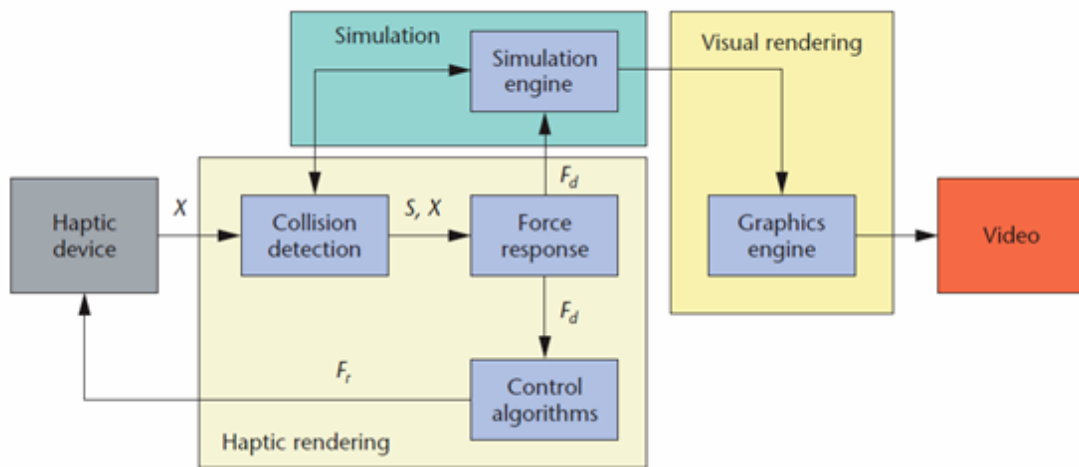


Figure 3. Typical haptic rendering architecture

#### 4.2. Tactile interactions on touch screen

Touch screens allow the user to touch, push and drag information directly with users fingers. This is why they have better user acceptance, faster input rates and are easier to use compared to keyboards. Despite these advantageous features, touch screens suffer from a lack of sufficient feedback for users. Sufficient feedback becomes even more important when the touch screen is used in a mobile setting where things like small display size, outside noise and social restrictions make audio and visual feedback significantly less effective.

Poupyrev and Maruyama [2003] presented a tactile interface for small touch screens where users could feel graphical user interface controls with their fingers. They believed that mobile devices can be used more effectively and comfortably by augmenting the GUI with tactile feedback. They based their assumption on superiority of touch as a feedback channel and the metaphor of directness and physicality in touch screen interaction.

Much like exploring normal everyday environments, interaction with touch screens is based on gestures that start when the user touches the screen and fin-

ish when the user lifts the finger up from the screen. The conditions of the gestures can be classified into touching, dragging, holding and lifting and these can be used in the development of rich and expressive tactile interfaces.

## 5. System design

The technique proposed in this paper was designed for the Immersion TouchSense Demonstrator [Immersion, 2010] and is called the *Tactile Depth Generator* (TDG). The demonstration kit utilizes the system architecture presented in Figure 3 where video and haptic interaction is handled on a 10.4-in LCD touch monitor with tactile feedback. A PC running the Windows operating system runs the simulation program while both the haptic and visual rendering is conducted using the Microsoft .NET-platform. A Windows ActiveX control utilizing Immersion's TouchSense Software Toolkit is used to send tactile effects to haptic device.

The TDG provides depth values for the collision detection algorithms. The TDG visually renders a static scene and haptically renders the depth maps. The collision algorithm then sends depth values for the simulation which then calculates the force response to present changes in depth in the scene. Lastly, the force response algorithm sends the calculated forces to the control algorithms in order to present depth information to the user.

The haptic rendering loop is presented in Figure 4. Loop starts when the user touches the screen and the system stores the scene's depth value for the position the user touched. When the user starts to drag their finger on the display the system observes changes in depth. Events for ascending or descending edges occur when the threshold value is exceeded. The TDG system plays a tactile effect for each event and stores the current depth value. The haptic rendering loop ends when the user lifts their finger up.

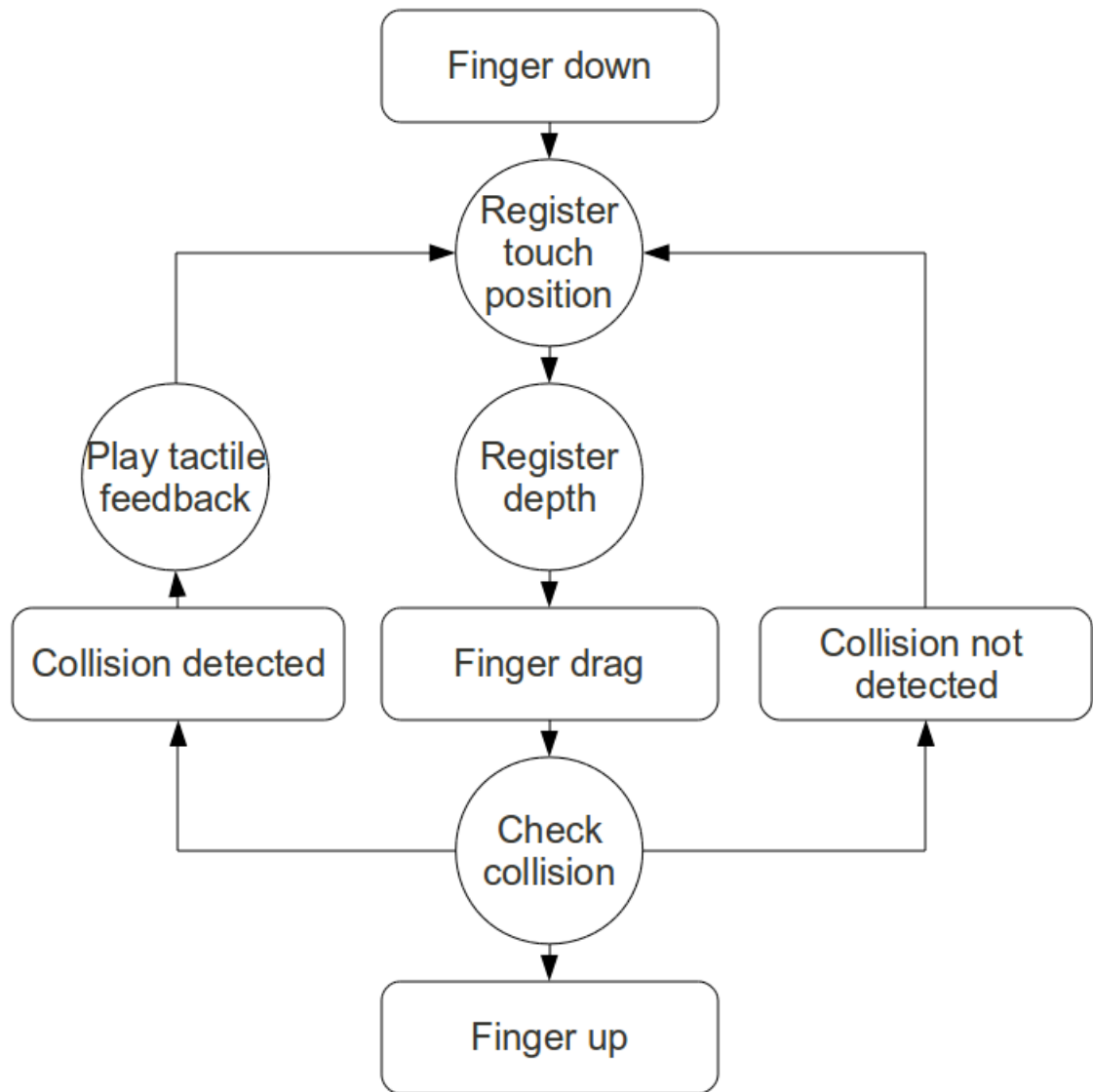


Figure 4. TDG haptic rendering loop

### 5.1. Feedback design

Robles-De-La-Torre and Hayward [2001] based their force-feedback presentation on the forces applied from the user's index finger and the surface's physical features. When transforming this force-feedback to tactile feedback on a touch screen where applied force cannot be determined, some attributes must be hard coded. The TDG makes use of three-dimensional scenes and this means the laws of physics such as gravity can be applied. In the real world, particles with mass get gravitational potential energy when they are lifted from the ground and, in turn, they lose this energy when falling. The amount of energy can be calculated by multiplying the mass of the particle, the gravitational constant and the height from the ground. Because any forces based on real world physics are not presented, the feedback can be calculated using a function with the

same dimension. This can be used to present the depth to the user by mapping it for example to duration, amplitude or frequency of the tactile feedback depending on platform capabilities.

Robles-De-La-Torre and Hayward [2001] used Gaussian profiles to present surface geometry with holes and bumps. These changes in altitude can be mapped to tactual attributes like frequency or amplitude. However, some platforms have limitations over the control of vibration attributes. The TDG addresses these issues by manipulating the parameters of vibration. For example, an ascending edge could be presented using a short pulse first with low frequency and then a short pulse with a higher frequency, and vice versa for descending edges.

## 6. Conclusions and future work

This paper presents the Tactile Depth Generator technique for presenting depth information using tactile feedback. Most of the work done on haptic rendering uses specialized devices and this limits the amount and quality of rendering that can be done. That is why the author proposes that the techniques presented in this paper could be useful for many mobile applications on many mobile devices.

The techniques presented in this paper are applicable for most of the current mobile devices using touchscreens. Real time applications such as reverse camera systems where accurate depth information is needed would profit most from the Tactile Depth Generator technique as touchscreens are becoming more common in automobiles. Furthermore, augmented reality applications could be made more realistic by offering depth information using the Tactile Depth Generator.

## Acknowledgements

This paper was supported by Immersion Corp, California. Many thanks to Eve Hoggan for advice and help with this thesis. I also want to thank Erkki Mäkinen, Jukka Raisamo and Roope Raisamo.

## References

- [Coon and Mitterer, 2008] Dennis Coon and John O. Mitterer, *Introduction to Psychology: Gateways to Mind and Behavior*. Cengage Learning, 2008.
- [Gibson, 1986] James Jerome Gibson, *The Ecological Approach to Visual Perception*. Routledge, 1986.
- [Grünwald, 2008] Martin Grünwald, *Human Haptic Perception: Basics and Applications*. Springer, 2008.



- [Hoggan et al., 2008] Eve Hoggan, Stephen A. Brewster and Jody Johnston, Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens. In: *Proc. of ACM CHI 2008*. ACM Press, 1573-1582.
- [Immersion, 2010] Immersion Corporation, <http://www.immersion.com>
- [Kim and Ryu, 2004] Jong-Phil Kim and Jeha Ryu, Hardware Based 2.5D Haptic Rendering Algorithm using Localized Occupancy Map Instance. In: *Proc. 14th Int. Conf. Artificial Reality and Telexistence*, 132-137.
- [Marshall and Martin, 1992] A. Dave Marshall and Ralph R. Martin, *Computer Vision, Models, and Inspection*. World Scientific, 1992.
- [Poupyrev and Maruyama, 2003] Ivan Poupyrev and Shigeaki Maruyama, Tactile interfaces for small touch screens. In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, ACM, 217-220.
- [Rantala et al., 2009] Jussi Rantala, Roope Raisamo, Jani Lylykangas, Veikko Surakka, Jukka Raisamo, Katri Salminen, Toni Pakkanen and Arto Hippula, Methods for presenting braille characters on a mobile device with a touchscreen and tactile feedback. *IEEE Transactions on Haptics* **2**, 1 (January-March, 2009), 28-39.
- [Robles-De-La-Torre and Hayward, 2001] Gabriel Robles-De-La-Torre and Vincent Hayward, Force can overcome object geometry in the perception of shape through active touch. *Nature* **412** (July 2001), 445-448.
- [Salisbury et al., 2004] Kenneth Salisbury, Federico Barbagli and Francois Conti, Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE* **24**, 2 (March-April, 2004), 24 - 32.
- [SensAble, 2010] SensAble Technologies, <http://www.sensable.com>

# Lifeloggaus – tausta, hyödyt ja ongelmat

Saila Oldén

## Tiivistelmä.

Lifeloggaus, ”elämän taltiointi”, on toimintaa, jossa ihminen tallentaa digitaalisesti kaiken omasta elämästään käyttäen erilaisia välineitä, kuten esimerkiksi videokameraa ja tietokonetta. Tässä tutkielmassa esitellään lifeloggauksen taustaa ja ensimmäisiä lifeloggauksia, sekä lifeloggauksen hyötyjä ja haittoja.

**Avainsanat ja -sanonnat:** Lifeloggaus, SenseCam, EyeTap.

**CR-luokat:** H.1.2.

## 1. Johdanto

Lifeloggaus, ”elämän taltiointi”, on toimintaa, jossa ihminen tallentaa digitaalisesti jokaisen sekunnin omasta elämästään käyttäen erilaisia välineitä, kuten videokameraa, tavallista kameraa, tietokonetta tai matkapuhelinta. Tallennettuja tietoja voivat olla esimerkiksi tietokoneella suoritettut toiminnot (verkkosivut, joita on selailtu, lähetetyt sähköpostiviestit, digitaaliset valokuvat, tekstitiedostot), matkapuhelinaktiviteetit (tekstiviestit, soitetut puhelut, kännykkäkameralla otetut valokuvat), biologiset toiminnot (verenpaine, sykearvot) sekä erilaiset valokuvat ja videot käyttäjän aktiviteeteista (Byrne & Jones, 2008).

Aivan ensimmäiseksi lienee syytä mainita muutama sana käyttämästäni termistöstä. Tein tietoisin valinnan olla käyttämättä englanninkielisen termin ”lifelogging” suomen kielessä olevaa vastinetta ”elämän taltiointi”, vaan käytän sen sijaan termiä ”lifeloggaus”. Yksi syy tähän oli se, että mielestäni termi ”elämän taltiointi” on hyvin hankala käyttää tekstissä luontevasti, eikä se kuvaa toimintaa aivan yhtä hyvin kuin alkuperäinen, englanninkielinen termi. Toisaalta taas tekstissä käytettynä ”lifelogging” ei myöskään ole luonteva. Blogin kirjoittamisesta käytetään yleisesti termiä ”bloggaus”, joten sen kautta päädyin termiin ”lifeloggaus”. Toinen syy valintaan oli se, että tietokonealan termistö on joka tapauksessa hyvin pitkälti englannin kielestä johdettua, joten se mielestäni oikeuttaa tällaisen ikään kuin sekakielisen termin käyttämisen.

Tämä tutkielma on kirjallisuuskatsaus lifeloggauksesta. Tutkielman tarjoamissa puitteissa esimerkiksi lifeloggauksen haastattelu ei ollut mahdollista, eikä itselläni ole omakohtaista kokemusta tällaisesta elämän taltioinnista, mitä olisin voinut tässä tutkielmassa reflektoida.

Tutkielman rakenne on seuraavanlainen. Ensin esittelen lifeloggauksen taustaa ja ensimmäisiä lifeloggauksia. Mukana on LifeLog-projekti, Steve Mann ja hänen EyeTap-laitteensa, sekä MyLifeBits-projekti. Kolmannessa luvussa käyn läpi lifeloggauksen hyötyjä ensin hieman yleisemmällä tasolla, ja sen jälkeen esittelen konkreettisen esimerkin eli SenseCamin käytön Alzheimer-potilaiden apuvälineenä. Neljännessä luvussa kuvataan lifeloggauksen mahdollisia haittoja.

## 2. Lifeloggauksen taustaa ja ensimmäisiä lifeloggaajia

Tässä luvussa esitellään ensin Vannevar Bushin Memex-laite, jota voidaan pitää lifeloggaus-tekniikan varhaisena esikuvana. Tämän jälkeen käydään läpi muutama ensimmäinen lifeloggaus-projekti. Ensimmäisenä esittelyssä on LifeLog, DARPA:n (The Defense Advanced Research Projects Agency) projekti, joka alkoi vuonna 2003. Tämän jälkeen kerrotaan hieman Steve Mannista ja hänen kehittämisestään EyeTap-laitteesta. Viimeiseksi esittelyssä on MyLifeBits-projekti.

### 2.1 Vannevar Bush

Vuonna 1945 Vannevar Bush, yhdysvaltalainen insinööri ja tiedevaikuttaja, esitteli Memexin artikkelissaan *As We May Think*. Memex oli laite, johon käyttäjä pystyi tallentamaan kaikki kirjansa, tietonsa ja yhteydenpitonsa. Laite oli mekaaninen, joten sieltä pystyi hakemaan tietoa nopeasti ja joustavasti. Bush kuvaili laitetta ihmisen muistin ”intimiksi lisäksi” (Bush, 1945).

Laite koostui pöydästä, jossa oli kaltevat, läpinäkyvät näytöt, joihin luettava materiaali heijastettiin (Bush, 1945). Laitteessa oli myös näppäimistö, sekä erilaisia painikkeita ja vipuja. Muuten laite näytti tavalliselta pöydältä. Suurin osa Memexin sisällöstä (kirjat, kuvat, sanomalehdet) oli tarkoitettu ostaa käyttövalmiina mikrofilmeinä. Laitteen päällä oli läpinäkyvä levy, johon voi asettaa esimerkiksi kuvan tai muistiinpanoja, ja ottaa niistä valokuvan, joka tallentui Memexiin. Laitteesta pystyi myös hakemaan tietoa sinne tallennetusta materiaalista, esimerkiksi tarkastelemaan jotakin tiettyä kirjaa. Kirjoihin pystyi lisäksi tekemään muistiinpanoja ja huomautuksia. Käyttäjä pystyi myös muodostamaan niin sanottuja polkuja (*trails*), jotka yhdistivät eri lähteitä toisiinsa, ja joita seuraamalla asioista sai enemmän tietoa.

Visio Memexistä syntyi sen pohjalta, että Bushin mielestä tieteen tekemisen ja arvioinnin metodit olivat vanhentuneita ja toimimattomia (Bush, 1945). Bushin mukaan ihmisen mieli toimi assosiaation kautta, ja tällä periaatteella toimisi myös tehokas tiedonsäilyttäjä, jollaiseksi hän Memexin artikkelissaan kuvaili. Bush suunnitteli myös Memexin käyttäjän biologisten toimintojen tallentamista: esimerkiksi silmän sähköiset impulssit voitaisiin tallentaa, jolloin saataisiin kuva siitä, mitä käyttäjä milloinkin näki (Byrne & Jones, 2008).

Bushin Memexiä voidaan pitää ensimmäisenä lifeloggaus-tekniikan visiona. Nykyajan versio Memexistä on tosin hyvin erilainen. Bushin visiossa yhdellä laitteella pystyi tallentamaan monenlaista tietoa, mutta nykyajan lifeloggaajat joutuvat käyttämään monia erilaisia laitteita elämänsä tallennukseen, esimerkiksi tietokoneita ja matkapuhelimia (Byrne & Jones, 2008).

## 2.2 Ensimmäiset lifeloggaus-projektit

Seuraavaksi esitellään kolme lifeloggaus-projektia. Ensin esittelyssä on LifeLog-projekti, sen jälkeen Steve Mann ja hänen EyeTap-laitteensa, ja viimeiseksi esitellään MyLifeBits-projekti.

### 2.2.1 LifeLog

The Defense Advanced Research Projects Agency (DARPA) on Yhdysvaltojen asevoimien keskeinen kehitys- ja tutkimusorganisaatio. Vuonna 2003 DARPA aloitti projektin nimeltä LifeLog, jonka tarkoituksena oli kerätä yhteen paikkaan tieto kaikesta siitä, mitä ihmisilö sanoo, näkee tai tekee elämänsä aikana, eli toisin sanoen kaikki puhelut jotka hän soittaa, TV-ohjelmat joita hän katsoo, lehdet joita hän lukee, lentoliput jotka hän ostaa, sähköpostit jotka hän lähettää ja vastaanottaa ja niin edelleen (Shachtman, 2004). Tästä valtavasta tietomäärästä tutkijoiden oli määrä koota yhteen ihmissuhteita, muistoja, tapahtumia ja kokemuksia.

LifeLogiin oli tarkoitus kerätä kolmenlaista dataa: fyysistä dataa, transaktionaalista dataa ja konteksti- tai mediadataa (IPTO, 2003). Visuaaliset, auditiiviset ja mahdollisesti jopa haptiset sensorit tallentaisivat kaiken mitä ihminen näkee, kuulee ja tuntee: GPS-paikannin, digitaalinen kompassi ja inertiasensorit tallentaisivat ihmisen suunnan ja liikkeen, ja biolääketieteelliset sensorit tallentaisivat ihmisen fyysisen tilan. LifeLog tallentaisi myös ihmisen tietokoneella tekemät asiat, kuten sähköpostin ja kalenterin käytön sekä internetissä surffailun. Myös puhelutiedot pystyttäisiin tallentamaan, ja fakit ja muu materiaali voitaisiin skannata ja tallentaa.

LifeLogia oli tarkoitus käyttää joko itsenäisenä järjestelmänä (*stand-alone system*), joka toimisi tehokkaana automatisoituna multimediatäydä- tai leikekirjana, ja josta voisi hakea hyvinkin tarkkaa tietoa (kuvia, ääntä, videota) mistä tahansa menneestä tapahtumasta, tai vaihtoehtoisesti osana muita järjestelmiä (*as a subsystem*) esimerkiksi lääketieteen tai talouden alalla (IPTO, 2003). Tarkoituksena oli, että kun yhä useammat ihmiset perustavat omat LifeLoginsa, niistä kerättyä tietoa voidaan käyttää muun muassa sairaus epidemioiden varhaiseen havaitsemiseen ja opettamaan roboteille ihmismäisempää elämää.

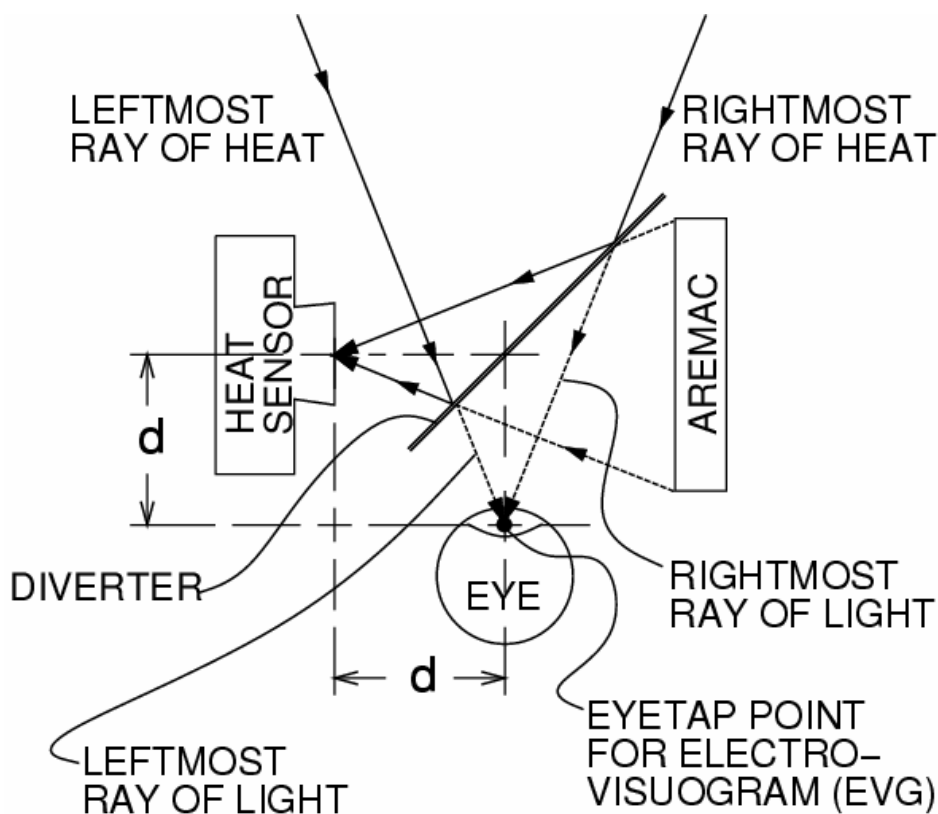
LifeLog-projektin kannattajien mielestä järjestelmästä olisi voinut kehittyä lähes täydellinen digitaalinen muisti, joka auttaisi käyttäjänsä muistamaan menneet asiat (Shachtman, 2004). Projektilla oli kuitenkin myös vastustajia, jotka olivat huolissaan järjestelmän vaikutuksista yksityisyyteen: heidän mukaansa LifeLogista voisi tulla työkalu, joilla profiloidaan mahdollisia valtion vihollisia. Vuonna 2004 projekti lopetettiin kaikessa hiljaisuudessa. Lopettamisen syyksi DARPA ilmoitti ainoastaan ”prioriteettien muutoksen” (Shachtman, 2004).

### 2.2.2 Steve Mann ja EyeTap

Steve Mannia on kutsuttu ”maailman ensimmäiseksi kyborgiksi”, koska hän käyttää älyvaatteita (*wearable computing*) (O'Hara, Tuffield & Shadbolt, 2009). Lapsuudestaan asti

Mann on suunnitellut, rakentanut ja käyttänyt tietokonejärjestelmiä kehitelläkseen sähköisen näön (*electronically mediated vision*) (Mann, 2004). EyeTap syntyi näistä kokeiluista. Laitetta on kehitelty viimeiset 25 vuotta.

EyeTap on kokemukset tallentava järjestelmä, joka saa silmän toimimaan ikään kuin kamerana ja näyttönä samaan aikaan (Mann, 2004). EyeTap tallentaa ja analysoi silmän läpäiseviä valonsäteitä, muokkaa niitä ja yhdistelee uudelleen (kuva 1). Laitte mahdollistaa "välitetyin todellisuuden" ("*mediated reality*") luomisen, koska se muuntelee silmän läpäisevää valoa. Näin voidaan muunnella silmän näkemää visuaalista sisältöä, kuten esimerkiksi jättää mainokset huomiotta, tai yksinkertaisemmin sanottuna nähdä paremmin erilaisilla tavoilla.



Kuva 1. EyeTap tallentaa ja analysoi silmän läpäiseviä valonsäteitä, muokkaa niitä ja yhdistelee uudelleen (Mann, 2004).

Perinteinen linssitekniologia mahdollistaa monien optisten häiriöiden korjaamisen ja parantaa joitakin visuaalisia kykyjämme (Mann, 2004). Esimerkiksi aurinko- ja hitsauslasien avulla pystymme katsomaan hyvin kirkkaita asioita. Suurennuslaseilla voimme katsella hyvin pieniä kohteita. On myös keksitty laitteita, joiden avulla voi nähdä pimeässä. EyeTap kokoaa kaikki nämä ominaisuudet yhteen laitteeseen, ja se toimii tilanteissa joissa nämä ominaisuudet yksinään eivät riittäisi. EyeTap voi auttaa myös niitä, jotka kärsivät vaikeammista visuaalisista puutteista.

EyeTapin tärkeimmälle ominaisuudelle ei kuitenkaan löydy vastaavuutta perinteisistä näön apuvälineistä (Mann, 2004). EyeTap voi nimittäin auttaa meitä muistamaan parem-

min tallentamalla kaiken, mitä elämämme aikana näemme. Mann kutsuu tätä ominaisuutta lifeglogiksi (*lifelong cyborglog*) tai lyhyemmin glogiksi. Tällainen visuaalinen teko-muisti (*visual memory prosthetic*) on todella hyödyllinen kaikille, sillä ympäristömme tulvii informaatiota. EyeTap auttaa meitä palauttamaan mieleen yksityiskohtia, joita emme muuten muistaisi. Lifeglog lisää myös turvallisuutta ja vähentää rikollisuutta tallentamalla todisteita rikollisesta toiminnasta. EyeTapiia voidaan käyttää myös suodattamaan visuaalista informaatiota, jotta keskeinen tieto erottuu taustasta. Näin voidaan luoda maailma ilman mainoksia.

### 2.2.3 MyLifeBits

Yksi parhaiten tunnetuista lifelog-projekteista on MyLifeBits, joka käynnistyi vuonna 2001. Projekti on jatkoa Gordon Bellin CyberAll-projektille, jonka tarkoituksena oli koodata ja tallentaa kaikki ihmisen yksityinen ja ammatillinen tieto yhteen arkistoon, sekä tarjota helppo tapa päästä käsiksi näihin tietoihin (Bell, 2001). Arkistoon tallennettiin kirjoja, CD-levyjä, yhteydenpitoa (kirjeitä, muistioita ja sähköposteja), papereita, valokuvia ja albumeita, videoita, sekä muita tapahtumia.

CyberAll-projektissa ei ollut käytössä tietokantaa, vaan tallennettavat tiedostot nimettiin tarkkaan ja kansioita ja oikopolkuja käytettiin järkevästi (Gemmell et al., 2006). Tietokokoelman kasvaessa kansioiden käyttö kuitenkin muuttui ylivoimaiseksi, varsinkin kun etsintätyökalut olivat tuolloin (vuonna 2000) hyvin hankalia käyttää. Nykyiset etsintätyökalut ovat valtavasti parempia, mutta nekin pelaavat silti vielä tiedostoilla ja kansioilla. Projektiin osallistujat halusivat tehokkaamman työkalun, jolla pystyisi muun muassa järjestelemään ja luokittelemaan tietoa eri tavoilla. Näiden haasteiden takia fokus siirtyi tiedon tallentamisesta sellaisen ohjelman kehittämiseen, joka tekisi tallennetusta materiaalista helposti käsiteltävää ja hyödyllistä. Niin sai alkunsa MyLifeBits, jonka tavoitteena oli parantaa huomattavasti tallennetun materiaalin järjestämistä, etsimistä, kommentoimista ja hyödyntämistä. Tarkoituksena oli myös luoda yksi yhtenäinen tietokanta vanhojen "datasarekkeiden" sijaan.

MyLifeBits-projektissa on neljä suunnitteluperiaatetta: 1) tiedon järjestämiseen tulisi voida käyttää kokoelmia ja hakua hierarkioiden sijaan, 2) erilaiset visualisoinnit tulisi mahdollistaa, 3) kommentointi on ehdotonta ei-tekstuaaliselle medialle ja se tulee tehdä helpoksi, ja 4) järjestelmän tulee tukea kaksisuuntaista linkitystä (Gemmel et al., 2002). MyLifeBits on siis tietokanta, joka koostuu lähteistä (mediasta) ja linkeistä. Linkki tarkoittaa sitä, että yksi lähde selittää toista lähdettä. Yksi lähde voi selittää kuinka montaa muuta lähdettä tahansa, ja yhtä lähdettä voi selittää kuinka monta muuta lähdettä tahansa. MyLifeBits-toimii siis samaan tapaan kuin Bushin Memex, joka toimikin projektin innoittajana ja esikuvana: linkityksillä ja kommentteilla on keskeinen rooli tietojen järjestämisessä.

MyLifeBits-järjestelmä tallentaa jokaisen mahdollisen tiedon Gordon Bellin, projektin keskiössä olevan miehen, elämästä: artikkelit, kirjat, kortit, CD-levyt, kirjeet, muistiot, musiikin, valokuvat, maalaukset, esitykset, kotivideot ja kalenterimerkinnot (Nack, 2005). Järjestelmä tallentaa myös kaikki vierailut www-sivut, kaikki Instant Message -viestit,

kaikki puhelut, tapaamiset, radio- ja televisio-ohjelmat, kaikki tietokoneen hiiren liikkeet ja näppäimistön painallukset (Gemmell et al., 2006). Projektin päämääränä on kuvata vuorokauden ympäri, viikon jokaisena päivänä kaikki, mitä Bell tekee: kaikki äänet, kuvat, videot, terveydentilan muutokset ja niin edelleen (Nack, 2005).

Tässä luvussa on lyhyesti kuvattu lifeloggauksen taustaa käsittelemällä Vannevar Bushin visiota Memexistä. Tämän jälkeen esiteltiin kolme lifelog-projektia, Yhdysvaltojen asevoimien kehitys- ja tutkimusorganisaation LifeLog-projekti, Steve Mann ja hänen EyeTap-laitteensa, sekä MyLifeBits-projekti. Seuraavassa luvussa käydään läpi lifeloggauksen hyötyjä.

### 3. Lifeloggauksen hyödyt

Tässä luvussa käsitellään lifeloggauksen hyötyjä. Ensin esitellään lifeloggauksen potentiaalisia hyötyjä yleisellä tasolla, ja sen jälkeen kerrotaan konkreettisesta hyödyistä, eli siitä, miten lifeloggausta voisi käyttää Alzheimeria sairastavien ihmisten ja heidän hoitajiensa apuna.

#### 3.1 Lifeloggauksen potentiaaliset hyödyt

Sellen ja Whittaker (2010) nimeävät viisi tapaa, miten lifelog-tietoja voitaisiin käyttää ihmisten apuna. Mieleen palauttaminen (*recollecting*) tarkoittaa sitä, että elämme uudelleen joitakin elämämme vaiheita, esimerkiksi muistelemme, mitä jossakin kokouksessa puhuttiin tai mikä olikaan tapaamamme ihmisen nimi. Muistelu (*reminiscing*) on tavallaan mieleen palauttamista, mutta sillä on emotionaalinen tai sentimentaalinen sävy. Esimerkiksi kotivideoiden katselu tai valokuva-albumien selailu perheen tai ystävien kanssa on muistelua. Lifelogit myös auttavat palauttamaan muistiin (*retrieve*) tiettyä digitaalista informaatiota, jota olemme keränneet vuosien myötä, esimerkiksi dokumentteja, sähköposteja ja www-sivuja. Reflektointi (*reflecting*) voi olla esimerkiksi menneisyyden käyttäytymiskaavojen analysointia tai jo tapahtuneiden asioiden katsomista eri perspektiivistä, eli toisin sanoen asioiden näkemistä eri valossa. Asioiden muistaminen (*remembering intentions*) tarkoittaa jokapäiväisten asioiden muistamista, esimerkiksi lääkkeiden ottamista ja tapaamisiin saapumista, ei menneisyydessä tapahtuneiden asioiden muistamista kuten mieleen palauttaminen.

O'Hara ja muut (2009) puolestaan esittävät, että lifelog-tietoja pystyttäisiin käyttämään neljään eri tarkoitukseen. Ensinnäkin tietoja olisi mahdollista hyödyntää antamalla käyttäjälle neuvoja ja suosituksia hänen aikaisemman käytöksensä pohjalta. Hänelle voitaisiin esimerkiksi suositella tietynlaista musiikkia sen perusteella, mitä musiikkia hän on aikaisemmin verkosta ladannut. Jotkut palveluthan, kuten Amazon tai last.fm, käyttävät jo tämäntyyppistä suosittelua, mutta ne eivät pysty ottamaan huomioon sitä, että käyttäjän mieltymykset yleensä vaihtelevat esimerkiksi paikan ja olosuhteiden mukaan (esimerkiksi kotona halutaan kuunnella rauhallisempaa musiikkia, autossa äänekkäämpää). Lifelog-tietoja käyttämällä tämä pystyttäisiin ottamaan huomioon, koska ne ovat kokonaisval-

taisempia ja niitä kerätään koko ajan ja useissa konteksteissa. Tällainen ominaisuus saattaisi kiehtoa monia käyttäjiä ja tuoda esimerkiksi Amazonille paljon uusia asiakkaita. Käytännössä tämä "laajennettu suosittelu" tarkoittaisi sitä, että jos vieraillee Amazonin verkkosivuilla sen jälkeen, kun on ladannut Facebookiin kuvia koiranpennustaan, kauppa osaisi ehdottaa ostoslistalle vaikkapa pennunkasvatus- tai ruokintaopasta.

Toisekseen lifelog-tietoja voitaisiin käyttää metadatan antamiseen. Esimerkkinä tästä on Photocopin-järjestelmä, joka auttaa käyttäjää digitaalisten valokuvien arkistoinnissa ja kommentoinnissa yhdistämällä tietoa kuvanottokontekstista ja muista lähteistä. Esimerkiksi lomakuvien lajittelu ja verkkoon laitto olisi paljon mielekkäämpää ja helpompaa jos siihen saisi apua ohjelmalta, joka pystyy automaattisesti antamaan valokuvista tiettyjä perustietoja, kuten esimerkiksi sen, minkälaisessa ympäristössä se on otettu tai minkälaisia hahmoja kuvassa on.

Kolmanneksi lifelog-tietojen pohjalta käyttäjä voisi luoda eräänlaisen avatar-hahmon, joka olisi tietoinen itsestään, ja joka toimisi tietynlaisena välittäjänä käyttäjän ja muiden online-toimijoiden välillä. Avatar tekisi päätöksiä esimerkiksi informaation jakamisen ja yksityisyyden suhteen käyttäjän puolesta, ja sen avulla käyttäjä voisi osittain automatisoida yhteydenpitonsa esimerkiksi eri yritysten kanssa.

Neljänneksi lifelog-tietoja voitaisiin käyttää sairaushistorian luomiseen ja terveystietojen tallentamiseen. Näiden tietojen pohjalta pystyttäisiin havaitsemaan muutokset esimerkiksi käyttäjän terveydentilassa.

Allenin (2007) mukaan emme muista kaikkia kokemuksiamme, ja ne jotka muistamme, saatamme muistaa väärin. Muisti myös heikkenee iän myötä. Jopa ikimuistettavat asiat saattavat unohtua: Alzheimerin tautia sairastanut Ronald Reagan unohti olleensa Yhdysvaltain presidentti. Entisaikoina ihmiset käyttivät tarinankerrontaa ja kuvia muistinsa tukena, mutta nykyään tähän tarkoitukseen voidaan käyttää tietokonetta. Allen ennustaakin, että lifeloggaus voisi joskus syrjäyttää tai ainakin täydentää tällä hetkellä olemassa olevia muistojen tallentamisen käytäntöjä. Lifelog voisi toimia päiväkirjan lailla, ja säilyttää yksilölle tärkeät faktat ja käsitykset. Lifeloggaus voisi myös parantaa yksilön elämän laatua, sillä se saattaisi rohkaista ihmisiä itsetutkiskeluun ja -tuntemukseen. Oman lifelogin jakaminen toisten kanssa voisi lisätä läheisyyttä, ymmärrystä ja vastuuta ihmisuhteissa. Kuolleen vanhemman, puolison tai lapsen lifelogin periminen voisi auttaa säilyttämään perheen historiaa ja helpottaa menetyksen aiheuttamaa tuskaa. Lifelogit voisivat myös lisätä turvallisuutta: ryöstäjä tai raiskaaja voisi miettiä aikeitaan kahdesti jos tietäisi päätyvänsä uhrinsa lifelogiin.

Lifeloggaus voisi siis olla monella tavalla hyödyllistä. Hyötyjä on kuitenkin vaikea arvioida tarkasti ja luotettavasti sillä kovinkaan moni ihminen maailmassa ei vielä käytä kaiken tallentavia lifeloggaus-järjestelmiä. Seuraavassa luvussa kerrotaan kuitenkin tutkimuksesta, jossa saatiin konkreettisia tuloksia lifeloggaus-järjestelmien käytöstä Alzheimer-potilaiden apuvälineenä.



### 3.2 Alzheimerin tauti ja SenseCam

Lee ja Dey (2008) tutkivat sitä, miten lifeloggaus-tekniikkaa voisi käyttää Alzheimerin tautia sairastavien ihmisten apuna. Alzheimerin taudissa pääoire on episodisen muistin häiriö (*episodic memory impairment*) eli EMI. Episodinen lähimuisti (*recent episodic memory*) tukee tietämystä itsestämme (*sense of self*) mahdollistamalla sen, että voimme ajatussissamme palata ajassa taaksepäin ja elää uudelleen kaikki miellyttävät kokemuksemme. EMI:stä kärsiville ihmisille tämä ei ole mahdollista, ja he joutuvat koko ajan muuttamaan käsityksiään siitä, mitä he voivat ja eivät voi tehdä. EMI voi johtaa yksilön elämän autonomian ja kontrollin menetykseen, ja aiheuttaa epävarmuutta, ärsytystä ja turhautumista. EMI:stä kärsivä voi myös vetäytyä sosiaalisesta vuorovaikutuksesta ja jopa sairastua masennukseen, koska ei enää suoriudu asioista niin kuin ennen. Hän joutuu turvautumaan toisten ihmisten apuun ja kysymään heiltä toistuvasti neuvoja koskien menneitä ja nykyisiä asioita. EMI:stä kärsivän ihmisen läheinen voi myös sairastua masennukseen, sillä hän joutuu huolehtimaan rakkaansa kognitiivisista ja fyysisistä tarpeista.

Lee ja Dey suunnittelivat järjestelmän, johon kuului Microsoftin SenseCam-laite, ääninauhuri ja GPS-paikannin. SenseCam on kaulassa pidettävä kamera, joka ottaa valokuvia automaattisesti, tässä tutkimuksessa 30 sekunnin välein. Valokuvien lisäksi tallennettiin kaikki tutkimuksessa mukana olleiden kuulemat äänet, ja heidän sijaintinsa mitattiin GPS-paikantimella 1–5 sekunnin välein. Kaikki tutkimukseen osallistuneet henkilöt käyttivät näitä kolmea laitetta koko tutkimuksen ajan, joten dataa kertyi suuri määrä. EMI:stä kärsivä ihminen ei kuitenkaan pysty käsittelemään näin suurta määrää tietoa, joten tietoa tulee suodattaa ja näin tarjota ihmiselle parhaat muistivihjeet.

Leen ja Deyn järjestelmään kuului tietokonesovellus nimeltä CueChooser, joka automaattisen sisältöanalyysin avulla erotteli tietomäärästä ne valokuvat ja äänet, jotka voisivat toimia hyvinä muistivihjeinä. Lopullisen valinnan vihjeistä teki kuitenkin tutkimushenkilön hoitaja. Vihjeistä koottiin eräänlainen kämmentietokoneella esitettävä diashow, jossa näytettiin siis SenseCamin ottamia valokuvia. Lisäksi valokuviin voitiin liittää muita vihjeitä, kuten esimerkiksi ääntä. Tätä diashow'ta tutkimushenkilöt pystyivät katselemaan myös itsekseen, hoitajan ei tarvinnut olla läsnä.

Tutkimukseen osallistui kolme potilas-hoitaja -paria, heistä jokainen oli aviopari, joissa mies oli potilas ja vaimo hoitaja. Otos oli siis hyvin pieni, ja sen pohjalta tulee tehdä ainoastaan hyvin varovaisia johtopäätöksiä järjestelmän toiminnasta. Tutkimuksessa oli kolme hypoteesia. Ensimmäinen oli, että järjestelmää käyttämällä tutkimushenkilöt muistavat enemmän yksityiskohtia kokemuksistaan. Toinen hypoteesi oli, että järjestelmää käyttämällä tutkimushenkilöt luottavat enemmän muistiinsa. Kolmas hypoteesi oli, että järjestelmän käyttö vähentää tutkimushenkilöiden hoitajien taakkaa. Tutkimuksessa käytettiin kolmea erilaista järjestelmää: Leen ja Deyn suunnittelemaa järjestelmää, jossa käytettiin CueChooser-ohjelmaa; järjestelmää, jossa tutkimushenkilön hoitaja valitsi diashow'hun tulevat kuvat ilman CueChooserin apua, ja järjestelmää, jossa tutkimushenkilö ja hoitaja kävivät läpi kaikki SenseCamilla otetut kuvat. Näiden järjestelmien käytön lisäksi jokainen tutkimukseen osallistunut pari osallistui kontrolliryhmään.

Tutkimuksessa selvisi, että kun tutkimushenkilöt käyttivät Leen ja Deyn järjestelmää, he muistivat menneistä tapahtumista enemmän yksityiskohtia kuin käyttäessään muita järjestelmiä, ja he myös luottivat muistiinsa enemmän. Tutkimuksen mukaan Leen ja Deyn järjestelmän käyttö myös vaati hoitajilta vähiten aikaa, eli vähensi heidän työtaakkaansa. Tämän tyyppinen lifelog-datan ja -tekniikan käyttö siis näyttäisi helpottavan Alzheimerin tautia sairastavien henkilöiden ja heidän hoitajiensa elämää.

Leen ja Deyn tutkimuksessa saatiin rohkaisevia tuloksia. Täytyy kuitenkin muistaa, että heidän tutkimukseensa osallistui vain kolme Alzheimeria sairastavaa henkilöä ja heidän hoitajansa. Otokoko on siis aivan liian pieni jotta sen pohjalta voitaisiin tehdä luotettavia johtopäätöksiä. Tutkimustulokset onkin syytä ottaa lähinnä rohkaisevana esimerkkinä lifeloggaus-tekniikan käytännön hyödyistä.

Tässä luvussa esiteltiin lifeloggauksen hyötyjä ensin hieman yleisemmällä tasolla, ja sitten konkreettisen esimerkin avulla. Seuraavassa luvussa käydään läpi lifeloggaukseen liittyviä ongelmia.

## 4. Lifeloggaukseen liittyviä ongelmia

Tässä luvussa esitellään lifeloggaukseen kohdistuvaa kritiikkiä ja niitä ongelmia, joita lifeloggaukseen katsotaan liittyvän.

### 4.1 Lifeloggauksen hyödyllisyys

Kuten kolmannessa luvussa esitettiin, lifeloggauksessa on paljon potentiaalista hyvää, mutta onko se pohjimmiltaan hyödyllistä? Kaiken tallentavia lifelog-järjestelmiä käyttää vain muutama ihminen maailmassa (Sellen & Whittaker, 2010), joten niiden hyödyllisyyttä ja vaikuttavuutta on hyvin hankala tutkia. Jonkun verran on kuitenkin tutkittu järjestelmiä, jotka tallentavat tietoa jossakin tietyssä tilanteessa (*situation-specific capture systems*) (Sellen & Whittaker, 2010). Näissä tutkimuksissa on huomattu, että esimerkiksi luentojen nauhoittaminen ei mainittavasti paranna oppilaiden arvosanoja (Abowd, 1999).

Tutkimukset vahvistavat myös että digitaaliset arkistot saattavat olla yleisesti ottaen vähemmän arvokkaita kuin mitä ihmiset kuvittelevat niiden olevan (Sellen & Whittaker, 2010). Kun tutkittiin perheiden digitaalisia muistoja, kuten valokuvia, videoita, skannattuja kuvia ja sähköposteja, huomattiin että digitaalisissa arkistoissa ei vierailta kovin usein (Petrelli & Whittaker, 2010). Samassa tutkimuksessa pyydettiin ihmisiä, joilla oli kattava kokoelma digitaalisia muistoja, valitsemaan esineitä, joilla oli heille tunner merkitys. Vain kaksi prosenttia valituista esineistä oli digitaalisia. Toisessa tutkimuksessa on huomattu, että suurin osa ihmisistä, joilla on tuhansien digitaalisten valokuvien arkisto, eivät juuri koskaan katsele suurinta osaa näistä arkistoiduista valokuvista (Whittaker, Bergman & Clough, 2010). Vaikka näistä tutkimustuloksista ei voidakaan suoraan päätellä, että digitaaliset tietoarkistot olisivat sinällään turhia, voidaan kuitenkin miettiä niiden hyödyllisyyttä ja roolia menneisyyden muistelemisessa (Sellen & Whittaker, 2010). Mitä iloa on tuhansien valokuvien digitaalisesta arkistosta jos valokuvia ei koskaan katsella? Miten ihmiset voivat muistella jos he eivät koskaan uppoudu muistoihinsa vaikkapa vanhoja

valokuvia katselemalla? Vaikka lifeloggaus yleistyisikin, kävisikö ihmisten lifelogeille samalla lailla kuin digitaalisille valokuva-arkistoille: ne olisivat olemassa, mutta niitä ei juurikaan käytettäisi?

Allen (2007) on puolestaan huolestunut siitä, että lifeloggaus voi ruokkia liiallista itseensä syventymistä. Koska suurin osa ihmisistä dokumentoi omaan elämäänsä liittyviä asioita mielellään, lifelog-teknologian saatavuus voisi johtaa siihen, että ihmiset antaisivat elämässään liikaa arvoa jopa aivan tyhjänpäiväisille asioille. Allenin mielestä ”suurin osa jokaisesta ihmiselämästä on yhtä korvattavaa ja unohdettavaa kuin massatuotettu soppapurkki”, joten kaikkea ei tarvitsisi dokumentoida. Tämä Allenin mainitsema ilmiö näkyy mielestäni jo hyvin esimerkiksi joidenkin Facebook-käyttäjien toiminnassa. Monet kirjoittelevat status-päivityksiä montakin kertaa tunnissa, ja usein ne ovat sisällöltään ja tarkoitukseltaan melko köyhiä, esimerkiksi ”Kävin kaupassa, ostin maitoa ja juustoa” tai ”Surffailen netissä”. Miksi tällaiset asiat tulisi dokumentoida jälkipolville?

#### 4.2 Haitallinen ”muisti” ja haitallinen valvonta

Allenin (2007) mukaan laajassa, koko elämän kattavassa lifeloggauksessa on kaksi ilmeistä ongelmaa: haitallinen ”muisti” (*pernicious “memory”*) ja haitallinen valvonta (*pernicious surveillance*). Molemmat ovat uhka yksityisyydelle. Muisti on todella hyvä asia, mutta se voi myös rohkaista menneisyyden haitalliseen kaiveluun. Valvonta voi samoin olla todella hyvä asia, mutta se muuttuu sosiaaliseksi pahaksi jos se saa ihmiset turhaan vakoilemaan toisiaan.

Allen viittaa haitallisella ”muistilla” (*pernicious “memory”*) haitallisiin taltiointeihin ja niiden uudelleen läpikäymiseen ja muisteluun. Ihmiset dokumentoivat kokemuksiaan mielellään, ja monet toivovat muistinsa olevan parempi. Allenin mukaan lifeloggaus voi parantaa muistia, koska se auttaa ihmistä muistamaan asioita ja näkemään menneet asiat uudessa valossa. Esimerkiksi ihmiselle, joka usein toisten kanssa keskustellessaan unoh- telee asioita, lifeloggaus olisi korvaamaton apu. Hän voisi nauhoittaa kaikki keskustelut ja tarvittaessa tarkistaa tallenteista, mistä viimeksi tavatessa oli puhuttu. Muistin sähköinen parantelu ei ole kuitenkaan ainoastaan hyvä asia, se näet mahdollistaa tuhoisan muistuttelun ja muistamisen (*destructive reminding and remembrance*).

Allen kertoo kuvitteellisen esimerkin, jossa hän menettää malttinsa ja läpsäyttää ystä- vänsä juhlassa. Tapahtuma tallentuu hänen lifelogiinsa. Pyydettyään lyöntiään anteeksi (ja saatuaan anteeksi), hän päättää poistaa tapahtuman lifelogistaan. Tapahtuma on kuitenkin tallentunut myös muiden juhlassa olleiden vieraiden lifelogeihin, joista hän ei pysty sitä poistamaan. Tällöin video tapahtuneesta saattaa milloin tahansa ilmestyä esiin ja päätyä vaikkapa potentiaalisen työnantajan tai seurustelukumppanin nähtäväksi.

Allen haluaa tällä esimerkillään valaista sitä, että lifeloggaus voi lisätä yksilön epä- onnen ja virheiden elinikää, eikä anna hänelle (eikä muille) mahdollisuutta päästä niistä yli unohtamalla ne. Tehdyt virheet voivat milloin tahansa ilmestyä taas muistuttamaan meitä hetkellisestä typeryydestämme ja tehdä elämästämme hyvin epämiellyttävää. O'Hara ja muut (2009) kuitenkin huomauttavat, että lifeloggaus voisi päinvastoin auttaa

ihmistä oppimaan läksynsä ja ottamaan virheistään opikseen, mitä pelkkä asian unohtaminen ei mahdollista.

Allenin mielestä lifeloggaus aiheuttaa myös menneisyyden ja tulevaisuuden hämartymistä. Entisaikoina, kun ihmisillä ei ollut muuta kuin oma muistinsa ja paperiset arkistot, oman menneisyytensä salaaminen oli helppoa. Elämänsä pystyi halutessaan aloittamaan alusta, puhtaalta pöydältä. Lifelogien yleistyessä tämä ei enää tule olemaan mahdollista: kenestä tahansa voidaan tallentaa mitä tahansa tietoa, ja levitellä sitä missä tahansa. Salasanat ja muut salauskeinot varmistavat ehkä sen, että lifelogit pysyvät salassa vierailta ihmisiltä, mutta sosiaaliset normit aiheuttavat sen, että emme pysty piilottamaan lifelogejamme läheisiltämme. Ja vaikka voisimmekin tehdä näin, joudumme silti, tahdoimmepa tai emme, esiintymään muiden ihmisten lifelogeissa. Ja toisten ihmisten tallentamasta tiedosta emme pysty määräämään, mihin ja miten sitä käytetään.

Lifelogit myös mahdollistavat menneisyyden murehtimisen, joka saattaa olla haitaksi yksilön mielenterveydelle. Esimerkiksi jonkin traumaattisen kokemuksen yksityiskohtien unohtamisella voi olla suuri merkitys ihmisen hyvinvoinnille. Sodasta selviytyneelle ihmiselle on hyväksi, jos hän käsiteltyään traumaattisen asian pystyy unohtamaan sen ja jatkamaan elämäänsä. Mutta jos traumaattiset kokemukset ovat tallentuneet hänen lifeloggiinsa, ei niiden unohtaminen ole helppoa. Tämäntyyppinen ongelma on Allenin mukaan helppo ratkaista tekemällä lifeloggaus-järjestelmistä sellaisia, että niistä on helppo poistaa jotakin tietoa, esimerkiksi tietyn aikavälin kokemuksia. Tämä ei kuitenkaan poista jo edellä mainittua ongelmaa eli sitä, että toisten lifelogeista ei voi poistaa tietoa, ainoastaan omastaan. Traumaattiset kokemukset saattavat siis ilmestyä uudelleen, tosin eivät ehkä niin helposti, mutta mahdollisuus on kuitenkin olemassa. Ja eikö tavallaan olisi entistä stressaavampaa, jos tietäisi, että saattaa joutua vielä kohtaamaan nuo asiat, mutta ei tiedä milloin ja missä?

Toinen lifeloggauksen mahdollinen haitta mielenterveydelle on Allenin mukaan patologinen menneisyyden kaivelu ja murehtiminen. Lifeloggaus mahdollistaa esimerkiksi masentuneelle ihmiselle menneiden, negatiivisten tapahtumien (epäonnistumisten, epäoikeudenmukaisuuksien kokemisen) jatkuvan uudelleenelämisen ja murehtimisen. Tällaisessa tapauksessa voi olla vaikeaa saada masentunutta ihmistä ymmärtämään, että lifeloggiin tallentunut tieto ei välttämättä edusta koko totuutta hänen elämästään, vaan lifeloggiin tallentuneet asiat tulisi nähdä ennemminkin tulkinnanvaraisina ja epäoleellisina.

Allen raottaa myös haitallisen valvonnan (*pernicious surveillance*) käsitettä. Se viittaa siihen, että lifelogien avulla voi saada hyvin paljon erilaista tietoa lifeloggia pitävän yksilön elämästä. Toki nykyäänkin ihmisistä kerätään ja tallennetaan suuri määrä tietoa, joko ihmisen itsensä tai toisten toimesta. Lifelogien yleistyessä hallitus ja yritykset voisivat kuitenkin kerätä rutiininomaisesti ja systemaattisesti yksityiskohtaisia tietoja yksilöistä markkinointi- ja turvallisuustarkoituksiin, sekä sosiaalisen kontrollin ylläpitämiseksi. Lisäksi lifeloggaus saa aikaan sen, että ihmisten tallentaessa omaa elämänsä, he tallentavat samalla myös toisten ihmisten elämiä. Näin heistä tulee tavallaan toisten ihmisten tarkkailijoita ja silmälläpitäjiä. Lifelogeja voitaisiin näin käyttää toisten ihmisten vakoi- luun.

## 5. Lopuksi

Tässä tutkielmassa olen esitellyt toimintaa nimeltä lifeloggaus. Ensin kävin läpi lifeloggauksen taustaa esittelemällä Vannevar Bushin Memexin, jota voidaan pitää ensimmäisenä visiona lifeloggaus-teknologiasta. Vaikka Bushin visio sai paljon huomiota ja kiinnostusta osakseen omana aikanaan, nykyajan versio Memexistä on hyvin erilainen. Bushin visiossa yhdellä laitteella pystyi tallentamaan kaikenlaista tietoa, mutta nykyajan lifeloggaajat joutuvat käyttämään monia erilaisia laitteita elämänsä tallennukseen, esimerkiksi tietokoneita ja matkapuhelimia (Byrne & Jones, 2008). Toisaalta nykyajan ihmiselle on hyvin luontevaa käyttää sujuvasti erilaisia laitteita yhdessä ja erikseen, ja esimerkiksi uudet matkapuhelimet alkavat muistuttaa yhä enemmän tietokoneita kuin varsinaisia matkapuhelimia. Ehkäpä nykyajan lifeloggaajat eivät siis olekaan niin kaukana Bushin Memexistä kuin äkkiseltään luulisi.

Memexin esittelyn jälkeen kävin läpi kolme lifeloggaus-projektia: DARPA:n LifeLog-projektin, Steve Mannin ja hänen EyeTap-laitteensa ja MyLifeBits-projektin. Nämä ovat kaikki mielenkiintoisia projekteja, esimerkiksi Steve Mannin EyeTap-laitetta voidaan käyttää hyvin paljon muuhunkin kuin vain elämän tallentamiseen.

Kolmannessa luvussa esittelin lifeloggauksen hyötyjä. Lifeloggausta voitaisiin käyttää hyvin monella tavalla helpottamaan ihmisten elämää ja parantamaan heidän elämänlaatuaan. Ilmeisin hyöty on se, että lifelogia pitävällä ihmisellä on aina mahdollisuus muistaa tarkasti, mitä hänen elämässään on tapahtunut. Lifelogiaan selailemalla voi tarkastaa, mitä työpaikan palaverissa käytiinkään läpi, tai mitä puolison kanssa sovittiin viikonlopun vietosta. Lifeloggausta voidaan käyttää myös muistihäiriöistä kärsivien ihmisten apuna, kuten esittelemässäni Leen ja Deyn tutkimuksessa on tehty. Lifeloggaus voi helpottaa elämää paitsi toimimalla elektronisena muistina, myös avustamalla ihmistä erilaisissa toimissa, kuten O'Hara ja muut (2009) visioivat.

Hyötyjen lisäksi lifeloggauksessa on myös ongelmia. Näitä ongelmia esittelin neljännessä luvussa. Pääasiassa ongelmat liittyvät ihmisten yksityisyyteen ja siihen, onko loppujen lopuksi kovin hyvä asia, jos ihminen ei voi unohtaa asioita, vaan ne ovat aina löydettävissä omasta tai toisten lifelogeista. Digitaalisten tietoarkistojen hyödyllisyys voidaan myös kyseenalaistaa, kuten muun muassa Sellen ja Whittaker (2010) ovat tehneet.

Nykyään monet ihmiset tallentavat itsestään jonkin verran tietoa internetiin, esimerkiksi Facebook-profiileihin ja blogeihin. Tästä on kuitenkin vielä pitkä matka koko elämän digitaaliseen tallentamiseen. On jännittävää nähdä, yleistyykö toiminta tulevaisuudessa. Ongelmatonta se ei kuitenkaan tule olemaan. Yksityisyysongelmat ovat ilmeiset, mutta samoin ovat lifeloggauksen tarjoamat hyödyt. Pitää vain punnita, kummat painavat vaakakupissa enemmän.

### Viiteluettelo

Abowd, G. D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38, 508–530.

- Allen, A.L. (2007). Dredging-up the past: Lifelogging, memory and surveillance. *Scholarship at Penn Law*. Paper 173. [http://lsr.nellco.org/upenn\\_wps/173/](http://lsr.nellco.org/upenn_wps/173/)
- Bell, G. (2001). A personal digital store. *Communications of the ACM*, 44, 86–91.
- Bush, V. (1945). As we may think. In: D. Trend (ed.). *Reading Digital Culture*, Blackwell Publishing, 9–13.
- Byrne, D. & Jones, G. J. F. (2008). Towards computational autobiographical narratives through human digital memories. *Proceedings of the 2008 ACM International Conference on Multimedia, with Co-Located Symposium & Workshops*, 9–12.
- Gemmell, J., Bell, G., Lueder, R. (2006) MyLifeBits: A personal database for everything. *Communications of the ACM*, 49, 89–95.
- Gemmell, J., Bell, G., Lueder, R., Drucker, S., & Wong, C. (2002) MyLifeBits: Fulfilling the Memex vision. *ACM Multimedia '02*, 235–238.
- IPTO. (2003). LifeLog Proposer Information Pamphlet, Defense Advanced Research Projects Agency Information Processing Technology Office Document PIP\_03.30 [http://web.archive.org/web/20030603173339/http%3A//www.darpa.mil/ipto/Solicitations/IP\\_03-30.html](http://web.archive.org/web/20030603173339/http%3A//www.darpa.mil/ipto/Solicitations/IP_03-30.html)
- Lee, M. L. & Dey, A. K. (2008). Lifelogging memory appliance for people with episodic memory impairment. *Proceedings of the Tenth International Conference on Ubiquitous Computing*, 44–54.
- Mann, S. (2004). Continuous lifelong capture of personal experience with EyeTap. *Proceedings of the First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 1–21.
- Nack, F. (2005). You must remember this. *IEEE Multimedia*, 12, 4–7.
- O'Hara, K., Tuffield, M. M. & Shadbolt, N. (2009). Lifelogging: Privacy and empowerment with memories for life. *Identity in the Information Society*, 1, 155–172.
- Petrelli, D. & Whittaker, S. (2010). Family memories in the home: contrasting physical and digital mementos. *Personal and Ubiquitous Computing*, 14, 153–169.
- Shachtman, N. (2004). Pentagon kills LifeLog project. *Wired*, 12.4.2010. <http://www.wired.com/politics/security/news/2004/02/62158>
- Sellen, A. J. & Whittaker, S. (2010). Beyond total capture: a constructive critique of lifelogging. *Communications of the ACM*, 53, 70–77.

Whittaker, S., Bergman, O. & Clough, P. (2010). Easy on that trigger dad: a study of long term family photo retrieval. *Personal and Ubiquitous Computing*, 14, 31–43.

# Polunetsintä tietokonepeleissä

Ville Rahikainen

## Tiivistelmä

Tässä tutkielmassa tarkastellaan tietokonepelien polunetsinnässä käytettäviä algoritmeja ja niiden tehostamiseen kehitettyjä tekniikoita. Tutkielmassa tarkastellaan myös yleisiä, käytettävästä algoritmista riippumattomia haun yksinkertaistamis- ja nopeuttamistekniikoita sekä tekniikoita polun esteettisyyden parantamiseen.

**Avainsanat ja -sanonnat:** A\*, IDA\*, reunahaku, optimointi

**CR-luokat:** G.2.2

## 1. Johdanto

Tietokonepelien suosion kasvaessa myös odotukset aina vain realistisemmasta pelikokemuksesta ovat yleistyneet. Pelit eivät enää ole vain pienen, tietokoneista ja ohjelmoinnista kiinnostuneen joukon juttu. Niistä on tullut suurten massojen päivittäinen viihdemuoto, joka kilpailee myyntiarvossaan tasapäisesti elokuvien ja musiikin kanssa [Chatfield, 2009]. Polunetsintä (path finding) on lähinnä tietokonepelien yhteydessä käytetty termi, joka tarkoittaa esteettömän polun etsimistä 2D- tai 3D-kartan pisteestä  $s$  pisteeseen  $t$ . Tässä tutkielmassa  $s$ :llä merkitään etsinnän aloitusolmua ja  $t$ :llä etsinnän lopetusolmua. Kuvissa 1 ja 2 on esimerkit 2D- ja 3D-kartoista.



Kuva 1. 2D-kartta, Baldur's Gate II: ToB pelistä



Kuva 2. 3D-kartta, Gothic 3 pelistä



Kartan kuljettavista alueista muodostetaan graafi, josta polku voidaan etsiä käyttäen graafin tarkasteluun tarkoitettuja algoritmeja.

Todentuntuisen grafiikan ja vakuuttavan fysiikanmallinnuksen ohella älykkäät polut ovat yksi tärkeimmistä pelin realistisuuden muodostavista tekijöistä. Hiekkasäkkien taakse jumiin jäävät sotilaat, jyrkkiin mäkiin takertuvat apulaiset ja selkeästi kiinteistä esteistä vaivattomasti läpi juoksevat vastustajat eivät vain romuta pelin immersiota, vaan ne ovat aito turhautumisen lähde pelaajalle, jonka tarvitsee rentoutumisensa ohessa alkaa avustamaan jumiutunutta tekoälyä.

Polkua valittaessa huomioon otettavia asioita ovat polkua tarvitsevan yksikön tyyppi, koko, kääntymissäde ja nopeus. Tyyppi määrittää, millaisessa ympäristössä yksikkö voi liikkua – vene voi liikkua vedessä, mutta ei maalla. Koko määrittää mihin yksiköllä on mahdollista mennä – henkilö mahtuu hissiin, mutta auto ei. Koko vaikuttaa myös siihen, kuinka läheltä estettä yksikölle annettavan polun on mahdollista kulkea. Yksikön kääntymissäde vaikuttaa valittavaan polkuun määrittämällä, millainen liike yksikölle on mahdollista. Esimerkiksi ihminen voi suorittaa  $90^\circ$  käännöksen paikallaan, mutta autolle vastaava käännös on mahdoton. Yksikön nopeus vaikuttaa sen kääntymissäteeseen ja sitä kautta sen polkuun.

Pelikartalla voi myös olla monia erilaisia maastotyyppisiä, kuten aavikko, savanni, suo, maantie, jne. Maastotyyppit voidaan erottaa toisistaan määrittelemällä niillä kulkemiselle kerroin. Maantie on helppokulkuinen, joten sillä etäisyyden 5 omaava kaari saisi kertoimekseen esimerkiksi 1. Suo sen sijaan on vaikeakulkuinen, joten vastaava kaari voisi saada kertoimen 5. Samalla periaatteella voidaan myös ohjata yksiköjä tunnettujen vaara-alueiden ohi. Esimerkiksi alueille, jotka tiedetään ansoitetuksi, voitaisiin määritellä kertoimeksi 10 [Jönsson, 1997].

Tavallinen pelaaja ei ole tietoinen, saati sitten kiinnostunut, polunetsinnän käytössä olevista rajallisista muisti- ja prosessoriresursseista tai uskottavan polunetsinnän toteuttamisen hankaluudesta [Botea *et al.*, 2004]. Häntä kiinnostaa vain ruudulla avautuva lopputulos.

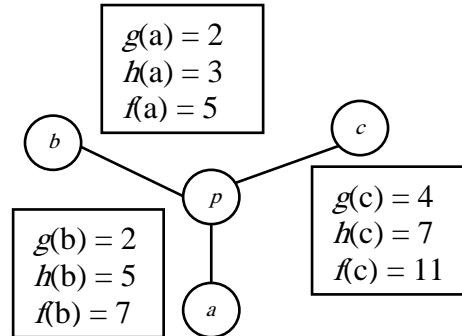
Tässä tutkielmassa tarkastellaan tietokonepelien polunetsinnässä käytettäviä algoritmeja ja niiden tehostamiseen kehitettyjä tekniikoita. Tutkielmassa tarkastellaan myös yleisiä, käytettävästä algoritmista riippumattomia haun yksinkertaistamis- ja nopeuttamistekniikoita sekä tekniikoita polun esteettisyyden parantamiseen.

## 2. A\*

A\* on vuonna 1968 esitetty ”paras ensin” algoritmi (best-first). Se on pääasiallisesti käytetty ratkaisu tietokonepelien polunetsintäongelmiin [Björnsson *et al.*, 2005]. Polkua muodostaessaan A\* arvioi mahdollisen seuraavan solmun  $n$  hyvyyden käyttäen tähän asti kuljetun polun pituuden ja jäljellä olevan matkan arvion yhdistelmää. Tämä arvio ilmaistaan kaavana muodossa

$$f(n) = g(n) + h(n),$$

missä  $g(n)$  on lyhin tunnettu polku solmusta  $s$  solmuun  $n$ ,  $h(n)$  on heuristinen arvio matkasta solmusta  $n$  solmuun  $t$  ja  $f(n)$  on arvio lyhimmästä polusta solmusta  $s$  solmuun  $t$  kulkien solmun  $n$  kautta [Hart *et al.*, 1968; Nilsson, 1980; Matthews, 2002]. Kuvassa 3 on yksinkertainen esimerkki A\*-algoritmin soveltamisesta.



Kuva 3. Seuraavan solmun valinta.  $p$  on tarkasteltavan polun alkuosan viimeinen solmu ja  $a$ ,  $b$  ja  $c$  solmuja, joihin solmusta  $p$  on mahdollista liikkua. Seuraavaksi tarkasteltavaksi valittaisiin  $a$ , koska se saa pienimmän arvon  $f(n)$ :lle.

A\* on pätevä (admissible), jos käytetty heuristinen arvio on pienempi tai yhtäsuuri kuin haetun polun varsinainen pituus. Pätevyydellä tarkoitetaan, että A\* löytää aina mistä tahansa graafista lyhimmän mahdollisen polun solmusta  $s$  solmuun  $t$ , jos tällainen polku on olemassa [Hart *et al.*, 1968; Nilsson, 1980; Russell and Norvig, 2010].

A\*:n on väitetty olevan optimaalinen. Optimaalisuus tarkoittaa, että jos kahdella algoritmilla A\* ja B on käytössään täysin samat tiedot ongelmaympäristöstä, ei algoritmi B voi löytää lyhyintä polkua solmuun  $t$  tarkastelemalla pienemmän määrän solmuja kuin A\* [Hart *et al.*, 1968; Nilsson, 1980]. Alkuperäinen todistus ei kuitenkaan ota huomioon kahta tapaa, joilla algoritmin B on mahdollista löytää polku tehokkaammin kuin A\*:n. Ensiksi B voi edetä graafissa alueille, joihin A\*:n heuristiikat eivät ohjaa sitä, ja näiltä, A\*:n huomiotta jättämillä alueilla saamansa tiedon avulla B voi muodostaa A\*:ä tehokkaamman ratkaisun. Toiseksi A\* ei käytä tietoa jo läpikäymistään solmuista heuristisen arvionsa muodostamiseen, mutta B:llä ei ole tätä rajoitusta. Algoritmi B voi siis tarkastella tietoa jo läpikäymistään solmuista ja huomata, että arvon  $h(n)$  kuuluisi olla korkeampi kuin mitä sen tarkastelu tyhjiössä antaa ymmärtää. Tästä seuraa, noudattaen A\*:n normaalia toimintaa, että lähempänä oikeaa arvoa oleva heuristinen arvio voi vähentää tarkasteltavien solmujen määrää, jolloin B olisi A\*:ä tehokkaampi [Dechter and Pearl, 1985]. Tarkasteltavan graafin muoto voi vaikuttaa tähän yleiseen sääntöön. Jos graafissa on vain yksi polku, joka johtaa loputussolmua kohti, eivät suuretkaan erot heuristisessa arviossa vaikuta polun valintaan.

A\* ei siis ole optimaalinen kaikissa tapauksissa [Dechter and Pearl, 1985]. Jos käytetty heuristinen arvio täyttää monotonisuusrajoituksen (monotonicity restriction), A\* on myös optimaalinen.

Monotonisuusrajoitus vaatii, että  $h(n_i) \leq d(n_i, n_j) + h(n_j)$ , missä  $d(n_i, n_j)$  on solmujen  $n_i$  ja  $n_j$  välinen etäisyys. Kyseessä on siis kolmioepäyhtälö. Monotonisuusrajoituksen ollessa täytettynä voidaan uutta solmua valittaessa olla varmoja, että solmuun on löytynyt optimaalinen polku. Valittavat solmut ovat optimaalisia, koska kolmioepäyhtälö takaa, että arvio matkasta solmusta  $n_i$  solmuun  $t$  (kolmion sivu) voi olla korkeintaan yhtä suuri kuin kolmion kahden muun sivun pituuden summa. Kaksi muuta sivua olisivat matka solmusta  $n_i$  mihin tahansa muuhun solmuun  $n_j$  plus matka solmusta  $n_j$  solmuun  $t$ . Koska valittavat solmut ovat optimaalisia, ei ole tarpeellista varmistaa, onko niihin jo aikaisempaa polkua [Hart *et al.*, 1968; Nilsson, 1980; Russell and Norvig, 2010; Dechter and Pearl, 1985].

Heuristinen arvio  $h(n)$  saadaan käyttämällä hyväksi tunnettua tietoa kuljettavasta graafista ja se käytännössä ratkaisee  $A^*$ :n nopeuden, tehokkuuden ja pätevyyden. Kun käytetään  $h(n)$ :n arvoa 0, vastaa  $A^*$  käytännössä leveyshakua, joka on varsin tehoton. Jos  $h(n)$ :n arvo on sen sijaan täsmälleen sama kuin matka solmusta  $n$  solmuun  $t$ , niin  $A^*$  tutkii pienimmän mahdollisen määrän solmuja matkallaan solmuun  $t$ . Jos  $h(n)$  on suurempi kuin matka solmusta  $n$  solmuun  $t$ ,  $A^*$  valitsee solmuja, jotka ovat lähempänä lopetussolmua. Se voi myös toimia nopeammin ja tehokkaammin, mutta takeita lyhyimmän polun löytymisestä ei enää ole. Tietokonepeleissä ei olla niinkään kiinnostuneita juuri lyhyimmästä polusta, vaan melkein lyhyin polku nopeammin ja vähemmän resursseja kuluttaen saatuna on mielekäs vaihtoehto lyhyimpään polkuun verrattuna [Hart *et al.*, 1968; Nilsson, 1980; Patel, 2009].

Jos solmun  $s$  ja solmun  $t$  välissä ei ole esteitä, joita ei voi kiertää, hienoinen yliarviointi johtaa yleensä pienempään tutkittavien solmujen määrään ja nopeampaan toimintaan. Jos välissä sen sijaan on este, jota ei voida kiertää, johtaa yliarviointi suurempaan määrään tutkittavia solmuja ja täten hitaampaan toimintaan [Rabin, 2000b]. Yhtä oikeaa yliarviointia kaikkiin peliympäristöihin ei ole olemassa, vaan se on määritettävä erikseen jokaiselle pelille. Riippuen pelialueen esittämiseen käytetystä tavasta yliarvioinnin määrittelyyn voi käyttää joko tunnettujen solmujen ja tarkasteltujen solmujen maksimi määrää tai haun muodostaman alueen muotoa. Sopivassa yliarvioinnissa hakualueen muoto muistuttaa timanttia tai kuusikulmiota, jossa solmut  $s$  ja  $t$  ovat hakualueen ääripäissä. Juuri mahdollisuus muuttaa  $A^*$ :n käyttäytymistä tilanteesta riippuen, muuttamalla heuristiikan määrittelyä, tekee  $A^*$ :sta mielekkään algoritmin tietokonepelien polunetsintään [Rabin, 2000b; Patel, 2009].

Yleisesti käytettyjä heuristiikan arviointi metodeja ovat Euklidinen etäisyys (Euclidean distance) ja Manhattanin etäisyys (Manhattan distance) eli korttelietäisyys [Patel, 2009; Rabin, 2000b]. Euklidinen etäisyys on janan pituus kahden  $k$ -ulotteisen pisteen  $p = (p_1, \dots, p_k)$  ja  $q = (q_1, \dots, q_k)$  välillä euklidisessa avaruudessa, ja se saadaan kaavasta

$$\sqrt{\sum_{i=1}^k (p_i - q_i)^2}.$$

Manhattanin etäisyys on kahden  $k$ -ulotteisen pisteen  $p = (p_1, \dots, p_k)$  ja  $q = (q_1, \dots, q_k)$  etäisyys vektoriarvuudessa, ja se saadaan kaavasta

$$\sum_{i=1}^k |p_i - q_i|.$$

$A^*$ :n varsinainen toteutus ylläpitää kahta tietorakennetta, joista käytetään yleensä nimitystä OPEN-lista ja CLOSED-lista. Tämän lisäksi jokainen löydetty solmu tuntee edeltäjänsä. OPEN-lista sisältää ne solmut, jotka on löydetty, mutta joita ei vielä ole tarkasteltu. CLOSED-lista sisältää ne solmut, joita on jo tarkasteltu [Patel, 2009; Matthews, 2002]. Haun alussa OPEN-lista sisältää solmun  $s$  ja CLOSED-lista on tyhjä. Solmua  $s$  ympäröivät solmut eli solmut, joihin solmusta  $s$  voidaan liikkua, lisätään OPEN-listaan. Solmu  $s$  poistetaan OPEN-listasta ja lisätään CLOSED-listaan. Seuraavaksi tarkasteltavaksi solmuksi valitaan se solmu OPEN-listasta, joka saa pienimmän arvon  $f(n)$ :lle. Jos kaksi solmua saa saman arvon, on niiden paremmuuden ratkaisu ohjelmoijasta kiinni. Jo OPEN-listassa olevaa solmua voidaan pitää automaattisesti parempana tai huonompana, tai solmujen paremmuus voidaan ratkaista vertaamalla  $g(n)$ :n tai  $h(n)$ :n arvoa. Jos uusi solmu on solmu  $t$ , on algoritmin suoritus päättynyt. Muussa tapauksessa tarkastellaan taas ympäröiviä solmuja [Matthews, 2002; Stout, 2000].

Koska käytetty heuristiikka vaihtelee tilanteesta riippuen, ja se voi myös olla yliarvio, ei monotonisuusrajituksen voida olettaa täyttyvän. Siksi OPEN-listassa ja CLOSED-listassa olevista solmuista tarvitsee tarkistaa, onko solmuun löytynyt uusi, lyhyempi polku. Jos tämä on uusi lyhyempi polku ja solmu on OPEN-listassa, niin solmua edeltävä solmu ja solmuun johtavan polun pituus päivitetään. Jos solmu on CLOSED-listassa, se avataan uudelleen. Solmut, joita ei ole kummassakaan, lisätään OPEN-listaan. Tätä jatketaan, kunnes saavutaan solmuun  $t$ . Solmun  $t$  löytymisen jälkeen kuljettu polku solmuun  $t$  saadaan seuraamalla solmujen edeltäjäketjua takaisin solmuun  $s$  [Patel, 2009; Stout, 2000].

Yleisin tapa toteuttaa OPEN-lista on prioriteettijono. Sen mahdollisia toteuttamisrakenteita ovat järjestämätön taulu, järjestämätön linkitetty lista, järjestetty taulu, järjestetty linkitetty lista, keko ja tasapainotettu binääripuu. Järjestämätön taulu ja järjestämätön linkitetty lista suorittavat lisäyksen ajassa  $O(1)$  ja poiston ajassa  $O(m)$ , missä  $m$  on listan solmujen määrä. Järjestetty taulu ja järjestetty linkitetty lista suorittavat lisäyksen ajassa  $O(m)$  ja seuraavan solmun haun ajassa  $O(1)$ . Keossa ja tasapainotetussa binääripuussa lisäys-, poisto- ja etsintäoperaatiot suoriutuvat ajassa  $O(\log m)$ . Yleisesti kekon ja järjestettyyn binääripuuhun perustuvat ratkaisut ovat tehokkaampia, jos suoritettavat haut ovat suuria, ja tauluun sekä listaan perustuvat ratkaisut ovat varteenotettavia vain pienemmissä hakutehtävissä [Stout, 1999; Higgins, 2002].

OPEN-listalle on myös mahdollista määritellä maksimikoko. Kun OPEN-lista on täynnä, poistetaan huonoimman  $f(n)$ :n arvon omaava solmu OPEN-listasta. Tämä saattaa johtaa samojen solmujen lisäämiseen OPEN-listaan moneen kertaan, mikä hidastaa hakua [Stout, 1999].

Huomattavaa kuitenkin on, että lisäys- ja poisto-operaatioiden suhde voi olla  $8 : 1$ , esimerkiksi ruudukossa yhdellä ruudulla voi olla 8 naapuria. Tällainen tilanne suosii vahvasti

järjestämättömän taulun tai listan  $O(1)$  lisäysaikaa. Järjestämättömän toteutuksen poisto-operaation hitaus on mahdollista kiertää ylläpitämällä ylimääräistä järjestettyä tietorakennetta, ns. halpaa listaa (cheap list). Halvalla listalla on ennalta määrätty pieni koko (esimerkiksi 15), kun sitä täytetään. Lisäyksiä suoritettaessa listan koko voi kumminkin kasvaa suuremmaksi, koska siitä ei suoriteta siirtoja OPEN-listaan. Halvassa listassa pidetään solmuja, joilla on pienempi  $f(n)$ -arvo, ja seuraava tarkasteltava solmu on mahdollista saada halvasta listasta ajassa  $O(1)$  tai  $O(\log m)$ , toteutustavasta riippuen. Uutta solmua lisättäessä tarkistetaan, onko sen  $f(n)$  arvo pienempi kuin halvan listan viimeinen solmu, ja jos se on, suoritetaan järjestetty lisääminen. Muussa tapauksessa se lisätään suoraan OPEN-listaan. Kun kaikki halvassa listassa olleet solmut on käyty läpi, täytetään halpa lista uudestaan käymällä läpi OPEN-lista [Higgins, 2002].

---

## A\*-algoritmi

---

Open: hakusolmu prioriteettijono

Closed: hakusolmu lista

AStarSearch( location StartLoc, location GoalLoc, agenttype Agent )

```

clear Open ja Closed
StartNode.Loc ← StartLoc
StartNode.CostFromStart ← 0
StartNode.CostToGoal ← PathCostEstimate( StartLoc, GoalLoc, Agent )
StartNode.Parent ← null
push StartNode Openiin
while Open ei ole tyhjä
  pop Node Openista
  if Node goal solmu
    muodosta polku takaperin Node:sta StartLoc:iin
    return success
  else
    for each Noden jälkeläiselle NewNode
      NewCost ← Node.CostFromStart + TraverseCost( Node, NewNode, Agent )
      if NewNode on Openissa tai Closedissa ja NewNode.CostFromStart ≤ NewCost
        continue
      else
        NewNode.Parent ← Node
        NewNode.CostFromStart ← NewCost
        NewNode.CostToGoal ← PathCostEstimate( NewNode.Loc, GoalLoc, Agent )
        NewNode.TotalCost ← NewNode.CostFromStart + NewNode.CostToGoal
        if NewNode on Closedissa
          remove NewNode Closedista
        if NewNode Openissa
          adjust NewNode's location in Open
        else
          push NewNode Openiin
    push Node Closediin
return failure

```

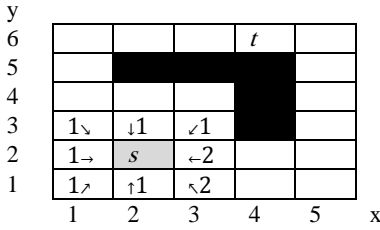
---

### Algoritmi 1. A\* [Stout, 2000]

Seuraavaksi käydään läpi algoritmissa 1 esitetyn A\*:n esimerkkihaku. Heuristiikkana on käytetty Manhattanin etäisyyttä; samansuuruisen  $f(n)$  arvojen tapauksessa suuremman  $g(n)$ :n arvon omaavaa solmua pidetään parempana. Jos  $f(n)$ :llä ja  $g(n)$ :llä on sama arvo, niin aiemmin OPEN-listaan syötettyä solmua pidetään parempana. Esimerkissä solmuja ei avata uudestaan, vaan solmun arvo päivitetään suoraan, jos siihen löytyy uusi, lyhyempi polku. Mustalla täytetyt ruudut ovat alueita, joilla ei voi liikkua,  $s$  osoittaa aloitussolmua,  $t$  lopetussolmua ja harmaa tausta tällä

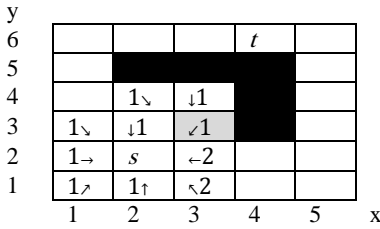
hetkellä tarkasteltavaa solmua. Nuolet osoittavat solmun edeltäjän, ja numerot kertovat etäisyyden edelliseen solmuun.

$$\begin{aligned} h(2,1) &= |2-4|+|1-6| = 7 & f(2,1) &= 1 + 7 = 8 \\ h(3,1) &= |3-4|+|1-6| = 6 & f(3,1) &= 2 + 6 = 8 \\ h(3,2) &= |3-4|+|2-6| = 5 & f(3,2) &= 2 + 5 = 7 \\ h(3,3) &= |3-4|+|3-6| = 4 & f(3,3) &= 1 + 4 = 5 \\ h(2,3) &= |2-4|+|3-6| = 5 & f(2,3) &= 1 + 5 = 6 \\ h(1,3) &= |1-4|+|3-6| = 6 & f(1,3) &= 1 + 6 = 7 \\ h(1,2) &= |1-4|+|2-6| = 7 & f(1,2) &= 1 + 7 = 8 \\ h(1,1) &= |1-4|+|1-6| = 8 & f(1,1) &= 1 + 8 = 9 \end{aligned}$$



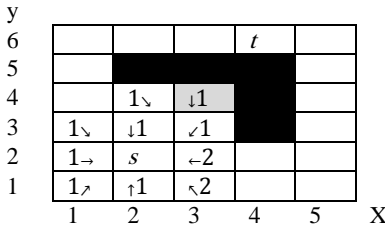
OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(2,1)1+7	
(1,2)1+7	
(3,1)2+6	
(1,3)1+6	
(3,2)2+5	
(2,3)1+5	
(3,3)1+4	

$$\begin{aligned} h(3,4) &= |3-4|+|4-6| = 3 & f(3,4) &= 2 + 3 = 5 \\ h(2,4) &= |2-4|+|4-6| = 6 & f(2,4) &= 2 + 6 = 8 \end{aligned}$$



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(2,1)1+7	(3,3)1+4
(1,2)1+7	
(3,1)2+6	
(1,3)1+6	
(3,2)2+5	
(2,3)1+5	
(2,4)2+4	
(3,4)2+3	

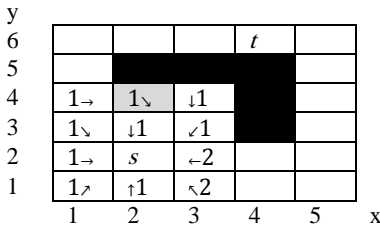
Kaikki ympäröivät solmut joko jo tunnettuja tai suljettuja tai aluetta, jolla ei voi kulkea.



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(2,1)1+7	(3,3)1+4
(1,2)1+7	(3,4)2+3
(3,1)2+6	
(1,3)1+6	
(3,2)2+5	
(2,3)1+5	
(2,4)2+4	

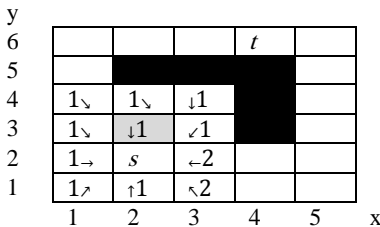
Kulman läpi ei voida kulkea/havaita uutta solmua.

$$h(1,4) = |1-4|+|4-6| = 5 \quad f(1,4) = 3 + 5 = 8$$



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(2,1)1+7	(3,3)1+4
(1,2)1+7	(3,4)2+3
(3,1)2+6	(2,4)2+4
(1,4)3+5	
(1,3)1+6	
(3,2)2+5	
(2,3)1+5	

Uusi reitti (1,4),  $f(1,4) = 2 + 5 = 7$



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(2,1)1+7	(3,3)1+4
(1,2)1+7	(3,4)2+3
(3,1)2+6	(2,4)2+4
(1,3)1+6	(2,3)1+5
(1,4)2+5	
(3,2)2+5	

$$h(4,2) = |4-4| + |2-6| = 4 \quad f(4,2) = 4 + 4 = 8$$

$$h(4,1) = |4-4| + |1-6| = 5 \quad f(4,1) = 4 + 5 = 9$$

y				t	
6					
5					
4	1↘	1↘	↑1		
3	1↘	↑1	↘1		
2	1→	s	←2	←2	
1	1↗	↑1	↘2	↘2	
	1	2	3	4	5 x

OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,3)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	
(1,4)2+5	

$$h(1,5) = |1-4| + |5-6| = 4 \quad f(1,5) = 3 + 4 = 7$$

y				t	
6					
5	1↓				
4	1↘	1↘	↑1		
3	1↘	↑1	↘1		
2	1→	s	←2	←2	
1	1↗	↑1	↘2	↘2	
	1	2	3	4	5 x

OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,3)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
(1,5)3+4	

$$h(1,6) = |1-4| + |6-6| = 3 \quad f(1,6) = 4 + 3 = 7$$

y				t	
6	1↓				
5	1↓				
4	1↘	1↘	↑1		
3	1↘	↑1	↘1		
2	1→	s	←2	←2	
1	1↗	↑1	↘2	↘2	
	1	2	3	4	5 x

OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,3)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
(1,6)4+3	(1,5)3+4

$$h(2,6) = |2-4| + |6-6| = 2 \quad f(2,6) = 5 + 2 = 7$$

y				t	
6	1↓	←1			
5	1↓				
4	1↘	1↘	↑1		
3	1↘	↑1	↘1		
2	1→	s	←2	←2	
1	1↗	↑1	↘2	↘2	
	1	2	3	4	5 x

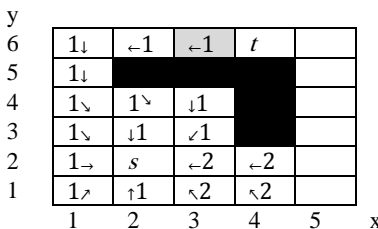
OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,3)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
(2,6)5+2	(1,5)3+4
	(1,6)4+3

$$h(3,6) = |3-4| + |6-6| = 1 \quad f(3,6) = 6 + 1 = 7$$

y				t	
6	1↓	←1	←1		
5	1↓				
4	1↘	1↘	↑1		
3	1↘	↑1	↘1		
2	1→	s	←2	←2	
1	1↗	↑1	↘2	↘2	
	1	2	3	4	5 x

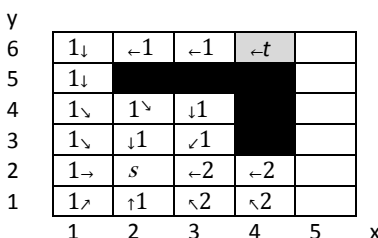
OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,3)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
(3,6)6+1	(1,5)3+4
	(1,6)4+3
	(2,6)5+2

$$h(4,6) = |4-4| + |6-6| = 0 \quad f(4,6) = 7 + 0$$



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,4)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
(4,6)7+0	(1,5)3+4
	(1,6)4+3
	(2,6)5+2
	(3,6)6+1

Solmu  $t$  löytyi. Kuljettu polku saadaan seuraamalla edeltävyyttä osoittavia nuolia ja kääntämällä tämä järjestys ympäri: (2,2), (2,3), (1,4), (1,5), (1,6), (2,6), (3,6), (4,6).



OPEN	CLOSED
(1,1)1+8	(2,2)0+5
(4,1)4+5	(3,3)1+4
(2,1)1+7	(3,4)2+3
(1,2)1+7	(2,4)2+4
(3,1)2+6	(2,4)1+5
(4,2)4+4	(3,2)2+5
(1,3)1+6	(1,4)2+5
	(1,5)3+4
	(1,6)4+3
	(2,6)5+2
	(3,6)6+1
	(4,6)7+0

### 3. IDA\*

Iterative Deepening A\* eli IDA\* perustuu Depth First Iterative Deepening (DFID)-algoritmiin, joka on leveys- ja syvyyshakujen yhdistelmä. DFID on vanha algoritmi, joka esitettiin ensimmäistä kertaa jo 1960-luvulla keinona shakkiohjelmien ratkaisun etsintään. Sen viimeisimmän uudelleen löytymisen on esittänyt Richard E. Korf vuonna 1985. DFID suunniteltiin ratkaisemaan leveyshaun tilaongelmat ja syvyysaun ratkeamattomuuden ja lopettamissyvyyden määrittelyn ongelmat. DFID toimii suorittamalla syvyysaun määritelyyn kynnsarvoon asti. Jokaisen iteraation, jossa solmu  $t$  ei ole löytynyt, päätteeksi kynnsarvoa kasvatetaan yhdellä. Ensimmäisessä iteraatiossa kynnsarvo on yksi, ja siinä tehdään syvyyshaku syvyyteen yksi. Toisessa iteraatiossa kynnsarvo kasvatetaan 2:een, ja siinä tehdään syvyyshaku syvyyteen kaksi. Tätä jatketaan, kunnes solmu  $t$  löytyy tai koko hakuavaruus on tutkittu. Toisin kuin leveyshaku, DFID ei tallenna tutkittuja solmuja, vaan ne hylätään, ja haku aloitetaan joka kerta uudestaan solmusta  $s$  [Korf, 1985].

IDA\* toimii kuten DFID, mutta kynnsarvona käytetään syvyyden arvoa  $f(n) = g(n) + h(n)$ . Alussa kynnsarvo on solmun  $s$  saama arvo, ja tämän jälkeen kynnsarvoksi otetaan aina seuraavaksi pienin löydetty  $f(n)$ :n arvo, joka on suurempi kuin kynnsarvo. Haun alkaessa  $s$ :n arvo  $f(n)$  voisi olla vaikkapa 100, jolloin kynnsarvo olisi myös 100. Graafin haaraa tutkitaan, kunnes kohdetaan solmu  $n$ , jonka arvo  $f(n)$  on korkeampi kuin 100. Koko etsinnän uusiminen joka kerta tekee IDA\*:n toiminnasta A\*:ä hitaamman, mutta IDA\* käyttää huomattavasti vähemmän muistia kuin A\*. Tämä takaa, että IDA\* pystyy ratkaisemaan myös hyvin laajat ongelmat – kuten Rubi-



kin kuution – joissa  $A^*$  käyttää muistin loppuun ennen ratkaisun löytymistä. Solmujen uudelleen tutkiminen ei myöskään vaikuta ajoajan asymptoottiseen kasvuun. Kuten  $A^*$ , myös  $IDA^*$  löytää lyhimmän mahdollisen polun solmusta  $s$  solmuun  $t$ , jos polku on olemassa ja heuristinen arvio ei ole yliarvio. Paremman muistitehokkuuden lisäksi  $IDA^*$  on myös huomattavasti helpompi toteuttaa kuin  $A^*$  [Korf, 1985; DeLisle, 2008; Reinefeld and Marsland, 1994].

$IDA^*$ :n toimintaa on koitettu tehostaa monilla tavoilla, joista solmujen uudelleen järjestämiseen pohjautuvat tekniikat on todettu varsin tehottomiksi ja osittain voimakkaasti ratkaistavan tapauksen vaikuttamiksi. Paras tunnettu tapa tehostaa  $IDA^*$ :n toimintaa on käyttää hajautustauluna (hash table) toteutettua transpositiotaulua (transposition table), joka sisältää kaikki solmut, joissa on vierailtu, ja niihin johtavan polun pituuden. Jos solmuun saavutaan uudestaan niin, että uuden polun pituus on pienempi tai yhtäsuuri kuin aikaisemman polun pituus, voidaan solmuun johtava polku järjestää uudelleen [Reinefeld and Marsland, 1994].

Pelkkää transpositiotaulua nopeampi tekniikka on transpositiotaulun ja tunnetun pisimmän polun yhdistelmä. Tällöin, kun graafissa saavutaan solmuun joka löytyy jo transpositiotaulusta, niin että kynnsarvoa ei vielä ole ylitetty, kokeillaan transpositiotaulusta löytyvää, tunnettua pisintä polkua ensimmäiseksi. Vaikka tämä tapa on yleensä pelkkää transpositiotaulua nopeampi, voi se joissain tapauksissa johtaa jopa pelkkää  $IDA^*$ :ä hitaampaan toimintaan [Reinefeld and Marsland, 1994].

---

### **IDA\*-algoritmi**

---

```
root ← aloitussolmu
threshold ← root f()
suorita syvyyshaku aloittaen rootista
if goal ei löydetty
    threshold ← min f() > threshold
    toista syvyyshaku alkaen rootista
depth-first search(node):
    if node = goal
        return goal löydetty
    if node f() > threshold
        return goal ei löydetty
    else
        for each child node, while goal ei löydetty, depth-first search(child)
```

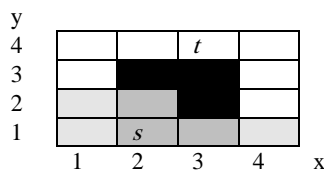
---

### **Algoritmi 2. IDA\* [DeLisle, 2008]**

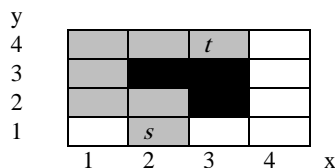
---

Seuraavaksi käydään läpi algoritmissa 2 esitetyn  $IDA^*$ :n esimerkkihaku. Heuristiikkana on käytetty Manhattanin etäisyyttä. Mustalla täytetyt ruudut ovat alueita, joilla ei voi liikkua,  $s$  osoittaa aloitussolmua,  $t$  lopetussolmua, vaaleampi harmaa solmua, jossa on vierailtu, ja tummempi harmaa solmua, jota on tarkasteltu.

Kynnysarvo:  $f(2,1) = 0 + (|2-3|+|1-4|) = 4$   
 $f(2,2) = 1 + (|2-3|+|2-4|) = 4$   
 $f(1,2) = 1 + (|1-3|+|2-4|) = 5$   
 $f(1,1) = 1 + (|1-3|+|1-4|) = 6$   
 $f(3,1) = 1 + (|3-3|+|1-4|) = 4$   
 $f(4,1) = 2 + (|4-3|+|1-4|) = 6$



Kynnysarvo : 5  
 $f(2,1) = 0 + (|2-3|+|1-4|) = 4$   
 $f(2,2) = 1 + (|2-3|+|2-4|) = 4$   
 $f(1,2) = 1 + (|1-3|+|2-4|) = 5$   
 $f(1,3) = 2 + (|1-3|+|3-4|) = 5$   
 $f(1,4) = 3 + (|1-3|+|4-4|) = 5$   
 $f(2,4) = 4 + (|2-3|+|4-4|) = 5$   
 $f(3,4) = 5 + (|3-3|+|4-4|) = 5$



#### 4. Reunahaku (Fringe Search)

Reunahakua voi toimintansa puolesta pitää IDA\*:<sup>n</sup> ja A\*:<sup>n</sup> välimuotona. Se ylläpitää kahta listaa, now ja later, jotka käytännössä vastaavat A\*:<sup>n</sup> OPEN-listaa. Now-lista sisältää tässä iteraatioissa mukana olevat solmut ja later-lista seuraavassa iteraatioissa käsiteltävät solmut. Kummatkin näistä listoista ovat järjestämättömiä, eikä käsittelyyn otettavien solmujen järjestyksellä ole väliä, jolloin lisäys- ja poisto-operaatiot suoritetaan ajassa  $O(1)$ . Reunahaku etenee IDA\*:<sup>n</sup> tavoin iteraatioissa, ja riippuen solmujen lisäysjärjestyksestä now-listaan, se voi myös tarkastella solmuja täsmälleen samassa järjestyksessä kuin IDA\*. Later-listoja voi olla useampiakin kuin yksi, mikä johtaa solmujen osittaiseen järjestämiseen. Reunahaku voidaan saada tarkastelemaan solmuja täsmälleen samassa järjestyksessä kuin A\*. Tällöin tarvittava järjestäminen saadaan aikaiseksi tekemällä jokaiselle solmulle oma later-lista [Björnsson *et al.*, 2005; DeLisle, 2008].

Reunahaun alussa now-lista sisältää solmun  $s$  ja later-lista on tyhjä. Jokaisessa iteraatioissa tarkastellaan ensin now-listan ensimmäistä solmua. Jos sen  $f(n)$  arvo on suurempi kuin kynnysarvo, poistetaan se now-listasta ja lisätään later-listaan. Jos  $f(n)$  on pienempi tai yhtäsuuri kuin kynnysarvo, lisätään solmun lapset now-listaan, ja poistetaan itse solmu now-listasta. Tätä jatketaan, kunnes ei enää löydy uusia solmuja, joiden  $f(n)$ :n arvo on pienempi tai yhtä suuri kuin kynnysarvo, tai solmu  $t$  löytyy. Jos solmua  $t$  ei löytynyt, siirretään later-listan sisältö now-listaan ja tyhjennetään later-lista. Uusi kynnysarvo määritetään kuten IDA\*:<sup>n</sup>:ssä. Tätä jatketaan, kunnes solmu  $t$  löytyy tai koko hakualue on tutkittu [Björnsson *et al.*, 2005; DeLisle, 2008].

Reunahaulla, IDA\*:<sup>n</sup>:lla ja A\*:<sup>n</sup>:llä tehdyissä testeissä reunahaku oli 25-40% A\*:<sup>n</sup>:ä nopeampi, ja yhä 10-20% vahvasti optimoitu A\*:<sup>n</sup>:ä nopeampi. A\* taas oli IDA\*:<sup>n</sup>:ä nopeampi. Björnssonin ja muiden [2005] mukaan reunahaku on A\*:<sup>n</sup>:ä nopeampi, koska sen, toisin kuin A\*:<sup>n</sup>:n, ei tarvitse ylläpitää järjestettyä OPEN-listaa. A\* olikin toteutettu käyttäen järjestettyä binääripuuta. Tämä toteutus ei kuitenkaan ota huomioon, että A\*:<sup>n</sup> OPEN-lista on myös mahdollista toteuttaa

suurelta osin järjestämättömänä käyttämällä hyväksi aikaisemmin esitettyä halpaa listaa, joka on oletettavasti järjestettyä OPEN-listan toteutusta nopeampi. A\*:n ja reunahaun nopeuserot tällä toteutuksella jäävät epäselväksi. Jos OPEN-lista halutaan toteuttaa järjestettynä, olisi Fibonaccin keko järjestettyä binääripuuta nopeampi ratkaisu [Fredman and Tarjan, 1987].

---

### Reunahaku-algoritmi

---

```

Initialize:
  Fringe F ← (s)
  Cache C[start] ← (0, null),
  C[n] ← null jos n ≠ start
  flimit ← h(start)
  found ← false
Repeat until found = true tai F tyhjä
  fmin ← ∞
  Iteroi solmut n ∈ F vasemmalta oikealle:
    (g, parent) ← C[n]
    f ← g + h(n)
    if f > flimit
      fmin ← min(f, fmin)
      continue
    if n = goal
      found ← true
      break
  Iteroi s ∈ successors(n) oikealta vasemmalle:
    gs ← g + cost(n, s)
    if C[s] ≠ null
      (g', parent) ← C[s]
      if gs ≥ g'
        continue
    if F sisältää s
      Poista s F:stä
    Lisää s F:ään n:n jälkeen
    C[s] ← (gs, n)
  Poista n F:stä
  flimit ← fmin
if found = true
  Rakenna polku välimuistin vanhempisolmuista

```

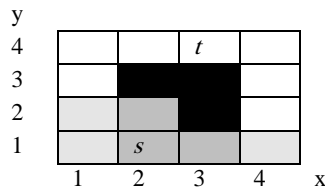
---

### Algoritmi 3. Reunahaku [Björnsson *et al.*, 2005]

---

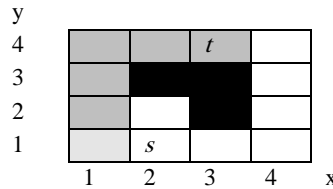
Käydään vielä läpi esimerkkihaku algoritmissa 3 esitetyllä reunahaku-algorimilla. Heuristiikana on käytetty Manhattanin etäisyyttä. Mustalla täytetyt ruudut ovat alueita, joilla ei voi liikkua,  $s$  osoittaa aloitussolmua,  $t$  lopetussolmua, vaaleampi harmaa solmua, jossa on vierailtu, ja tummempi harmaa solmua, jota on tarkasteltu.

Kynnysarvo:  $f(2,1) = 0 + (|2-3|+|1-4|) = 4$   
 $f(2,2) = 1 + (|2-3|+|2-4|) = 4$   
 $f(3,1) = 1 + (|3-3|+|1-4|) = 4$   
 $f(1,2) = 1 + (|1-3|+|2-4|) = 5$   
 $f(1,1) = 1 + (|1-3|+|1-4|) = 6$   
 $f(4,1) = 2 + (|4-3|+|1-4|) = 6$



now	later
(3,1)	(1,1)
(2,2)	(1,2)
(2,1)	(4,1)

Kynnysarvo: 5  
 $f(1,1) = 1 + (|1-3|+|1-4|) = 6$   
 $f(1,2) = 1 + (|1-3|+|2-4|) = 5$   
 $f(1,3) = 2 + (|1-3|+|3-4|) = 5$   
 $f(1,4) = 3 + (|1-3|+|4-4|) = 5$   
 $f(2,4) = 4 + (|2-3|+|4-4|) = 5$   
 $f(3,4) = 5 + (|3-3|+|4-4|) = 5$



now	later
(3,4)	(1,1)
(2,4)	
(1,4)	
(1,3)	
(1,2)	
(4,1)	

## 5. Optimointi

### 5.1 Nopeuden ja tehokkuuden optimointi

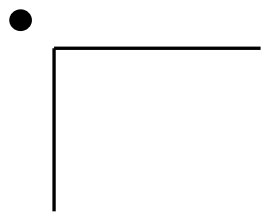
Koska polunetsintäalgoritmit kuluttavat varsin paljon resursseja, paras mahdollinen nopeusoptimointi mille tahansa polunetsintäalgoritmilta on olla käyttämättä sitä ollenkaan. Tästä syystä ennen polunetsintään aloittamista on hyödyllistä varmistaa kaksi seikkaa: onko määriteltynyt päätepisteeseen ylipäätään mahdollista päästä – liikutettava hahmo saattaa esimerkiksi olla saarella ja päätepiste mantereella? Tällöin validia polkua ei ole olemassa, eikä polunetsintäalgoritmia siis välttämättä tarvita. Polun hakeminen saaren reunalle on tietysti mahdollista. Jos polku voi olla olemassa, on täysin mahdollista, että suora sokkohaku päätepisteeseen suuntaan ei törmää esteisiin. Vasta näiden tarkistusten jälkeen siirrytään käyttämään itse polunetsintäalgoritmia [Rabin, 2000b].

Haun tehokkuudessa ei ole kyse vain käytettävän hakualgoritmin optimoinnista, vaan siihen vaikuttaa myös pelialueen esittämiseen käytetty rakenne. Ruudukko (grid) – neli- tai kuusikulmio – on yleisin 2D-karttojen esittämiseen käytetty tekniikka. Esteiden lisääminen ruudukkoon on helppoa, mutta se saattaa helposti kasvaa suureksi. Esim. 1000 x 1000 -ruudukossa on miljoona mahdollista ruutua. Ruudukko ei sovellu hyvin 3D-karttojen esittämiseen [Rabin, 2000b; Stout 1999].

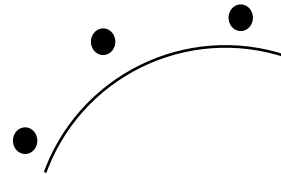
**Varsinainen monikulmainen lattia** (actual polygonal floor) käyttää hakualueena 3D-kartan merkittyä lattiaa. Tästä syystä hakualue on juuri niin monimutkainen kuin määritelty lattiakin on. Koska hakualue on itse lattia, ei hakualueella olevia esteitä voida esittää, vaan ne pitää havaita erillisellä algoritmilla [Rabin, 2000b].

**Monikulmainen lattian esitys** (polygonal floor representation) on manuaalisesti toteutettu, yksinomaan polunetsintää varten luotu hakualueen esitys, josta on voitu poistaa alueet, joilla pelihahmot eivät voi kulkea. Tämä johtaa pieneen hakualueen esitykseen, mutta pelihahmoja ei voida esittää hakualueella. Koska pelihahmoja ei voida esittää hakualueella, tarvitaan polun valinnassa erillistä algoritmia [Rabin, 2000b].

**Näkyvyyspisteet** (points of visibility) on manuaalisesti tai algoritmilla luotu minimaalinen esitys pelialueesta. Pisteet toimivat graafin solmuina ja muodostavat käytettävän hakugraafin. Pisteet asetetaan tarpeeksi etäälle kulmista, jotta pelihahmot eivät törmää kulmiin. Kulma-asettelunsa takia pelialueet, joilla on paljon pyöreitä kohteita, voivat johtaa monimutkaiseen esitykseen. Kuvissa 4 ja 5 on havainnollistettu näkyvyyspisteitä terävässä ja kaarevassa kulmassa.



Kuva 4. Näkyvyyspiste terävässä kulmassa



Kuva 5. Näkyvyyspisteitä kaarevassa kulmassa

Pelihahmo voi kulkea pisteestä toiseen, jos niiden välillä on suora näkyvyys, mikä johtaa täysin suoriin polkuihin. Esitykseen voidaan lisätä esteitä, mutta niitä ei voida enää poistaa, minkä takia tekniikka ei toimi erityisen hyvin dynaamisissa kartoissa. Pelihahmojen esittäminen ei ole mahdollista tällä tekniikalla [Rabin, 2000b; Stout 1999].

Tavallisesti jokainen hakualueen ruutu tai monikulmio sisältää polunetsinnässä käytettävää tietoa, mikä johtaa jatkuvaan valtavaan muistin varaukseen. Irrottamalla hakualueen esitys ja polunetsinnässä käytettävä tieto toisistaan tämä muisti voidaan vapauttaa. Tämä muutos mahdollistaa samanaikaiset haut, mutta samanaikaisten hakujen toteuttaminen ei ole yleisesti kannattavaa. Koska koko kartalle ei enää ole solmuja, tarvitsee uudet solmut joko luoda aina tarvittaessa tai vaihtoehtoisesti luoda valmiiksi tarpeelliseksi katsottu määrä solmuja, joita voidaan käyttää aina uudestaan. Jos ennalta luotu määrä ei riitä, voidaan lisää solmuja luoda tarpeen mukaan. Näitä solmuja voidaan varastoida hajautustaulussa [Rabin, 2000b; Stout 1999].

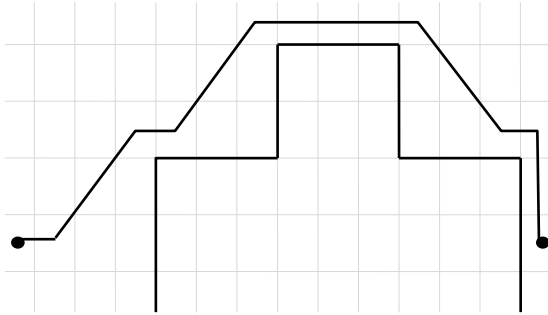
Hakuongelmaa yksinkertaistamalla on mahdollista helpottaa suoritettavaa hakua huomattavasti. Yksittäisistä ruudukon ruuduista voidaan esimerkiksi muodostaa suurempia  $n \times m$  -alueita. Tätä tekniikkaa kutsutaan hierarkkiseksi polunetsinnäksi (hierarchical pathfinding), ja sitä voidaan soveltaa mille tahansa pelialueelle. Talo voidaan esimerkiksi jakaa huoneisiin, joiden poistumiskohtia voivat olla ovet, ikkunat tai kenties reiät seinissä. Alueilla, joilla ei ole selkeää poistumiskohtaa, voidaan poistumiskohtana käyttää alueen koko reunaan. Hierarkkisen polunetsinnän

periaatteena on ensin ratkaista korkean tason ongelmana esimerkiksi polku keittiöstä makuuhuoneeseen ja tämän jälkeen yksittäisten huoneiden sisällä kuljettavat polut [Rabin, 2000b]. Karttaa ei ole välttämätöntä jakaa vain kahteen abstraktiotasoon, vaan abstraktiotasoja voi periaatteessa olla loputtomasti. Ainoana rajana ovat uusien kaarien muistivaatimukset ja terve järki [Botea *et al.*, 2004]. Käytettävät hakualueet voidaan joko määritellä käsin tai pelialue on voitu rakentaa alusta lähtien hakuun sopivista palikoista, jolloin haettavat korkean tason polut ovat polkuja palikoiden välillä ja matalan tason polut reittejä palikoiden sisällä [Rabin, 2000b]. Ruudukoilta on olemassa HPA\*-algoritmi (Hierarchical Path-Finding A\*), joka jakaa kartan automaattisesti sopiviin osiin ja suorittaa polunetsinnän kaikilla abstraktiotasoilla [Botea *et al.*, 2004]. Hierarkkisessa polunetsinnässä polku haetaan, toteutustavasta riippuen, joko alueen sisään-tulon keskeltä ulosmenon keskellä, sisään-tulon jommasta kummasta laidasta ulosmenon jompaan kumpaan laitaa tai näiden kahden vaihtoehdon yhdistelmänä. Ulosmenokohtien rajallisuudesta johtuen hierarkkinen polunetsintä ei välttämättä tuota optimaalisia polkuja [Rabin, 2000b; Botea *et al.*, 2004].

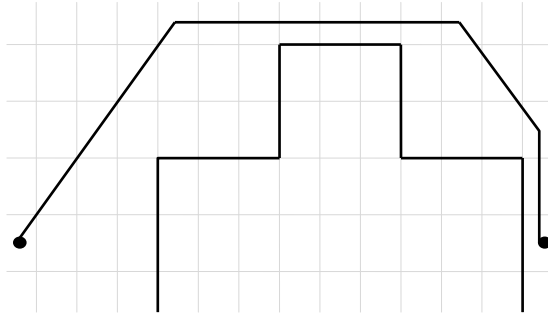
Hakuongelman jakamisella aliongelmiin on myös se hyöty, että koko polkua solmusta  $s$  solmuun  $t$  ei tarvitse etsiä turhaan. Koko polun hakeminen on tarpeetonta, koska dynaamisissa kartoissa ja kartoilla, joissa on muita yksiköitä, kokonaisuudessaan haettu polku saattaa katketa muutaman askeleen jälkeen toiseen yksikköön tai pommitettuun siltaan. Tällaisessa tapauksessa kartan toiselle puolelle haetusta polusta suurin osa menee hukkaan, ja resursseja on tuhlatu huomattavasti. Kun kartta on jaettu hierarkioihin, vain nykyinen ja seuraava aliongelma tarvitsee ratkaista uudestaan. Seuraava aliongelma tarvitsee ratkaista, jotta yksikön liike ei pysähdy hetkeksi jokaisen aluevaihdon välillä, kun uutta polkua haetaan [Rabin, 2000b; Botea *et al.*, 2004].

## 5.2 Polun esteettisyyden optimointi

Polunetsintäalgoritmien tuottamat optimaaliset tai lähes optimaaliset polut eivät yleensä ole paras valinta käytettäväksi poluksi, koska ne eivät vaikuta luonnolliselta pelaajalle. Käytetyn polun kokenemaisuus vähentää pelin realistisuuden tuntua ja tuottaa huonomman kokemuksen. Algoritmien tuottamissa poluissa on kolmenlaisia ongelmia, jotka ovat suoruuden puuttuminen poluista, suoruuden puuttuminen hierarkioita käytettäessä ja terävät käännökset poluissa [Rabin, 2000a].



Kuva 6. Suoristamaton polku

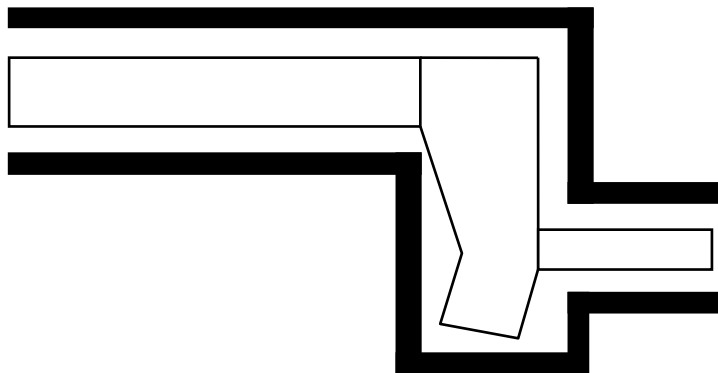


Kuva 7. Suoristettu polku

Polunetsintäalgoritmin tuottamat polut saattavat sisältää useita tarpeettomia käännöksiä matkallaan solmuun  $t$ , kuten kuvassa 6. Kuvissa 6 ja 7 esitetyt polut ovat kummatkin yhtä pitkiä, mutta kuvan 6 polku etenee kohti solmua  $t$  välittämättä selkeistä edessä olevista esteistä. Kuvan 7 polku sen sijaan on huomattavasti suoraviivaisempi ja uskottavampi. Käsiteltävän polun uskottavuus riippuu tietysti sovellusympäristöstä. Esimerkiksi tulisuojaa hakeva sotilas valitsisi luultavasti näistä kahdesta polusta kuvassa 6 esitetyn polun, mutta pakeneva vanki luultavasti valitsisi kuvan 7 polun.

Suoruuden aikaansaaminen voidaan toteuttaa kahdella tavalla. Käytetty tapa riippuu pelialueen toteuttamiseen käytetystä rakenteesta. Käytettäessä rakenteita, joiden koko on aina sama, kuten ruudukko, suoruus saadaan aikaiseksi kannustamalla polunetsintäalgoritmia valitsemaan suora polku. Tämä saavutetaan lisäämällä kaikkien sellaisten askelten pituutta, jotka eivät ole suorassa linjassa edellisen askeleen kanssa. Tämä metodi hidastaa polun löytymistä, koska enää ei etsitä vain lyhintä vaan myös suorinta polkua pisteestä  $s$  pisteeseen  $t$  [Rabin, 2000a].

3D-kartoissa suoruus voidaan saada aikaiseksi toteuttamalla kartta navigaatioverkkona (Navigation Mesh). Navigaatioverkko on tekniikka, jossa alue, jolla kulkeminen on mahdollista, määritellään erimuotoisilla ja kokoisilla monikulmioilla, jotka vastaavat tavallaan ruudukon ruutuja. Kuvassa 8 on esimerkki navigaatioverkosta käytävässä.

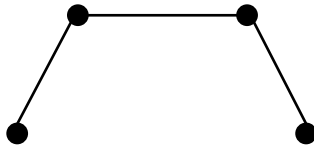


Kuva 8. Navigaatioverkko käytävässä

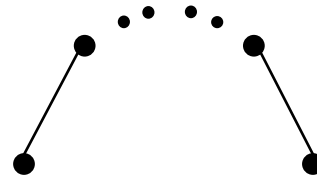
Polku monikulmiosta toiseen haetaan monikulmion reunan keskeltä seuraavan monikulmion reunan keskelle, mikä johtaa poukkoileviin polkuihin. Saatu polku voidaan suoristaa hakemalla saadusta polusta viimeinen sellainen monikulmio, johon tästä monikulmiosta on suora näköyhteys, ja kulkemalla suoraan sinne. Tällä tavalla saatavat polut ovat suoria [Snook, 2000; Tozour, 2002].

Hierarkioita käytettäessä saadut polut voivat aina alkaa alueelle määritellyn sisäänkäynnin keskeltä ja päättyä ulosmenon keskelle, mikä ei selvästikään vaikuta luonnolliselta eikä myöskään tuota lyhintä reittiä. Luonnollisempi lopputulos saadaan aikaiseksi hakemalla polku aina seuraavan alueen ulosmenoon. Näin poluista muodostuu järkevempiä, mutta niiden eteen joudutaan myös tekemään ylimääräistä työtä. Kun polku haetaan seuraavan alueen ulosmenoon, jälkimmäinen osa polusta hylätään ennen seuraavan aliongelman ratkaisua [Rabin, 2000a].

Polunetsintäalgoritmien tuottamat polut sisältävät usein varsin luonnottomilta vaikuttavia, teräviä käännoiksi. Nämä käännoiksi saadaan muutettua huomattavasti luonnollisemmalla vaikuttavaan, kaarevampaan muotoon käyttämällä Catmullin-Romin spliniä [Rabin, 2000a]. (ks. kuvat 9 ja 10).



Kuva 9. Polku ennen Catmullin-Romin spliniä



Kuva 10. Pisteiden 2 ja 3 väliin Catmullin-Romin splinillä muodostettu kaari

Catmull-Rom on kardinaalisplinin (cardinal spline) erikoistapaus, ja se esitetään matriisina muodossa

$$B = C^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -s & 0 & s & 0 \\ 2s & s-3 & 3-2s & -s \\ -s & 2-s & s-2 & s \end{bmatrix},$$

missä  $s$  on kaaren jännitys. Arvoksi  $s$  asetetaan  $(1-t)/2$ . Catmullin-Romin splinillä on  $t=0$ , ja kardinaalisplinillä  $t$  on jokin arvo arvojoukosta  $[0,1)$ . Kaaren jännitys vaikuttaa sen jyrkkyyteen. Matriisia  $C$  kutsutaan rajoitematriisiksi (constraint matrix), ja se saadaan ratkaisemalla kaavat

$$f(0) = p_2,$$

$$f(1) = p_3,$$

$$f'(0) = \frac{1}{2}(1-t)(p_3 - p_1) \text{ ja}$$

$$f'(1) = \frac{1}{2}(1-t)(p_4 - p_2),$$



missä  $p_i$  :t ovat vektorin kontrollipisteitä. Edellä  $f(0)$  määrittelee kaaren alkamispisteen,  $f(1)$  kaaren loppumispisteen,  $f'(0)$  kaaren alun derivaatan ja  $f'(1)$  kaaren lopun derivaatan. Matriisia B kutsutaan kantamatriisiksi (basis matrix), ja se on matriisin  $C$  käänteismatriisi [Shirley *et al.*, 2009].

Catmullin-Romin menetelmä vaatii toimiakseen neljä pistettä, ja sen tuottama uusi reitti kulkee annettujen pisteiden kautta. Catmull-Rom palauttaa kaaren saatujen toisen ja kolmannen pisteen välistä. Catmull-Romin tuottama uusi polku sisältää huomattavasti suuremman määrän pisteitä alkuperäiseen polkuun verrattuna. Tätä määrää voidaan ja kannattaa pienentää soveltamalla polkuun algoritmia, joka poistaa samassa linjassa olevia turhia pisteitä [Rabin, 2000a].

Polun ensimmäisen ja toisen pisteen välinen kaari saadaan antamalla ensimmäinen piste kahteen kertaan Catmull-Romille. Sama periaate pätee viimeisiin pisteisiin [Rabin, 2000a].

## 6. Lopuksi

Ottaen huomioon, kuinka nopeasti uudet ja hyväksi todetut ideat leviävät yleiseen käyttöön peliteollisuudessa, voi reunahaun odottaa yleistyvän nopeastikin, jos se todellakin on A\*<sup>2</sup>:ä huomattavasti tehokkaampi. Muistitehokkuutensa ansiosta IDA\*<sup>2</sup>:n voi olettaa olevan suosittu peleissä, jotka on toteutettu laitteille, joissa muistia on vähän, kuten mobiililaitteille ja vähäisemmässä määrin konsoleille.

Tämän rajoitetun tarkastelun ulkopuolelle on jäänyt monia polunetsinnässä käytettäviä tekniikoita. Aiheesta syvempää ymmärrystä hakeville voidaankin suositella tutustumista paikalliseen ohjaamiseen (local steering) (esimerkiksi [Tomlinson, 2004]), attraktoreihin (attractors) ja repulsoreihin (repulsors) (esimerkiksi [Olsen, 2004]), parveiluun (flocking) (esimerkiksi [Reynolds, 1987]) sekä joukkoihin (swarms) (esimerkiksi [Scutt, 2002]).

## Viiteluettelo

[Björnsson *et al.*, 2005] Yngvi Björnsson, Markus Enzenberger, Robert C. Holte, and Jonathan Schaeffer, Fringe search: Beating A\* at pathfinding on game maps. In: *IEEE Computational Intelligence and Games '05*, (2005), 125–132.

[Botea *et al.*, 2004] Adi Botea, Martin Müller, Jonathan Schaeffer, Near optimal hierarchical path-finding. *J. Game Development* 1 (2004), 7–28.

[Chatfield, 2009] Tom Chatfield, Videogames now outperform Hollywood movies.

<http://www.guardian.co.uk/technology/gamesblog/2009/sep/27/videogames-hollywood>. 2009.

Checked 3.5.2010.

[Dechter and Pearl, 1985] Rina Dechter and Judea Pearl, Generalized best-first search strategies and the optimality of A\*. *J. ACM* **32** (1985), 505-536.

[DeLisle, 2008] Robert Kirk DeLisle, Beyond A\*: IDA\* and fringe search. In: Scott Jacobs (ed.), *Game Programming Gems 7*. Charles River Media, 2008, 289-294.

[Fredman and Tarjan, 1987] Michael L. Fredman and Robert Endre Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM* **34** (1987), 596-615.

[Hart *et al.*, 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* **4** (1968), 100–107.

[Higgins, 2002] Dan Higgins, How to achieve lightning-fast A\*. In: Steve Rabin (ed.), *AI Game Programming Wisdom*. Charles River Media, 2002, 133-145.

[Jönsson, 1997] F. Markus Jönsson, An optimal pathfinder for vehicles in real-world digital terrain maps. <http://www.student.nada.kth.se/~f93-maj/pathfinder/>. 1997. Checked 13.5.2010.

[Korf, 1985] Richard E. Korf, Depth-first iterative deepening: an optimal admissible tree search. *Artificial Intelligence* **27** (1985), 97-109.

[Matthews, 2002] James Matthews, Basic A\* pathfinding made simple. In: Steve Rabin (ed.), *AI Game Programming Wisdom*. Charles River Media, 2002, 105-113.

[Nilsson, 1980] Nils J. Nilsson, *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers, 1980.

[Olsen, 2004] John M. Olsen, Attractors and repulsors. In: Andrew Kirmse(ed.), *Game Programming Gems 4*. Charles River Media, 2004, 355-364.

[Patel, 2009] Amit Patel, Amit's A\* pages. <http://theory.stanford.edu/~amitp/GameProgramming/>. 2009. Checked 17.12.2009.

[Rabin, 2000a] Steve Rabin, A\* aesthetic optimizations. In: Mark DeLoura (ed.), *Game Programming Gems*. Charles River Media, 2000, 264-271.

[Rabin, 2000b] Steve Rabin, A\* speed optimizations. In: Mark DeLoura (ed.), *Game Programming Gems*. Charles River Media, 2000, 272-287.

[Reinefeld and Marsland, 1994] Alexander Reinefeld and T. Anthony Marsland, Enhanced iterative-deepening search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16** (1994), 701-710.

[Reynolds, 1987] Craig W. Reynolds, Flocks, herds and schools: a distributed behavioral model. In: *SIGGRAPH'87*, (1987), 25-34.

[Russell and Norvig, 2010] Stuart Russell and Peter Norvig, *Artificial Intelligence: a Modern Approach*. Prentice Hall, 2010.

[Scutt, 2002] Tom Scutt, Simple swarms as an alternative to flocking. In: Steve Rabin (ed.), *AI Game Programming Wisdom*. Charles River Media, 2002, 202-208.

[Shirley *et al.*, 2009] Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen R. Marschner, Erik Reinhard, Kelvin Sung, William B. Thompson and Peter Willemsen, *Fundamentals of Computer Graphics*. A K Peters, 2009.

[Snook, 2000] Greg Snook, Simplified 3d movement and pathfinding using navigation meshes. In: Mark DeLoura (ed.), *Game Programming Gems*. Charles River Media, 2000, 288-304.

[Stout, 1999] Bryan Stout, Smart moves: Intelligent path-finding.  
[http://www.gamasutra.com/features/19990212/sm\\_01.htm](http://www.gamasutra.com/features/19990212/sm_01.htm). 1999. Checked 25.9.2009.

[Stout, 2000] Bryan Stout, The basics of A\* for path planning. In: Mark DeLoura (ed.), *Game Programming Gems*. Charles River Media, 2000, 254-263.

[Tomlinson, 2004] Simon L. Tomlinson, The long and short of steering in computer games. *I. J. of Simulation* **5** (2004), 33-47.

[Tozour, 2002] Paul Tozour, Building a near-optimal navigation mesh. In: Steve Rabin (ed.), *AI Game Programming Wisdom*. Charles River Media, 2002, 133-145.

# Suosittelujärjestelmät

## Markus Sairanen

### Tiivistelmä

Suosittelu sosiaalisena prosessina on tärkeää monille käyttäjille. On hankalaa ja ehkä jopa turhaa, että käyttäjä itsenäisesti tutustuu kaikkiin mahdollisiin vaihtoehtoihin. Suosittelujärjestelmät kehitettiin automatisoimaan suositteluprosessia. Suosittelujärjestelmiä käytetään esimerkiksi verkkokaupoissa, joista tunnetuin on Amazon.com. Nämä kaupalliset järjestelmät suosittelevat tuotteita potentiaalisille asiakkaille. Suosittelevat perustuvat aikaisempiin tapahtumiin ja käyttäjiltä saatuun palautteeseen. Tässä tutkielmassa esitellään suosittelujärjestelmien eri variaatiot ja niiden vahvuudet ja heikkoudet. Lisäksi mietitään, miten suosittelujärjestelmiä voitaisiin parantaa.

**Avainsanat ja -sanonnat:** Suosittelujärjestelmät, sisältöpohjainen suosittelu, yhteistoiminnallinen suosittelu, demografiapohjainen suosittelu, hyödyllisyyspohjainen suosittelu, tietämuspohjainen suosittelu, hybridisuosittelujärjestelmät.

**CR-luokat:** H.2.8, H.3.3

## 1. Johdanto

Käyttäjät joutuvat jatkuvasti tilanteisiin, joissa heillä on monia vaihtoehtoja valittavanaan, ja he tarvitsevat apua tutustuakseen vaihtoehtoihin tai supistaakseen niiden joukkoa. Esimerkiksi Internetin hakukoneet löytävät yleensä tuhansia asiaankuuluvia sivuja ja joka vuosi lehdissä ja konferenssijulkaisuissa julkaistaan tuhansia tutkimustuloksia mahdollisesti kiinnostavista aiheista [Pazzani, 2004]. Sen lisäksi valitseminen sadoista ravintoloista, elokuvista tai TV-sarjoista saattaa tuottaa monille ongelmia. Ennen nykyistä informaatiotulvaa käyttäjä saattoi kysyä suosituksia tuttaviltaan. Tilanne on kuitenkin muuttunut, koska valinnan vapaus on kasvanut ja tarjolla on liikaa vaihtoehtoja. Lisäksi on niitäkin käyttäjiä, jotka eivät tunne ketään, jolla olisi samanlainen maku asioiden suhteen. Tämä ongelma korostaa tarvetta tiedon erottamiseen pystyville järjestelmille, jotka pystyvät suodattamaan huomaamatonta tietoa ja ennustamaan, pitäisikö käyttäjä jostakin asiasta [Ghazanfar and Prugel-Bennett, 2010]. Tällaisia järjestelmiä kutsutaan suosittelujärjestelmiksi ja ne automatisoivat suositteluprosessin.

Suosittelujärjestelmät ovat tulleet tärkeäksi tutkimusalueeksi 1990-luvun puolivälissä julkaistujen ensimmäisten tutkimusartikkelien jälkeen [Adomavi-

cius and Tuzhilin, 2005]. Viime vuosikymmenellä on sekä yliopistomaailmassa että teollisuuden parissa tehty paljon töitä kehitettäessä uusia lähestymistapoja suosittelujärjestelmiin. Kiinnostus alalla on yhä suurta, koska se muodostaa haastavan tutkimusalueen ja koska sillä on runsaasti käytännöllisiä sovellutuksia, jotka auttavat käyttäjiä käsittelemään informaation tulvaa ja tuottavat heille personoituja suosituksia, sisältöä tai palveluja. Esimerkkinä edellä mainituista sovelluksista on kirjojen, CD-levyjen ja muiden tuotteiden suosittelu Amazon.com-verkkokaupassa. Vaikka suosittelujärjestelmien juuret juontavat useista muista tutkimusalueista kuten kognitiivisista tieteistä ja tiedonhausta, suosittelujärjestelmät tulivat omaksi itsenäiseksi tutkimusalueeksi 1990-luvun puolivälissä tutkijoiden keskittyessä suositteluongelmiin, jotka selkeästi ovat riippuvaisia arvioinnista [Adomavicius and Tuzhilin, 2005]. Yleisimmässä muodossaan suositteluongelma voidaan tiivistää ongelmaksi siitä, miten käyttäjä arvioi asian, johon hän ei ole vielä tutustunut. Normaalisti tämä arviointi perustuu käyttäjän aikaisemmin luomiin arvosteluihin muista asioista. Kun voimme arvioida arvostelut vielä arvostelemattomille asioille, voimme suositella käyttäjälle asioita, jotka saivat arvioinnissa parhaimmat arvostelut.

Muodollisemmin ongelma voidaan kuvata seuraavasti. Olkoon  $C$  kaikkien käyttäjien joukko ja  $S$  kaikkien asioiden joukko, joita voidaan suositella, kuten kirjoja, elokuvia, musiikkia tai ravintoloita. Molemmat joukot voivat kasvaa tuhansiin tai jopa miljooniin alkioihin. Olkoon  $u$  hyödyllisyysfunktio, joka mittaa asian  $s$  hyödyllisyyttä käyttäjälle  $c$  eli  $u: C \times S \rightarrow \mathbb{R}$ , kun  $\mathbb{R}$  on järjestetty joukko ei-negatiivisia kokonaislukuja tai reaalilukuja määrättyä väliltä. Jokaiselle käyttäjälle  $c \in C$  haluamme valita asioita  $s' \in S$ , jotka maksimoivat käyttäjän hyödyn. Formaalisemmin:

$$\forall c \in C, s'_c = \arg \max_{s \in S} u(c, s). \quad (1)$$

Suosittelujärjestelmissä asian hyödyllisyys esitetään usein arviona, joka ilmaisee, kuinka paljon tietty käyttäjä piti tietystä asiasta. Esimerkiksi Mikko Mallikas antoi elokuvalla "Ihmemaailma Oz" luokituksen 7 (maksimi 10). On kuitenkin otettava huomioon, että hyödyllisyys voi olla molemminpuolista. Tilanteen mukaan hyödyllisyyden  $u$  voi määrittellä käyttäjä, kuten usein on käyttäjän määrittelemissä arvosteluissa, tai se voidaan laskea sovelluksessa, kuten tehdään myyntivoitto-perusteisissa hyödyllisyysfunktioissa [Adomavicius and Tuzhilin, 2005].

Jokainen alkio joukossa  $C$  voidaan määrittellä profiilina, joka sisältää joitakin tunnusomaisia piirteitä kuten ikä, sukupuoli, siviilisääty tai tulot. Yksinkertaisimmassa tapauksessa profiili voi sisältää vain yksittäisen yksilöllisen elementin kuten käyttäjätunnuksen. Samoin jokainen alkio joukossa  $S$  on määritelty joukolla ominaisuuksia. Esimerkiksi elokuvien suosittelujärjestelmässä jokais-

ta elokuvasta voi olla esitettynä paitsi yksilöllinen tunnus myös nimi, tyyllilaji, ohjaaja, julkaisuvuosi, päänäyttelijät ja niin edelleen.

Keskeinen ongelma suosittelujärjestelmissä on se, että hyödyllisyys  $u$  ei yleensä ole määritelty koko karteesiselle tulolle  $C \times S$ , vaan ainoastaan sen osajoukolle. Suosittelujärjestelmissä hyödyllisyys esitetään tavallisesti arvosteluina, ja se on aluksi määritelty vain asioille, jotka käyttäjät ovat arvostelleet aiemmin. Esimerkiksi elokuvien suosittelujärjestelmässä käyttäjät aluksi arvostelevat osan näkemistään elokuvista. Sen jälkeen suosittelumoottorin on mahdollista laskea (ennakoida) arvio arvostelemattomille elokuva-käyttäjä -pareille ja näyttää sopivat suosittelut näiden ennakointien perusteella.

Ekstrapolointi tuntemattomista tunnettuihin arvioihin tehdään yleensä

1. nimeämällä heuristiikka, joka määrittää hyödyllisyysfunktion sekä vahvistamalla empiirisesti sen toiminta ja
2. arvioimalla hyödyllisyysfunktio, joka optimoi tietyt suorituskriteerit kuten keskineliövirheen.

Kun on tehty arviointi tuntemattomille arvosteluille, voidaan tehdä varsinaiset suositukset käyttäjälle valitsemalla ensimmäisessä vaiheessa hänen tapuksessaan korkeimmat arvostelut saaneet asiat.

Kaikesta kehityksestä huolimatta nykyinen suosittelujärjestelmien sukupolvi tarvitsee lisäkehitystä tehdäkseen suosittelumenetelmistä tehokkaampia ja soveltuvampia laajemmalle alueelle tosielämän sovelluksia kuten esimerkiksi lomien suosittelu. Nämä parannukset käsittävät paremmat menetelmät käyttäjien käyttäytymisen ja suositeltavien tuotteiden kuvaamiseen, asiayhteyksien sisällyttämisen suositteluprosessiin, monesta muuttujasta riippuvien arviointien käyttämisen ja vähemmän tungettelevien ja paljon mukautuvampien suosittelumetodien kehittämisen, jotka kuitenkin tehokkaammin mittaavat suosittelujärjestelmien tehokkuutta.

Tutkielman eteneminen on seuraavanlainen. Luvussa kaksi esittelen eri muunnelmat suosittelujärjestelmistä sekä niiden vahvuudet ja heikkoudet. Kolmannessa luvussa kuvaan, miten suosittelujärjestelmiä voitaisiin laajentaa toimimaan paremmin. Lopuksi neljännessä luvussa tehdään yhteenveto tutkielmassa käsitellyistä asioista.

## 2. Suosittelujärjestelmät

### 2.1. Sisältöpohjainen suosittelu

Sisältöpohjaisessa suosittelussa asian  $s$  hyödyllisyys  $u(c,s)$  käyttäjälle  $c$  arvioidaan käyttäjän  $c$  asian  $s$  kanssa samantapaisille asioille aikaisemmin tekemien hyötyarvioiden  $u(c,s_i)$  perusteella. Esimerkiksi suositellakseen elokuvaa käyttäjälle  $c$  elokuvien suosittelujärjestelmä yrittää ymmärtää yhteiset ominaisuudet, kuten näyttelijät, ohjaajat, tyyllilajit tai aihepiirit, elokuvista, jotka käyttäjä on

arvostellut korkealle aikaisemmin. Sen jälkeen suositellaan vain elokuvia, jotka ovat samankaltaisia käyttäjän aikaisempien mieltymysten kanssa.

Sisältöpohjaisten suosittelujärjestelmien taustalla käytetään käyttäjäprofiileja, jotka sisältävät tietoa käyttäjän mausta, mieltymyksistä ja vaatimuksista [Adomavicius and Tuzhilin, 2005]. Profiloitiedot voidaan saada käyttäjältä selkeästi esimerkiksi kyselyillä tai epäsuorasti seuraamalla käyttäjän käyttäytymistä pitkällä aikavälillä.

Oletetaan, että Content(s) on asian s profiili eli joukko ominaisuuksia, jotka kuvaavat asiaa s. Profiili lasketaan yleensä erottamalla joukko ominaisuuksia asiasta s (yleensä sisällöstä), ja sitä käytetään päättämään asian soveltuvuus suosittelutarkoitukseen. Koska sisältöpohjaiset suosittelujärjestelmät on suunniteltu suurimmaksi osaksi suosittelemaan tekstipohjaisia asioita, niiden sisältö on yleensä kuvattu avainsanoilla [Adomavicius and Tuzhilin, 2005]. Sanan  $k_i$  tärkeys dokumentissa  $d_j$  määräytyy painotuksen  $w_{ij}$  mukaan, joka voidaan määrittellä useammalla eri tavalla.

Tunnetuin määrittelytapa avainsanoille on termin esiintymisfrekvenssi dokumentissa suhteutettuna sen käänteiseen esiintymisfrekvenssiin kokoelmassa (term frequency - inverse document frequency, TF-IDF) [Adomavicius and Tuzhilin, 2005]. Se voidaan määrittellä seuraavasti: Oletetaan, että  $N$  on kaikkien käyttäjälle mahdollisesti suositeltavien dokumenttien määrä ja avainsana  $k_i$  esiintyy niistä  $n_i$ :ssä dokumentissa. Lisäksi oletetaan, että  $f_{i,j}$  on avainsanan  $k_i$  esiintymiskerrat dokumentissa  $d_j$ . Nyt  $TF_{i,j}$  eli avainsanan  $k_i$  termin esiintymisfrekvenssi dokumentissa  $d_j$  määritellään

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}}, \quad (2)$$

jossa maksimi lasketaan yli kaikkien niiden avainsanojen  $k_z$ , jotka esiintyvät dokumentissa  $d_j$ , esiintymistiheyksistä  $f_{z,j}$ . On otettava kuitenkin huomioon, että avainsanat, jotka esiintyvät monissa dokumenteissa, eivät erittele relevantteja dokumentteja epärelevantteista. Siksi käänteistä dokumenttifrekvenssiä ( $IDF_i$ ) käytetään yhdessä termin esiintymisfrekvenssin ( $TF_{i,j}$ ) kanssa. Käänteinen dokumenttifrekvenssi määritellään usein kaavalla

$$IDF_i = \log \frac{N}{n_i}. \quad (3)$$

Tällöin TF-IDF-paino avainsanalle  $k_i$  dokumentissa  $d_j$  määritellään muodossa

$$w_{i,j} = TF_{i,j} \times IDF_i \quad (4)$$

ja dokumentin  $d_j$  sisältö määritellään yhtäsuuruutena

$$Content(d_j) = (w_{1j}, \dots, w_{kj}). \quad (5)$$

Kuten aikaisemmin on mainittu, sisältöpohjaiset suosittelujärjestelmät suosittelevat asioita, jotka ovat samankaltaisia kuin asiat, joista käyttäjä on pitänyt aikaisemmin. Useita suositteluehdokkaita verrataankin käyttäjän arvostelemiin

asioihin ja vain parhaiten vastaavia suositellaan. Muodollisemmin, olkoon  $\text{ContentBasedProfile}(c)$  käyttäjän  $c$  profiili, joka sisältää tämän käyttäjän mieltymykset ja vaatimukset. Profiili saadaan yleensä analysoimalla asioita, jotka käyttäjä on arvostellut, ja kootaan analysoimalla avainsanoja. Esimerkiksi  $\text{ContentBasedProfile}(c)$  voidaan määritellä avainsanojen painoja sisältäväksi vektoriksi  $(w_{c1}, \dots, w_{ck})$ , jossa jokainen paino  $w_{ci}$  merkitsee avainsanan  $k_i$  tärkeyttä käyttäjälle  $c$ , ja se voidaan laskea yksittäin arvioiduista sisältövektoreista. Sisältöpohjaisissa suosittelevissa järjestelmissä hyödyllisyysfunktio  $u(c,s)$  määritellään tavallisesti muodossa

$$u(c, s) = \text{score}(\text{ContentBasedProfile}(c), \text{Content}(s)). \quad (6)$$

Käyttäen yllämainittuja kaavoja sekä käyttäjää  $c$  vastaava vektori  $\text{ContentBasedProfile}(c)$  että asiaa  $s$  vastaava vektori  $\text{Content}(s)$ , voidaan esittää avainsanojen painoja kuvaavina TF-IDF vektoreina  $w_c$  ja  $w_s$ . Hyödyllisyysfunktio  $u(c,s)$  esitetään yleensä tiedonhaun kirjallisuudessa kosinilausekkeena

$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^K w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}}, \quad (7)$$

jossa  $K$  on kaikkien avainsanojen määrä järjestelmässä [Adomavicius and Tuzhilin, 2005].

Esimerkiksi jos käyttäjä  $c$  lukee monta verkkoartikkelia bioinformatiikasta, sisältöpohjainen suositteleva järjestelmä voi suositella käyttäjälle  $c$  muita bioinformatiikkaan liittyviä artikkeleita. Tämä johtuu siitä, että näissä artikkeleissa on enemmän bioinformatiikkaan liittyviä termejä, kuten hybridisaatio, sekvenssi tai rekombinanttiproteiini, kuin muista aiheista kertovissa artikkeleissa ja siksi  $\text{ContentBasedProfile}(c)$ , joka esitetään vektorina  $w_c$ , esittää nämä termit  $k_i$  suurilla painoarvoilla  $w_{ci}$ . Vastaavasti suositteleva järjestelmä, joka käyttää kosinilauseketta tai vastaavaa samankaltaisuusmittaria, asettaa suuremman hyödyn  $u(c,s)$  niille artikkeleille  $s$ , joissa on enemmän painotettuja bioinformatiikkatermejä  $w_{s}$ :ssä ja alhaisemman hyödyn niille, jossa bioinformatiikkatermejä on painotettu vähemmän.

Käyttäjän täytyy arvostella riittävä määrä asioita ennen kuin sisältöpohjainen suositteleva järjestelmä voi täysin ymmärtää käyttäjän mieltymykset ja esittää luotettavia suosituksia. Sen vuoksi uusi käyttäjä, joka on tehnyt vain vähän arvosteluja, ei voi saada tarkkoja suosituksia.

Sisältöpohjaiset suosittelevat järjestelmät ovat rajoitettu suosittelemaan vain sellaisia asioita, joiden ominaisuudet liittyvät selkeästi asiaan itseensä [Adomavicius and Tuzhilin, 2005]. Taatakseen riittävän määrän ominaisuuksia asian sisällön pitää olla koneellisesti luettavassa muodossa kuten tekstinä tai ominaisuudet pitää lisätä käsin. Viimeksi mainittu ei kuitenkaan yleensä ole mahdollista, koska suositeltavia asioita on yksinkertaisesti liian paljon ja työvoimaa vain rajallinen määrä. Tiedonhaun menetelmät toimivat hyvin tekstitiedon osal-



ta, mutta muissa asioissa ominaisuuksien automaattinen erottaminen ei välttämättä onnistu yhtä helposti. Esimerkiksi multimedia, kuten valokuvat, äänet ja videokuva, tuottavat ongelmia [Adomavicius and Tuzhilin, 2005].

Toinen ongelma rajoittuneessa sisältöanalyysissä on se, että jos kaksi eri asiaa esitetään täsmälleen samojen ominaisuuksien avulla, niitä ei voi erottaa toisistaan [Adomavicius and Tuzhilin, 2005]. Koska tekstidokumentit kuvataan usein tärkeimpien avainsanojen avulla, sisältöperustainen suosittelujärjestelmä ei voi tehdä eroa hyvin kirjoitetun ja huonosti kirjoitetun artikkelin välillä, jos ne käyttävät samoja termejä.

Sisältöpohjaisen suosittelun heikkous on se, että käyttäjälle saatetaan suositella sellaisia dokumentteja, jotka muistuttavat sisällöltään liiankin paljon aiemmin suositeltuja dokumentteja. Ideaalisti käyttäjälle pitäisi esitellä vaihteleva valikoima vaihtoehtoja eikä joukkoa samankaltaisia asioita. Ei ole esimerkiksi tarpeellista suositella käyttäjälle kaikkia Mika Kaurismäen elokuvia, jos hän on pitänyt niistä yhdestä. Käyttäjälle ei myöskään pystytä perustapauksissa tuottamaan suosituksia dokumenteista, jotka olisivat käyttäjälle kiinnostavia, mutta joiden sanat eivät täsmää käyttäjän sisältöprofiiliin. Näin käy esimerkiksi silloin, kun dokumentin kieli on eri kuin sisältöprofiilissa. Toinen ongelma sisältöpohjaisessa suosittelussa liittyy uusiin käyttäjiin. Käyttäjän nimittäin on arvosteltava riittävä määrä asioita ennen kuin suosittelujärjestelmä voi täysin ymmärtää hänen mieltymyksensä ja esittää luotettavia suosituksia. Sen vuoksi uusi käyttäjä, jolla on vain vähän tehtyjä arvosteluja, ei voi saada tarkkoja suositteluja.

Sisältöpohjaisen suosittelun erityinen vahvuus verrattuna seuraavassa kohdassa esiteltävään yhteistoiminnalliseen suositteluun on se, että myös sellaisia kohteita voidaan suositella, joihin kukaan muu käyttäjä ei ole vielä tutustunut.

## 2.2. Yhteistoiminnallinen suosittelu

Toisin kuin sisältöpohjaisissa suosittelumenetelmissä yhteistoiminnalliset suosittelujärjestelmät yrittävät ennustaa asioiden hyödyllisyyttä tietylle käyttäjälle perustuen muiden käyttäjien aikaisemmin arvostelemiin asioihin. Formaalisimmin, asian  $s$  hyödyllisyys  $u(c,s)$  käyttäjälle  $c$  arvioidaan käyttäjän  $c$  kanssa samankaltaisten käyttäjien  $c_j \in C$  asiaan  $s$  liittämien hyötyjen  $u(c_j,s)$  perusteella. Esimerkiksi voidakseen suositella käyttäjälle  $c$  elokuvia yhteistoiminnallinen suosittelujärjestelmä yrittää löytää käyttäjän  $c$  kanssa samankaltaisia käyttäjiä. Samankaltaisilla käyttäjillä tarkoitetaan sellaisia käyttäjiä, joilla on samankaltainen maku elokuvien suhteen ja jotka arvostelevat elokuvia samaan tapaan. Sen jälkeen suositellaan vain niitä elokuvia, joista samankaltaiset käyttäjät pitivät eniten.

Uudet käyttäjät muodostavat samanlaisen ongelman yhteistoiminnalliseen suositteluun kuin sisältöpohjaiseen suositteluun. Voidakseen tehdä tarkkoja

suosituksia järjestelmän on ensin opittava käyttäjän mieltymykset käyttäjän tekemistä arvosteluista. Useita menetelmiä on ehdotettu ongelman kiertämiseen [Adomavicius and Tuzhilin, 2005]. Useimmat ehdotuksista käyttävät lähestymistapana hybridisuositelumenetelmiä, joissa yhdistetään sisältöpohjaista ja yhteistoiminnallista suosittelua.

Suosittelujärjestelmiin lisätään säännöllisesti uusia asioita. Yhteistoiminnalliset suosittelujärjestelmät ovat riippuvaisia yksinomaan käyttäjien mieltymyksistä tehdessään suosituksia. Siksi uusi asia tarvitsee riittävän monen käyttäjän arvostelun ennen kuin järjestelmä voi suositella sitä. Myös tätä ongelmaa voidaan kiertää hybridisuositelumenetelmillä.

Yhteistoiminnalliset suosittelujärjestelmät kärsivät myös harvuudesta. Harvuus tarkoittaa tilannetta, jossa binäärinen matriisi  $C \times S$ , jossa  $C$  kuvaa käyttäjiä,  $S$  suositeltavia asioita ja arvo 1 tilannetta, jossa käyttäjä on ostanut tai arvostellut kyseisen asian, sisältää vain harvakseltaan arvoja. Koska arvoja on harvakseltaan, on samankaltaisten käyttäjien löytäminen hankalaa. Korkealaatuisten suositusten takaamiseksi on kuitenkin tärkeätä, että jokaiselle käyttäjälle löytyisi samankaltainen käyttäjä, jonka profiili vastaisi käyttäjän omaa. Harvuus johtuu siitä, että suositeltavia asioita ja käyttäjiä on suuri määrä. Esimerkiksi elokuvien suosittelujärjestelmässä voi olla monia elokuvia, jotka vain muutama käyttäjä on arvostellut. Näitä elokuvia suositellaan vain harvoin, vaikka käyttäjät olisivatkin arvostelleet ne korkealle. Myös käyttäjille, joilla on erilainen elokuvamaku muun populaation kanssa, ei löydy merkittävästi samankaltaisia käyttäjiä, joten he saavat huonoja suosituksia.

Yhteistoiminnallisen suosittelun suurin etu on siinä, että arvioinnin kohteena voi olla mitä tahansa sisältöjä, koska tietokoneen ei tarvitse käsitellä kohteiden sisältöä tuottaakseen suosituksia (toisin kuin esimerkiksi sisältöpohjaisessa suosittelussa) [Burke, 2002]. Kohteet voivat esimerkiksi olla tekstidokumentteja, videoita tai abstraktimpia asioita. Toinen etu on se, että yhteistoiminnallisessa suosittelussa suosittelu voi perustua monimutkaisiin ja vaikeasti ilmaistaviin mielipiteisiin, kuten makukysymyksiin ja laatuun. Yhteistoiminnallisen suosittelun etuja on myös mahdollisuus löytää ennalta arvaamattomia suosituksia kohteista, jotka eivät esimerkiksi sisällöllisesti muistuta toisiaan, mutta joista samanhenkiset käyttäjät ovat pitäneet.

### 2.3. Demografiapohjainen suosittelu

Demografista tietoa, kuten ikää, sukupuolta tai asuinpaikkaa, voidaan käyttää tunnistamaan, minkälaiset käyttäjät pitävät tietyistä asioista. Näiden tietojen avulla voidaan tuottaa suosituksia kaikille samaan demografiseen ryhmään kuuluville, kuten esimerkiksi alle 20-vuotiaille naisille, jotka asuvat paikkakunnalla, jonka väkiluku on yli 50 000. Demografinen suosittelu mahdollistaa käyttäjätietojen yleistämisen siten, että käyttäjät kuvataan demografisten ryhmien

avulla eikä yksittäisten tuotteiden, kuten tapahtuu yhteistoiminnallisessa suosittelussa. Tämä pienentää vertailtavien ryhmien kokoa, jolloin käsittelyaika nopeutuu. Demografisten tietojen saaminen voi kuitenkin olla hankalaa [Pazzani, 1999]. Suositteletut uudelle käyttäjälle tehdään etsimällä ensin, mihin ryhmään käyttäjä kuuluu, ja sen jälkeen yhdistämällä ryhmän muiden käyttäjien ostokäyttäytyminen. Ryhmittely voidaan tehdä monella eri tavalla. Tässä tutkielmassa käytetään klusterointia. Klusterointia on tutkittu paljon etenkin tilastotieteessä ja koneoppimisen yhteydessä [Aïmeur et al., 2006]. Sen tavoitteena on muodostaa klustereita eli ryhmiä samankaltaisista asioista. Asiat tietyssä klusterissa ovat samankaltaisia keskenään, mutta erilaisia kuin muissa klustereissa olevat. Klusterointialgoritmi tarvitseekin mitan, jolla voidaan laskea, kuinka samankaltaisia tai erilaisia asiat ovat. Sen vuoksi asiat yleensä esitetään vektoreina moniulotteisessa avaruudessa ja etäisyydet lasketaan tyypillisesti käyttäen euklidista etäisyyttä (tai Minkowskin etäisyyttä) huomioiden kaikki tai vain osa ulottuvuuksista [Aïmeur et al., 2006]. Jokaista klusteria edustaa sentroidi tai medoidi, ja joskus myös informaatio säteestä ja tiheydestä. Sentroidi on virtuaalinen piste, joka vastaa kaikkien klusterin pisteiden keskipistettä. Medoidi taas vastaa mediaanipistettä klusterissa.

Demografisessa suosittelussa klusterointia käytetään luomaan edellä mainitut käyttäjäryhmät kaikista käyttäjistä. Käyttäjät muodostavat objektit ja jokainen ulottuvuus vastaa yhtä merkittävää demografista piirrettä. Tietyssä klusterissa  $C$  tiheys edustaa sen sisältämien käyttäjien määrää ja säde käyttäjien demografista erilaisuutta. Sen jälkeen jokaisen klusteriin  $C$  kuuluvan käyttäjän vanhat mieltymykset yhdistetään kuvaamaan koko klusterin  $C$  mieltymyksiä. Yksinkertaisimmassa muodossaan tämä yhdistelmä voi koostua listasta asioita  $P_C = \{p_1, p_2, \dots, p_c\}$ , joille käyttäjät ovat antaneet positiivista palautetta. Kun uusi käyttäjä tarvitsee suosituksia, suosittelujärjestelmä laskee käyttäjää lähinnä olevan klusterin  $C$  ja tuottaa suosittelulistan  $P_C$ . Toisin sanoen demografiapohjaiset suosittelujärjestelmät käyttävät käyttäjien välisiä korrelaatioita suosittelevien tuottamiseen.

Demografisen suosittelun suurin ongelma on se, että tunnistetut ryhmät eivät aina ennakoivat sitä, mistä käyttäjät ovat kiinnostuneet ja mistä eivät. Esimerkiksi suurkaupungissa asuva hyvätuloinen 20-vuotias mies voi olla hyvinkin kiinnostunut maanviljelystä.

#### **2.4. Hyödyllisyyspohjainen suosittelu**

Hyödyllisyyspohjainen suosittelu (utility-based recommendation) perustuu kullekin asialle erikseen laskettavaan hyödyllisyysarvoon, joka kuvaa, kuinka hyödyllinen kyseinen asia on käyttäjälle [Burke, 2002]. Käyttäjälle esimerkiksi nopea toimitus saattaa merkitä enemmän kuin tuotteen hinta. Hyödyllisyysfunktiossa voidaan painottaa nopeaa toimitusaikaa, jos toinen tuote löytyy os-

tohetkellä varastosta ja toinen ei. Käyttäjakohtainen hyödyllisyysfunktio voidaan muodostaa esimerkiksi interaktiivisella käyttöliittymällä, jonka avulla käyttäjä voi valita ja painottaa haluamiaan ominaisuuksia tarpeidensa mukaan.

Interaktiivisen hyödyllisyyspohjaisen suosittelun hyviä puolia on se, että järjestelmä mukautuu välittömästi käyttäjän muuttaessa asioihin liittyviä priorisointejaan. Hyödyllisyysfunktiossa voidaan lisäksi ottaa huomioon asioihin epäsuorasti liittyviä tietoja kuten niiden toimitusaika ja varastotilanne. Hyvänä puolena interaktiivisessa hyödyllisyyspohjaisessa suosittelussa on se, että se ei sisällä oppimisvaihetta, koska käyttäjältä pyydetään selkeät priorisoinnit etsittäville kohteille. Heikkoutena taas vastaavasti on se, että käyttäjältä vaaditaan syötteitä hyödyllisyysfunktion luomiseksi toisin kuin epäsuoraan tietoon perustuvissa menetelmissä.

## 2.5. Tietämyspohjainen suosittelu

Tietämyspohjaisten suosittelujärjestelmien toiminta perustuu tietoon käyttäjistä ja tuotteista haettaessa suosituksia ja päätettäessä, mitkä tuotteet vastaavat käyttäjän vaatimuksia. Esimerkiksi ravintoloiden suositteluun tarkoitettu Entree-järjestelmä [Burke, 2000] käyttää uudessa kaupungissa ravintoloiden suositteluun samankaltaisuutta ravintoloihin, jotka käyttäjä tuntee tai joista hän pitää. Järjestelmä antaa käyttäjän navigoida siten, että käyttäjä ilmoittaa mielipiteensä ehdotetusta ravintolasta. Jos ravintola esimerkiksi on käyttäjän mielestä liian kallis mutta muuten sopiva, voi hän käskää järjestelmää hakemaan halvempia ravintoloita. Seuraavat ehdotukset ovatkin sitten samankaltaisia, mutta halvempia.

Entree-järjestelmän taustalla käytetään FindMe-tekniikkaa, joka on yksi tietämyspohjaisista samankaltaisuuden hakumenetelmistä. Se jakautuu kahteen olennaisesti erilaiseen hakumuotoon: samankaltaisuuden löytämiseen ja hienosäätöön (tweak application). Samankaltaisuuden löytämisessä käyttäjä on valinnut listasta jonkin tuntemansa asian ja pyytänyt muita samankaltaisia asioita. Haun aluksi suuri määrä ehdokkaita haetaan tietokannasta, jotka sitten järjestetään samankaltaisuuden mukaan ja vain muutama alkupään asia näytetään käyttäjälle. Hienosäädössä periaate on sama, mutta ehdokasjoukkoa suodatetaan ennen järjestämistä siten, että vain ehdokkaat, jotka täyttävät hienosäädön ehdot, jätetään. Jos esimerkiksi käyttäjä vastaa asiaan X hienosäädöllä "mukavampi", järjestelmä päättää X:n "mukavuuden" ja hylkää kaikki muut ehdokkaat paitsi ne, joiden "mukavuusarvo" on suurempi.

Tietämyspohjaisissa suosittelujärjestelmissä heikkoutena on niiden tarvitseman tietämyskannan kokoaminen. Lisäksi eri käyttäjät voivat käsittää asioiden paremmuuden eri tavalla kuin suosittelujärjestelmän ylläpitäjät.

Tietämyspohjaiset suosittelujärjestelmät välttävät uuden käyttäjän ja uuden asian ongelmat, koska niiden toiminta ei riipu käyttäjien tekemistä arvosteluista

[Burke, 2000]. Niiden ei myöskään tarvitse kerätä tietoa yksittäisestä käyttäjästä ja hänen mieltymyksistään, koska suosittelut ovat riippumattomia hänen mieltymyksistään. Tämän takia tietämysperustaiset järjestelmät ovat paitsi arvokkaita järjestelmiä yksinään mutta myös täydentämällä muunlaisia suosittelujärjestelmiä.

## 2.6. Hybridisuosittelujärjestelmät

Useat suosittelujärjestelmät käyttävät hybridisuosittelua yhdistämällä sisältöpohjaisen ja yhteistoiminnallisen suosittelun. Tämä auttaa välttämään tiettyjä rajoituksia sisältöpohjaisessa suosittelussa ja yhteistoiminnallisessa suosittelussa [Adomavicius and Tuzhilin, 2005]. Tavat, jotka yhdistävät sisältöpohjaisen suosittelun ja yhteistoiminnallisen suosittelun, voidaan luokitella seuraavasti:

1. yhteistoiminnallinen ja sisältöpohjainen suosittelu toteutetaan erikseen ja niiden suositukset yhdistetään,
2. yhteistoiminnalliseen suositteluun sisällytetään ominaispiirteitä sisältöpohjaisesta suosittelusta,
3. sisältöpohjaiseen suositteluun sisällytetään ominaispiirteitä yhteistoiminnallisesta suosittelusta,
4. muodostetaan yleinen yhdistävä malli, johon sisällytetään sekä yhteistoiminnallisen että sisältöpohjaisen suosittelun piirteet.

Yksi tapa rakentaa hybridisuosittelujärjestelmä on toteuttaa erilliset yhteistoiminnallinen ja sisältöpohjainen suosittelujärjestelmä. Sen jälkeen on kaksi vaihtoehtoista tapaa jatkaa. Ensimmäisessä vaihtoehdossa yhdistetään eri suosittelujärjestelmien tuottamat suositukset. Toisessa vaihtoehdossa valitaan sen suosittelujärjestelmän suositukset, joka antaa luotettavimmat suositukset jollakin mittarilla mitattuna. Mittarina voidaan käyttää esimerkiksi luottamusta tai yhtäpitävyyttä käyttäjän aikaisempien arvostelujen kanssa.

Useat hybridisuosittelujärjestelmät perustuvat perinteisiin yhteistoiminnallisiin tekniikoihin, mutta säilyttävät silti sisältöpohjaiset profiilit jokaiselle käyttäjälle. Näitä sisältöpohjaisia profiileja käytetään, yhteisesti arvosteltujen asioiden sijaan, laskettaessa käyttäjien samankaltaisuutta. Tämä sallii yhteistoiminnallisessa suosittelussa ongelmana olevan harvuuden voittamisen, koska vain harvalla käyttäjäparilla on merkittävää määrää yhteisesti arvosteltuja asioita. Toinen hyöty tästä lähestymistavasta on, että käyttäjälle voidaan suositella asiaa paitsi silloin, kun muut samankaltaiset käyttäjät ovat arvostelleet asian korkealle, niin myös silloin, kun asia saa hyvän arvion käyttäjän profiilia vastaan.

Suosituin lähestymistapa yhteistoiminnallisten ominaispiirteiden sisällyttämisessä sisältöpohjaiseen suositteluun on käyttää ulottuvuuksien vähentämistä (dimensionality reduction) joukkoon sisältöpohjaisia profiileja, jolloin saadaan yhteistoiminnallinen näkymä käyttäjäprofiileihin.

Hybridisuosittelujärjestelmiä voidaan parantaa tietämuspohjaisilla suosittelutekniikoilla, kuten tapauskohtaisella päättelyllä, jolloin parannetaan suosittelutarkkuutta ja kierretään joitakin perinteisten suosittelujärjestelmien rajoituksia kuten uuden käyttäjän ja asian ongelmat.

### 3. Suosittelujärjestelmien laajentaminen

Suurin osa suosittelujärjestelmistä perustaa suosittelunsa rajoittuneeseen ymmärrykseen käyttäjistä ja asioista [Adomavicius and Tuzhilin, 2005]. Ne eivät hyödynnä täysin käyttäjän tapahtumahistoriaa tai muita saatavilla olevia tietoja. Esimerkiksi normaali yhteistoiminnallinen suosittelujärjestelmä ei käytä suositteluja tehdessään ollenkaan hyödyksi käyttäjistä ja asioista tehtyjä profiileja, vaan perustaa suosittelunsa yksinomaan käyttäjien tekemiin arvosteluihin. Vaikka edistystä on tapahtunut käyttäjien ja asioiden profiilien sisällyttämisessä suosittelujärjestelmiin, niin profiilit ovat kuitenkin vielä yksinkertaisia eivätkä hyödynnä kehittyneempiä profilointitekniikoita. Perinteisten avainsanojen ja demografiaryhmien lisäksi tarjolla on tiedonlouhintaan ja käyttäjän kiinnostusten ominaispiirteisiin perustuvia profilointimenetelmiä [Adomavicius and Tuzhilin, 2005]. Tiedonlouhintaan perustuvia profilointitekniikoita on käytetty pääasiassa verkkosivujen suositteluun, mutta arvostelupohjaisissa suosittelujärjestelmissä ne eivät ole laajalti omaksuttuja.

Nykyinen suosittelujärjestelmien sukupolvi toimii kaksiulotteisessa *Käyttäjä × Asia* -tilassa [Adomavicius and Tuzhilin, 2005]. Tämä tarkoittaa sitä, että suosittelujärjestelmien suositukset perustuvat vain tietoihin käyttäjistä ja asioista, eivätkä järjestelmät ota huomioon kontekstia, jolla voi joissakin tapauksissa olla suuri merkitys. Kuitenkin monissa tapauksissa asian hyödyllisyys voi riippua merkittävästi ajasta, kuten vuodenajasta tai viikonpäivästä. Se voi myös riippua siitä, kenen kanssa tai millä tavoin asiaa käytetään. Näissä tapauksissa ei siis riitä, että suosittelujärjestelmä suosittelee asiaa, vaan sen on suositusta tehdessään otettava huomioon myös aika, paikka ja käyttäjän seuralaiset. Esimerkiksi lomapakettia suositellessaan suosittelujärjestelmän on otettava huomioon vuodenaika, matkakumppanit, matkustusolosuhteet ja muita kontekstista riippuvia asioita. Käyttäjällä voi myös olla toisenlaiset mieltymykset elokuvien suhteen käydessään elokuvateatterissa poikaystävänsä kanssa lauantai-iltana kuin katsoessaan vuokraelokuvaa vanhempiensa kanssa keski- viikko-iltana. On siis tärkeää laajentaa perinteiset kaksiulotteiset suosittelujärjestelmät tukemaan moniulotteisuutta.

Useimmat nykyisistä suosittelujärjestelmistä käsittelevät arvosteluja vain yhden muuttujan suhteen, kuten kirjojen tai elokuvien arvostelut, mutta esimerkiksi ravintoloiden suosittelujärjestelmään on tärkeää sisällyttää monesta muuttujasta riippuvat arvostelut.

Monet suosittelujärjestelmät ovat tungettelevia siinä mielessä, että ne tarvitsevat suoraa palautetta ja usein myös merkittävää määrää osallistumista käyttäjältä. Esimerkiksi ennen kuin suosittelujärjestelmä voi suositella artikkelia, täytyy suosittelujärjestelmän tietää käyttäjän aikaisemmin lukemien artikkelien arvostelut. Usein arvioita tarvitaan useampi kappale. Koska on usein epäkäytännöllistä saada käyttäjä antamaan itse arvostelut, jotkut suosittelujärjestelmät käyttävät tungettelemattomia menetelmiä oikeiden arvosteluiden arviointiin. Esimerkiksi käyttäjän artikkelin lukemiseen käyttämä aika voi olla tällainen menetelmä. Jos käyttäjä siis lukee artikkelia kauemmin, voidaan olettaa, että hän pitää artikkelia kiinnostavana tai relevanttina ja arvostelisi sen korkealle. Kuitenkin tungettelemattomat menetelmät, kuten käyttäjän artikkelin lukemiseen käyttämä aika, ovat usein epätarkkoja eivätkä voi täysin korvata käyttäjän suoraan antamia arvosteluja. Sen vuoksi ongelma tungettelevuuden minimoimisen ja silti tietyn tarkkuustason säilyttämisen välillä jää suosittelujärjestelmien kehittäjien ratkaistavaksi.

Yksi tapa tutkia tungettelevuusongelmaa on määrittää optimaalinen määrä arvosteluja, jotka suosittelujärjestelmä sitten kysyy suoraan uudelta käyttäjältä. Esimerkiksi ennen kuin suosittelujärjestelmä suosittelee yhtään elokuvaa, suosittelujärjestelmä pyytää käyttäjää arvioimaan tietyn määrän elokuvia (esimerkiksi 20). Tällä menetelmällä on loppukäyttäjälle hintansa. Hinnalla tarkoitetaan tässä tapauksessa käyttäjälle aiheutuvaa lisätyötä elokuvien arvioinnista. Hinta voidaan muotoilla seuraavasti. Yhden elokuvan arvioimisen hinta on  $C$ , joten  $n$  kappaletta elokuvia arvioidaan hinnalla  $C \cdot n$ . Tungettelemattomuusongelma voidaan täten muuttaa arvosteltavien elokuvien optimaalisen määrän  $n$  etsinnäksi. Jokainen käyttäjän tekemä lisäarvostelu parantaa suosittelun tarkkuutta ja siten hyödyttää käyttäjää.

Useimmat suosittelujärjestelmistä ovat mukautumattomia siinä mielessä, että ne on sidottu toimittajan ennalta määräämiin suosittelumenetelmiin, ja sen takia loppukäyttäjä ei voi räätälöidä suosituksia omiin tarpeisiinsa [Adomavicius and Tuzhilin, 2005]. Lisäksi useimmat suosittelujärjestelmät suosittelevat yksittäisiä asioita yksittäisille käyttäjille eivätkä käsittele kokonaisuuksia. Kokonaisuuksien suosittelu on kuitenkin tärkeätä monissa sovelluksissa, kuten tuotemerkkien ja tuoteluokkien suosittelussa tietyille käyttäjäryhmälle. Esimerkiksi suosittelujärjestelmä voisi suositella kesäisin lapsiperheille matkoja Muumimaailmaan.

Suosittelujärjestelmiä käsittelevässä kirjallisuudessa on kattavasti osoitettu ongelma suosittelujärjestelmien tehokkuutta mittaavien mittareiden kehittämisessä [Adomavicius and Tuzhilin, 2005]. Suurimmassa osassa kirjallisuutta suorituskyvyn arviointi tehdään kattavuuden ja tarkkuuden perusteella. Kattavuus mittaa osuutta asioista, joista suosittelujärjestelmä pystyy tuottamaan ennustuksen. Tarkkuusmittarit puolestaan voivat olla joko tilastollisia tai päätöstä

tukevia. Tilastolliset tarkkuusmittarit pääasiallisesti vertaavat arvioituja arvosteluja *Käyttäjät* × *Asia* -matriisin todellisiin arvoihin R ja käsittävät keskivirheen (Mean Absolute Error, MAE), neliökeskivirheen (Root Mean Square Error, RMSE) sekä ennustusten ja todellisten arvojen välisen korrelaation. Päätöstä tukevat mittarit määrittävät, kuinka hyvin suosittelujärjestelmä pystyy tekemään ennusteita hyvin merkittävistä asioista, eli sellaisista asioista, jotka käyttäjä arvostelisi korkealle. Ne sisältävät tiedonhaun perinteiset mittarit kuten tarkkuus, saanti, F-mitta (F-measure) ja ROC (Receiver Operating Characteristic). F-mitta yhdistää saannin ja tarkkuuden. Siinä saantia ja tarkkuutta painotetaan parametrin  $\alpha$  avulla. ROC havainnollistaa kompromissin oikeiden positiivisten ja väärin positiivisten osuuden välillä.

Suosioista huolimatta näillä kokemusperäisillä arviointimittareilla on rajoituksensa. Yksi rajoitus on, että arviointimittareita tyypillisesti käytetään harjoitusaineistoon, jonka käyttäjä on suostunut arvostelevaan [Adomavicius and Tuzhilin, 2005]. Kuitenkin asiat, jotka käyttäjä on suostunut arvostelevaan, muodostavat todennäköisesti vinoutuneen näytteen. Käyttäjät voivat esimerkiksi arvostella vain ne asiat, joista he itse pitivät. Toisin sanoen kokemusperäiset arviointimenetelmät näyttävät vain, kuinka tarkka suosittelujärjestelmä on asioissa, jotka käyttäjä on suostunut arvostelevaan, kun taas suosittelujärjestelmän kykyä arvioida satunnainen asia kunnolla, josta sen pitäisi suoriutua normaalikäytössä, ei mitata. Ymmärrettävästi kontrolloitujen käyttäjätestien tekeminen on kallista ja aikaa vievää. Sen vuoksi testit, joissa testataan puolueettomasti satunnaisia asioita, ovat harvinaisia. Kuitenkin korkeatasoiset testit ovat välttämättömiä, että ymmärretään täysin edut ja rajoitukset ehdotetuista suosittelutekniikoista.

#### 4. Yhteenveto

Tämän tutkielman tavoitteena oli tutkia, mitä erilaisia suosittelujärjestelmiä on kehitetty ja selvittää niiden hyvät ja huonot puolet sekä pohtia, miten suosittelujärjestelmiä voitaisiin laajentaa toimimaan paremmin. Kuten olemme saaneet huomata, täydellistä suosittelujärjestelmää ei ole vielä kehitetty vaan edessä on vielä pitkä matka. Sisältöpohjaisissa ja yhteistoiminnallisissa suosittelujärjestelmissä ongelmana ovat uudet asiat ja käyttäjät. Lisäksi yhteistoiminnalliset suosittelujärjestelmät kärsivät harvuudesta. Demografiapohjaiset suosittelujärjestelmät eivät pysty täydellisesti päättelemään käyttäjien mieltymyksiä pelkkin demografisten tietojen perusteella. Hyödyllisyyspohjaisissa suosittelujärjestelmissä taas ongelmana on tietojen pyytäminen suoraan käyttäjältä. Tietämyspohjaisissa suosittelujärjestelmissä ongelmia tuottaa tietämyskannan kokoaminen. Tällä hetkellä parhaiten näyttäisivät toimivan erilaiset hybridijärjestelmät, joissa on yhdistelty muita suosittelujärjestelmiä ja niiden hyviä puolia.



## Viiteluettelo

- [Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on KDE* **17**, 6 (June 2005), 734-749.
- [Aïmeur et al., 2006] Esma Aïmeur, Gilles Brassard, Jose Manuel Fernandez and Flavien Serge Mani Onana, Privacy-preserving demographic filtering. In: *Proc. of the 2006 ACM Symposium on Applied Computing SAC '06* (2006), 872-878.
- [Burke, 2000] Robin Burke, Knowledge-based recommender systems. In A. Kent (ed.), *Encyclopedia of Library and Information Systems* **69** (2000).
- [Burke, 2002] Robin Burke, Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* **12** (2002), 331-370.
- [Ghazanfar and Prugel-Bennett, 2010] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, A scalable, accurate hybrid recommender system, In: *Proc. of 3rd International Conference on Knowledge Discovery and Data Mining, 2010. WKDD '10*, **9-10** (January 2010), 94-98.
- [Pazzani, 1999] Michael Pazzani, A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Rev.* **13**, 5-6 (1999), 393-408.

# Videopelien vika? – kirjallisuuskatsaus moderniin väkivaltaisten videopelien tutkimukseen

**Auvo Salmenharju**

## Tiivistelmä.

Tässä tutkielmassa käsitellään väkivaltaisia videopelejä ja niiden vaikutuksia pelaajiin. Joidenkin tutkimusten mukaan väkivaltaiset videopelit aiheuttavat pelaajissaan väkivaltaista käyttäytymistä ja aggressiota, kun taas toiset tutkimukset eivät ole löytäneet kyseistä yhteyttä. Tutkielmassa perehdytään aluksi videopelikonsolien ja väkivaltaisten videopelien historiaan. Lopuksi tutkielmassa esitellään ja analysoidaan neljä väkivaltaisten videopelien vaikutuksia tutkinutta tutkimusta.

**Avainsanat ja -sanonnat:** videopelit, väkivaltainen käytös, väkivaltaiset videopelit

**CR-luokat:** K.2, K.4, K.8

## 1. Johdanto

Nykyään suuri osa lapsista ja nuorista pelaa videopelejä säännöllisesti. Maailmalla on myyty tähän päivään mennessä yli miljardi erilaista videopelikonsolia [VGchartz, 2010]. Kyseiseen lukuun voidaan vielä lisätä myös tietokoneet ja matkapuhelimet. Videopelaamisesta voidaan siis puhua isona ja ennen kaikkea maailmanlaajuisena ilmiönä.

Ilmiö on kiinnostanut myös tutkijoita ympäri maailmaa ja etenkin 2000-luvulla on tehty huomattava määrä tutkimuksia erilaisista videopeleihin ja pelaamiseen liittyvistä aiheista. Yksi suosituimmista tutkimusaiheista on ollut väkivaltaisia aiheita sisältävät videopelit ja niiden vaikutus nuoriin pelaajiin. Tutkimuksissa on usein päädytty ristiriitaisiin tuloksiin. Joidenkin tutkimusten mukaan väkivaltaiset videopelit aiheuttavat väkivaltaista käytöstä nuorissa pelaajissa, kun taas toiset tutkimukset kumoavat kyseisen yhteyden kokonaan. On myös sanottu, että videopeliteollisuudesta on tullut helppo syntipukki useisiin nykypäivän yhteiskunnan ongelmiin [Cumberbatch, 2004].

Puhuttaessa väkivallasta, väkivaltaisista peleistä ja niiden mahdollisista vaikutuksista on tärkeää jotenkin määritellä, mitä väkivallalla itse asiassa tarkoitetaan. Vaikka useimmiten sanalla "väkivalta" viitataan nimenomaan ruumiilliseen voimankäyttöön, johtavat sanan erilaiset käyttötavat väkivaltatutkijoiden, päättäjien ja suuren yleisön keskuudessa hämmennykseen [Gentile, 2007]. Lasten määritelmässä taistelemisen ja ampumisen peleissä eivät välttä-

mättä olleet sinällään mitenkään erityisen väkivaltaista toimintaa [Ermi et al., 2004].

Tässä tutkielmassa haluan selvittää, miten erilaiset tutkimusmenetelmät vaikuttavat väkivaltaisten videopelien tutkimusten tuloksiin ja voivatko ne selittää kyseisen aiheen tutkimuksissa saadut ristiriitaiset tulokset. Esittelen aluksi hieman videopelaamisen historiaa ja sen tärkeimpiä merkkipaaluja. Sen jälkeen esittelen tarkemmin käsitteen *väkivoaltainen videopeli* sekä videopelien yhteydessä Euroopassa käytettävän ikärajojen luokitusjärjestelmän. Seuraavaksi käyn läpi muutamia 2000-luvulla julkaistuja väkivaltaisia videopelejä käsitteleviä tutkimuksia keskittyen etenkin niissä käytettyihin tutkimusmenetelmiin ja näillä saatuihin tuloksiin. Lopuksi teen yhteenvedon edellä mainituista tutkimusmenetelmistä ja tuloksista.

## 2. Videopelien historia

Termin *videopeli* voidaan katsoa käsittävän kaikki erilaiset elektroniset pelit, joiden pelaamiseen tarvitaan käyttöliittymä ja näyttölaite. Videopelit ovat kehittyneet useilla eri alustoilla kuten erillisillä pelihalleissa pelattavilla kolikkopelilaitteilla, henkilökohtaisilla tietokoneilla ja pelkästään pelaamista varten suunnatuilla pelikonsoleilla. Videopelejä kontrolloidaan peliohjaimella, jonka ominaisuudet vaihtelevat alustan mukaan. Pelikonsoleilla ohjaimet ovat varta vasten pelaamista varten suunniteltuja, mutta tietokonepelejä pelataan yleisimmin näppäimistöllä ja hiirellä. [Wikipedia, 2010a]

Videopelit ovat kehittyneet tietokoneiden laskentatehon lisääntyessä viimeisten 50 vuoden aikana. Voidaan jopa väittää, että videopelit ovat työntäneet tietokoneiden prosessoreiden laskentatehon rajoja eteenpäin saavuttaakseen yhä terävämpää grafiikkaa ja sulavamman pelikokemuksen. [Guillory, 2010]

Pelikonsoleiden historia on jaettu yleisimmin seitsemään eri vaiheeseen tai sukupolveen. Yhdessä sukupolvessa on aina tietyllä aikavälillä ilmestyneitä ja samantapaisia pelikonsoleita. Pelikonsolisukupolvet voidaan Guilloryn [2010] mukaan jakaa myös määrältään pienempiin mutta sisällöltään laajempiin aikausiin. Guilloryn [2010] kuvailema jaottelu on seuraava:

- 1) Varhaisvuodet 1960–1975. Ensimmäiset pelikonsolit olivat teknisesti vaatimattomia yhden tai kahden pelin laitteita. Pelit olivat graafisesti yleensä karkeita ja mustavalkoisia. Atarin kehittämän Pong-pelin sanotaan popularisoineen pelikonsolit suurelle yleisölle.
- 2) Pelikonsolien vallankumous 1975–1990. Pelikonsoleilla pystyi pelaamaan useita eri pelejä vaihdettavien pelikasettien ansiosta. Pelikonsoleiden ja pelien valmistajat alkoivat erottua toisistaan. Atarin lisäksi markkinoille tulivat myös Sega ja Nintendo. Teknologian kehitys myös mahdollisti kotikonsoleille entistä

näyttävämpiä pelejä, mikä myös osaltaan viestitti pelihallien kolikkopelien al-  
kavasta alamäestä. Vuoteen 1987 mennessä Nintendo 8-bit -konsolista oli tullut  
Yhdysvaltojen parhaiten myyvä pelikonsoli/lelu. Tietokonepelien suosio ja  
myynti jää jatkuvasti konsolien varjoon. Suurimmaksi osaksi tämä johtuu peli-  
en kalliista hinnasta ja tietokoneiden harvinaisuudesta tavallisilla kuluttajilla.

3) Kannettavat pelikonsolit ilmestyvät 1987–2000. Suuret konsolivalmistajat  
alkoivat valmistaa myös kannettavia pelikonsoleita. Parhaiten kilpailussa pärjä-  
si Nintendon valmistama Game Boy. Vaikka Game Boy oli kilpailijoiden kan-  
nettavia konsoleita teknillisesti alkeellisempi, se voitti markkinat halvalla hin-  
nallaan ja runsaalla määrällä tunnettuja pelinimikkeitä, jotka olivat peräisin  
Nintendo 8-bit -konsolilta. Nintendon alkaessa ajaa Segaa pois konsolimarkki-  
noilta 1990-luvun puolella välissä valmisteli toinen elektroniikkajätti sinne  
saapumista. Sony julkaisi 1995 vuonna Playstation-konsolinsa, jonka tarkoituk-  
sena oli alusta alkaen tavoitella aikuisempaa pelaajayleisöä.

4) Kolmas ja neljäs videopelihistorian sukupolvi olivat osittain päällekkäisiä,  
kun konsolivalmistajien sotaan markkinoista lisätään vielä erilaiset kannettavat  
pelikonsolit. Samalla myös tietokonepelien puolella tapahtui kehitystä Win-  
dows 95 -käyttöjärjestelmän julkaisun tuodessa tietokonepelien kehittäjille yh-  
tenäisen ja vakaan alustan, johon suunnitella pelejä.

5) Konsolisodat 1995–2005. Vuonna 2001 Sega myönsi tappionsa konsoli-  
markkinoilla ja keskittyi tekemään pelkästään pelejä. Markkinoille saapui jäl-  
leen uusi kilpailija, ohjelmistojätti Microsoft. Microsoftin vuonna 2001 julkai-  
sema Xbox-konsoli alkoi hämärtää konsoli- ja tietokonepelaamisen rajoja tieto-  
konemaisilla piirteillään joita olivat esimerkiksi kiintolevy ja verkkoyhteys.  
Konsolisodassa vastakkain olivat nyt kolme suurta valmistajaa: Nintendo, Mic-  
rosoft ja Sony. Samalla tietokonepelaaminen jatkoi vahvaa kasvuaan.

6) Kokonaisvaltainen verkko- ja mobiilipelaaminen vuodesta 2005 eteenpäin.  
Internet-laajakaistayhteyksien kehittyessä nopeammiksi alkoivat myös pelit  
hyödyntää verkkoyhteyksiä. Tietokoneiden tapaan jokaisessa viimeisen suku-  
polven pelikonsolissa on mahdollisuus hyödyntää laitteella internet-yhteyttä  
moninpelaamisessa ja sisällön hakemisessa. Nintendon vuonna 2006 julkaise-  
massa Wii-pelikonsolissa oleva liikeohjain ja pelaamisen helppous houkuttele-  
vat uudenlaista yleisöä pelien pariin. Wii-pelikonsoli sopii kaikenikäisille, ja  
sillä pelaaminen onnistuu, vaikka ei olisi koskaan aikaisemmin edes kokeillut  
videopelejä. Luonnollisesti myös Sony ja Microsoft julkaisivat omat liikeoh-  
jaimensa konsoleilleen tajutessaan niiden räjähdysmäisen suosion. Matkapuhe-  
limien ja kannettavien pelikonsolien laitteistopuolen kehitys on johtanut ti-  
lanteeseen, jossa käyttäjä pystyy pelaamaan paikastaan riippumatta entistä laa-  
dukkaampia ja monipuolisempia pelejä.

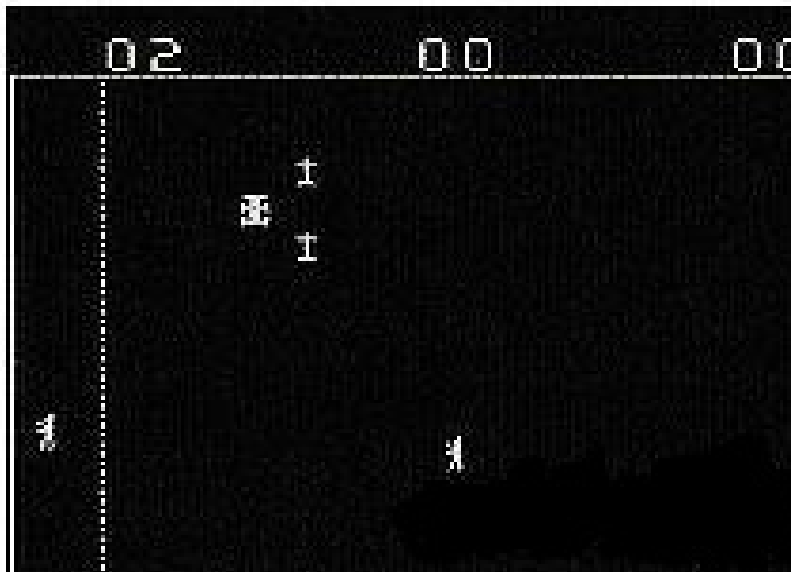
Nykypäivän pelikonsoleilla pystyy tekemään paljon muutakin kuin vain pelaamaan. Playstation 3-pelikonsolilla voi katsella DVD- sekä Blu-Ray- elokuvia. Samalla konsoli pystyy toistamaan esimerkiksi videoita, musiikkia tai valokuvia samassa lähiverkossa olevalta tietokoneelta. Playstation 3-, Xbox360- ja Nintendo Wii -konsoleissa on kaikissa myös mahdollisuus käyttää internet-selainta. Internetiin kytkettynä konsolit pystyvät hakemaan itseensä päivityksiä, ja pelaajat pystyvät hankkimaan maksullista ja ilmaista lisäsisältöä peleihin ja konsoliin.

### **3. Väkivaltaiset videopelit ja pelien ikäraajat**

Tässä luvussa tarkastellaan väkivaltaisia aiheita sisältäviä videopelejä, niiden historiaa sekä pelien ikärajoja määrittäviä järjestöjä. Luvussa mainittu termi *ensimmäisen persoonan ammuntapelejä* (engl. First Person Shooter) on ampumiseen keskittyvä videopeligenre, jossa pelimaailma esitetään pelihahmon näkökulmasta (ks. kuva 3 ja 5). Tyypillisesti kuvassa näkyy myös hahmon kädessä oleva ase, jolla ammutaan ympäristössä liikkuvia vihollisia tai muita kohteita. Luvussa mainittu termi *Avoin pelimaailma* (engl. Open World tai Sandbox) on videopeleissä käytettävä virtuaalimaailma jota pelaaja voi vapaasti tutkia. Peli-suunnittelun lähtökohtana ovat epälineaarinen pelaaminen ja laajat pelialueet joissa on monta eri keinoa saavuttaa asetetut tavoitteet.

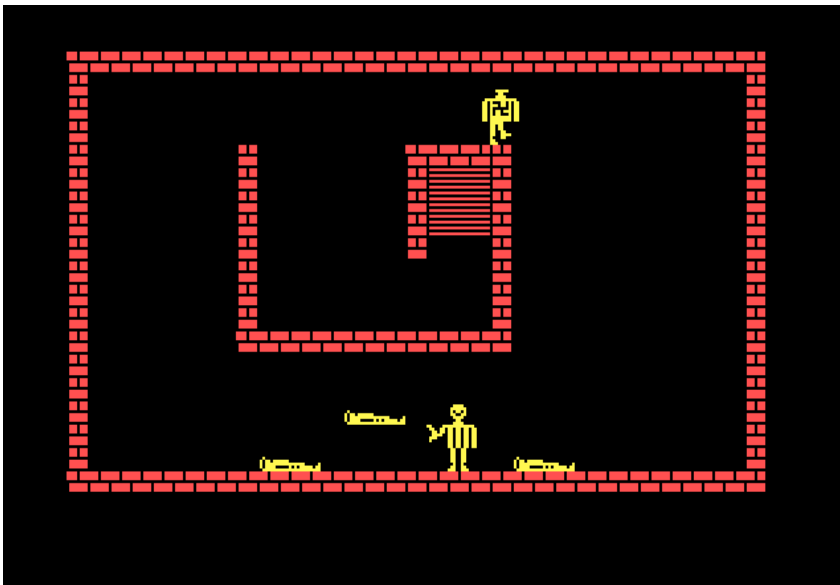
#### **3.1. Väkivaltaisten videopelien historian merkittävimmät pelit**

Ensimmäisenä väkivaltaisena videopelinä pidetään yleisesti Exidy Games -nimisen yrityksen vuonna 1976 julkaisemaa *Death Race* -kolikkopeliä [Gonzalez, 2004]. Kyseisessä ajosimulaattorissa oli tarkoituksena ajaa autolla pienten tikku-ukkojen ylitse. Kun tikku-ukkoihin osui, ne muuttuivat pieniksi ristillä varustetuiksi hautakiviksi (ks. kuva 1). Peli sai valtavasti huomiota mediassa, minkä seurauksena pelin myynti tuplaantui.



Kuva 1. Death Race [Gonzalez, 2004]

Seuraava merkittävä väkivaltainen videopeli oli vuonna 1981 Muse-nimisen yrityksen julkaisema *Castle of Wolfenstein*. Pelin tarkoituksena oli selviytyä läpi lukuisista huoneista ja sokkeloista tappaen samalla natsisymboleilla merkittyjä vihollisia (ks. kuva 2). Peli julkaistiin useille eri alustoille kuten Apple II, MS-DOS, Atari 400/600 ja Commodore 64.



Kuva 2. Castle of Wolfenstein [Gonzalez, 2004]

Id Software ja Apogee Software julkaisivat vuonna 1992 *Wolfenstein 3D*-pelin (ks. kuva 3). Pelin voidaan sanoa olevan jatko-osa *Castle of Wolfenstein* pelille ja sen jatko-osalle *Beyond Castle Wolfensteinille* [Wikipedia, 2010b]. *Wolfenstein 3D* -pelin katsotaan yleisesti synnyttäneen ensimmäisen persoonan am-

muntapeli -genren sekä tuoneen sen tietokonepelaajien suosioon [Wikipedia, 2010b]. *Wolfenstein 3D* oli myös erittäin väkivaltainen peli [Gonzalez, 2004].

Pelissä seikkailtiin amerikkalaisella sotilaalla nimeltään William "B.J." Blazkowicz, joka yritti paeta natsien tukikohdasta nimeltään Castle Wolfenstein. Pelin ideana oli tutkia huoneita löytäen erilaisia aarteita, ensiapupakkauksia ja ammuksia samalla tappaen natsseja. Pelissä veri lensi liioitelluissa kaarissa, ja kuolleet viholliset jäivät lattialle makaamaan omaan verilammikkoonsa.

Valtavan suosion saanut *Wolfenstein 3D* julkaistiin myöhemmin myös useilla muilla pelikonsoleilla ja alustoilla. Pelissä seinillä olevien hakaristien sekä väkivaltaisen sisällön takia *Wolfenstein 3D* on kielletty Saksassa.



Kuva 3. *Wolfenstein 3D*

*Mortal Kombat* julkaistiin vuonna 1992 aluksi pelihalleihin ja myöhemmin myös kotikonsoleille. *Mortal Kombat* on klassinen tappelupeli, jossa pelaaja voi otella tietokoneen ohjaamaa vihollista tai ihmispelaajaa vastaan. Peli kuitenkin erottui massasta sen graafisen ilmeen ja teemansa takia. Gonzalezin [2004] mukaan *Mortal Kombat*issa oli kaikkein isoimmat ja realistisimman näköiset pelihahmot mitä videopeleissä oli siihen mennessä nähty. Pelaaja kirjaimellisesti katseli digitaalisesti animoituja valokuvia ihmistä hyppimässä ja hakkaamassa toisiaan kuolemaan asti. Taistelussa verta lensi litroittain, kunnes lopulta vastustaja viimeisteltiin erityisellä Fatality-lopetusliikkeellä (ks. kuva 4). Fatality tarkoitti usein vihollisen raajojen, pään tai sydämen repimistä irti ruumiista mahdollisimman raa'alla tavalla. Gonzales [2004] kertoo, kuinka peli ei pelkästään shokeerannut pelaajiaan, vaan jäi myös raakuutensa ansiosta välittömästi kaikkien niiden mieliin, jotka olivat edes kävelleet pelikoneen ohitse.



Kuva 4. Mortal Kombat ja Fatality-lopetusliike

*Mortal Kombat*in suosio ja väkivaltainen sisältö alkoivat saada huomiota myös vanhemmilta ja lainsäätäjiltä. Vuonna 1993 Yhdysvaltain kongressi otti väkivaltaisten videopelien myynnin alaikäisille käsittelyynsä [Kohler, 2009]. Tarkoituksena oli saada videopeleille yhtenäiset ikäraja järjestelmät. Vuoden 1994 helmikuussa tehdyssä lakialoitteessa videopeliteollisuudelle annettiin yksi vuosi aikaa kehittää kyseinen järjestelmä tai muuten sen kehittäisi valtio. Suuret pelien kehittäjät kuten Nintendo, Sega, Electronic Arts ja Acclaim päättivät yhdessä luoda heidän etujaan Washingtonissa edustavan järjestön. Järjestön nimeksi tuli *Interactive Digital Software Association* (IDSA), joka sai myöhemmin nimekseen *Entertainment Software Association* (ESA). Syyskuussa 1994 Yhdysvaltain kongressi hyväksyi IDSA:n esittämän *Entertainment Software Ratings Board* (ESRB)-ikäraja järjestelmän. ESRB:stä tuli varoitusjärjestelmä vanhemmille, jotka halusivat ostaa lapsilleen heidän ikänsä mukaisia pelejä [Gonzalez, 2004]. Gonzalezin [2004] mukaan ESRB:n luominen oli hyvä idea peliteollisuudelta. Sen lisäksi, että valtio ei päässyt säätelemään ikärajalukuituksen sisältöä, ESRB myös uudisti käsityksen siitä, että pelit eivät ole tarkoitettu vain lapsille ja että kaikki pelit eivät sovi kaikille. Se myös antoi vanhemmille arvokkaan työkalun ymmärtää pelien sisältöä ja tehdä päätöksiä niiden sopivuudesta lapsilensa.

Väkivaltaisten videopelien ollessa median päällimmäisinä puheenaiheina ja kongressin käsittelyssä vuonna 1993 Id Software julkaisi *Doom*-nimisen ensimmäisen persoonan ammuskelpelin tietokoneille. Vakava, nopeatempoinen ja jännittävä *Doom* saavutti välittömästi pelaajien suosion [Gonzalez, 2004]. Pelissä oli syvyyttä ja se oli erityisen hyvin toteutettu (ks. kuva 5). Pelaaja pääsi oh-



jaamaan Mars-planeetalle ansaan jäänyttä avaruusjalkaväen sotilasta, jonka ainoa toivo pelastua oli ampua tiensä loputtomien hirviö- ja demonilaumojen läpi. *Doomin* grafiikka ja interaktiivinen väkivalta sekä sen saatanalliset teemat tekivät pelistä huomattavan kiistelyn kohteen [Wikipedia, 2010c]. *Doom* esitteli tietokonepelaajille myös uudenlaisen pelimuodon, moninpelin. Moninpelissä pelaajat pystyivät tietokoneen ohjaamien vihollisten sijaan kamppailemaan keskenään internetissä. Tietokone- ja videopeliihteitä käsittelevän Gamespy-sivuston vuonna 2004 tekemässä äänestyksessä *Doom* sijoittui ensimmäiseksi saaden tittelin ”kaikkien aikojen paras peli” [Gamespy, 2004].



Kuva 5. Doom

*Doom* liitettiin myös Yhdysvalloissa huhtikuussa 1999 tapahtuneeseen Columbinen kouluammuskeluun. Ammuskelijat Eric Harris ja Dylan Klebold olivat kumpikin olleet suuria *Doom*-pelin faneja [Gonzalez, 2004]. Ammuskelun uhrien omaiset, koulun opiskelijat ja opettajat haastoivat useita videopeliyrityksiä oikeuteen vaatien yhteensä viiden miljardin korvauksia. Tämä ja useat muut videopeliyrityksiä vastaan nostetut haasteet kuitenkin hylättiin oikeudessa. Yhtä tapauksista käsitellyt tuomari Danny Boggs kommentoi asiasta seuraavasti: ”Näemme, että videopelihahmojen ampumisesta tv-ruudulla (miljoonien harrastama aktiviteetti) on yksinkertaisesti liian pitkä matka ryhtyä oikeasti ampumaan ihmisiä luokkahuoneessa” [Gonzalez, 2004].

Vuonna 2001 Rockstar Games -nimisen yrityksen Playstation 2 -konsolille julkaiseman *Grand Theft Auto* -sarjan kolmannen osan sanotaan aloittaneen kokonaan uusi avoimen pelimaailman-peligenre ja samalla tehneen lähtemättömän vaikutuksen videopelien maailmaan [Geddes, 2007]. *Grand Theft Auto III* -pelissä pelaaja ohjaa pikkurikollista jonka tarkoituksena on nousta kuvitteelli-

sen Liberty City -kaupungin rikollismaailman huipulle suorittaen tehtäviä eri rikollisjärjestöille. Peli yhdistelee autolla ajamista ja kolmannen persoonan ammuskelua laajassa vapaasti tutkittavassa 3D-maailmassa (ks. kuva 6). *Grand Theft Auto III* -peliä on myyty maailmanlaajuisesti 14,5 miljoonaa kappaletta ja se on voittanut useita palkintoja eri pelisivustoilta ja lehdiltä. Väkivaltaisen sisältönsä vuoksi peli on saanut myös hyvin paljon negatiivista huomiota osakseen. Pelissä pelaaja pystyy muun muassa varastamaan autoja, tappamaan sivullisia sekä poliiseja ja harrastamaan seksiä prostituoidun kanssa. Tämän takia peli onkin kielletty melkein kaikkialla maailmassa alle 18-vuotiailta.



Kuva 6. Grand Theft Auto III

### 3.2. Pelien ikäraajat

Tässä kohdassa esitellään Euroopassa käytössä oleva videopelien ikärajojen luokitusjärjestelmä PEGI. Yhdysvalloissa käytössä oleva ikärajojärjestelmä ESRB ja Japanissa käytössä oleva CERO ovat sisällöltään hyvin samantapaisia PEGI:n kanssa ja tämän takia niitä ei esitellä erikseen.

PEGI on lyhenne sanoista Pan European Game Information. Se auttaa eurooppalaisia vanhempia tekemään tietoon perustuvia päätöksiä, kun he ostavat tietokonepelejä. PEGI otettiin käyttöön keväällä 2003, ja sen jälkeen monia kansallisia ikäluokitusjärjestelmiä on korvattu tällä yhteisellä järjestelmällä, jota käytetään nykyään suurimmassa osassa Eurooppaa kaikkiaan 30 maassa, myös Suomessa. [Pegi, 2010]

Pelit on merkitty pakkauksen etu- ja takakannessa PEGI-merkinnällä (ks. kuva 7), joka tarkoittaa ikärajaa 3, 7, 12, 16 tai 18. Merkintä antaa luotettavaa tietoa pelin sisällön soveltuvuudesta alaikäisille. Luokitus ei kerro pelin vaikeudesta tai sen pelaamiseen tarvittavista taidoista.



Kuva 7. PEGI-merkinnät

PEGI [2010] on määritellyt PEGI-18 merkinnän seuraavasti: Täysi-ikäisyyden luokitusmerkintää käytetään, kun pelissä esitetty väkivalta on luonteeltaan törkeää ja/tai tietyn tyyppistä. Törkeää väkivaltaa on vaikea määritellä, sillä näkemykset voivat monesti olla hyvin erilaisia, mutta yleisesti katsoen se voidaan määritellä väkivallaksi, joka herättää katsojassa vastenmielisyyttä.

Pakkauksen kyljessä esitettävät sisältösymbolit kertovat tärkeimmät syyt, joiden takia peli on saanut tietyn luokituksen. Sisältösymboleja on kahdeksan ja ne on esitelty kuvassa 8.



Kuva 8. PEGIn sisältösymbolit

## 4. Videopelitutkimuksia

Tässä luvussa esitellään ja analysoidaan neljä 2000-luvulla tehtyä tutkimusta väkivaltaisista videopeleistä. Ensimmäiset kaksi tutkimusta kertovat löytäneensä linkin väkivaltaisten videopelien ja aggressiivisen ja väkivaltaisen käyttäytymisen välillä. Jälkimmäiset kaksi tutkimusta taas eivät päätyneet samanlaisiin tuloksiin. Tutkimus kerrallaan esitellään ensin niissä käytettyjä tutkimusmenetelmiä, minkä jälkeen käydään läpi tutkimuksen tulokset ja lopulta analysoidaan tarkemmin tutkimusmenetelmien vaikutusta tutkimuksessa raportoituihin tuloksiin.

### 4.1. Bartholow ja muut

Bartholow ja muita [2006] tutkimukseen osallistua 39 yleistervettä, miespuolista ylioppilasta, joiden ikien keskiarvo oli 19,5 vuotta. Osallistujat olivat kaikki oikeakätisiä ja heidän näkökykynsä oli normaali. Tutkimukseen osallistuneille henkilöille tehtiin kysely, jonka pohjalta pystyttiin määrittelemään, ketkä heistä pelasivat väkivaltaisia videopelejä ja kuinka usein he niitä pelasivat.

Osallistujille ei kerrottu tutkimuksen varsinaista tarkoitusta, vaan he luulivat osallistuvansa kuvien katseluun liittyvään reaktionopeuden testaukseen. Jokaiselle osallistujalle näytettiin satunnaisesti neutraaleja, väkivaltaisia ja väkivallattomia, mutta sävyiltään negatiivisia kuvia, minkä aikana heidän aivojensa ERP (*event-related brain potentia*) eli tapahtumasidonnainen herätepotentiaali mitattiin. ERP syntyy aivoissa, kun ulkoinen heräte aiheuttaa aivosähköisen muutoksen. ERP-arvoista pystytään selvittämään osallistujien P300-aaltojen arvot, jotka kertovat siitä, onko jokin asia tuttu heille. Mitä suurempi P300-aallon arvo on, sitä odottamattomamman havainnon hän on tehnyt. Tämän jälkeen osallistujille suoritettiin kilpailuhenkinen testi, jossa heidän annettiin luulla kilpailevansa toista tutkimuksen osallistujaa vastaan. Testissä oli tarkoituksena painaa painiketta äänimerkin jälkeen toista kilpailijaa nopeammin. Osallistujille kerrottiin myös, että testin hävinnyt osapuoli tulisi saamaan kuulokkeisiinsa "äänipamauksen", jonka voimakkuuden ja keston he olivat saaneet itse määrittää ennen jokaista kilpailuhenkistä testiä. Äänipamauksen kesto määriteltiin alkaen 0,25 (taso 1) sekuntista 2,5 (taso 10) sekuntiin asti ja äänenvoimakkuus alkaen 60 (taso 1) desibelistä 105 (taso 10) desibeliin asti. Osallistujille mainittiin myös mahdollisuudesta käyttää ei-aggressiivista äänetöntä asetusta (taso 0).

Tutkijat mittasivat osallistujien aggressiota laskemalla keskiarvon heidän käyttämistään äänipamausten kestoista ja äänenvoimakkuuksista. Sekä kuvien näyttö että kilpailuhenkinen testi toistettiin 25 kertaa kullekin osallistujalle. Kilpailuhenkisen testin 25 erän voittoja ja häviötä ohjaili oikeasti tietokone, joka asetti satunnaisilla arvoilla osallistujalle häviöstä koituvan äänipamauksen voi-

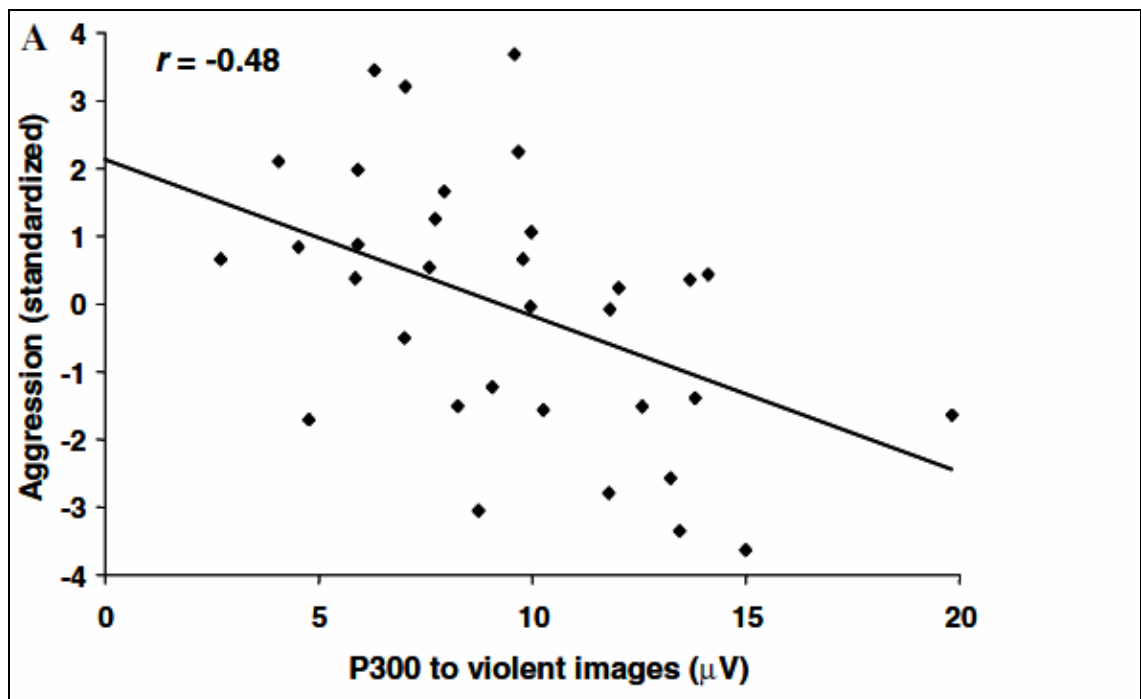
makkuuden ja keston. Kilpailuhenkinen testi oli tehty niin, että jokainen kilpailija hävisi aina ensimmäisen erän ja loppujen 24 erän tulokset tietokone päätti satunnaisesti.

#### 4.1.1. Tutkimuksen tulokset

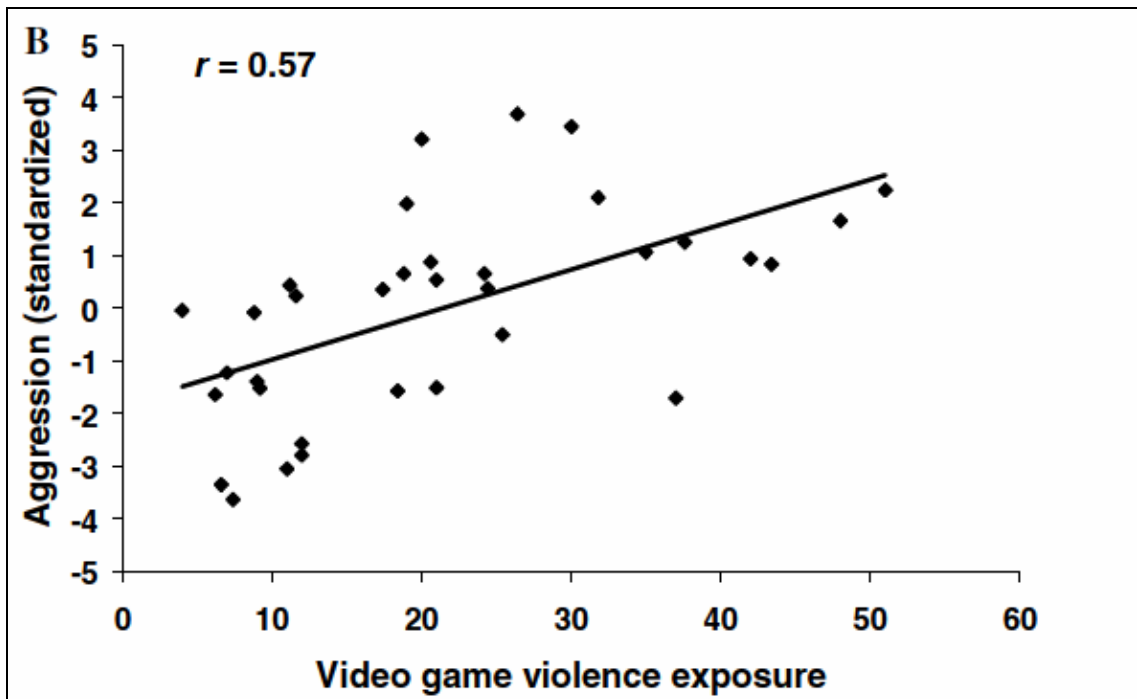
Tutkimukseen osallistuneiden henkilöiden ERP-arvoista mitattujen P300-aaltojen perusteella tutkijat väittävät, että väkivaltaisista videopelejä pelanneiden henkilöiden aivoilla kesti kauemmin tunnistaa ja kategorisoida testissä näytetyt väkivaltaiset kuvat. Tutkijoiden mukaan tämä tarkoittaa sitä, että pitkäaikainen altistuminen väkivaltaisille videopeleille voi johtaa heikentyneeseen kykyyn hallita aggressiota ja vähentää henkilöiden reagoitua väkivallan uhrien tuskaan ja kärsimykseen.

Kuvan 9 kuviosta nähdään, kuinka ERP-arvoista mitatut P300- arvot ovat pienempiä niillä henkilöillä, joilla mitattiin suurempia aggression määriä. Kuvasta 10 puolestaan nähdään, kuinka henkilöillä, jotka ovat altistuneet enemmän väkivaltaisille videopeleille, on mitattu myös suuremmat aggression määrät.

Yhteenvedonä tutkijat pitävät tätä tutkimusta ensimmäisenä, joka yhdistää videopeliväkivallalle altistumisen ja aggressiivisen käyttäytymisen aivojen prosesseihin.



Kuva 9. Aggressiivisen käyttäytymisen ja P300-arvojen yhteys [Bartholow et al., 2006].



Kuva 10. Aggressiivisen käyttäytymisen ja väkivaltaisille peleille altistumisen yhteys [Bartholow et al., 2006].

#### 4.1.2. Tutkimuksen analysointia

Bartholow ja muut [2006] väittävät tutkimuksensa todistavan, kuinka väkivaltaisten videopelien pelaaminen tekee pelaajistaan väkivallalle turtuneita ja mahdollisesti aiheuttaa heissä väkivaltaista käytöstä. Tutkimuksen tuloksista voidaan kuitenkin varmuudella päätellä vain se, että väkivaltaisia videopelejä pelanneiden henkilöiden aivot reagoivat hitaammin heille näytettyihin väkivaltaisiin kuviin. Tutkimuksesta voidaan myös todeta korrelaatio, että väkivaltaiseen käytökseen taipuvaiset henkilöt pelaavat mielellään väkivaltaisia videopelejä. Tutkimuksen alussa tehdyssä pohjatietolomakkeessa osallistujien itse arvioimat videopelit arvosanoineen jätettiin kertomatta. Näin lukijalle jää epäselväksi, minkälaisia pelejä osallistujat olivat aikaisemmin pelanneet ja minkälaisena he niiden sisällön kokivat. Tutkimuksessa tehdyssä pohjatietokyselyssä ei myöskään huomioitu mitenkään osallistujien muita harrastuksia tai television katselua. Aggressiivisen käyttäytyminen testissä voisi selittyä yhtä hyvin myös osallistujan katsomista väkivaltaisista televisiosarjoista tai vaikkapa jonkin kontaktilajin, kuten jääkiekon, harrastamisella.

Äänipamauksen keston ja äänenvoimakkuuden pitäminen aggressiivisuuden mittarina tässä tutkimuksessa ei ole mielestäni toimiva ratkaisu. Ensinnäkin testiin osallistuneille ei kerrota missään vaiheessa, että äänipamauksen antaminen nähdään aggressiivisena toimenpiteenä osallistujalta ja tämän takia osallistuja voi luulla sen olevan osa testin tarkoitusta ja näin ollen myös hyväksyttävää käytöstä testin aikana. Toiseksi äänipamauksen antaminen ei välttä-



mättä johdu osallistujan väkivaltaisesta videopelitaustasta, vaan voi olla myös vastatoimenpide tietokoneen arpomaan satunnaisen häviön äänipamaukseen osallistujalle.

Huomattavaa oli myös se, että kyseisessä tutkimuksessa ei ollut käytetty ol- lenkaan kontrolliryhmänä esimerkiksi ihmisiä, jotka eivät ole pelanneet kos- kaan tai pelaavat todella vähän videopelejä. Tutkimukseen ei myöskään osallis- tunut yhtään naispuolista henkilöä. Kuvan 10 mukaan jokainen testiin osallis- tunut henkilö oli ollut jonkin verran tekemisissä väkivaltaisten videopelien kanssa.

## **4.2. Uhlmann ja Swanson**

Uhlmanin ja Swansonin [2003] tutkimuksen tarkoituksena on tarkastella suh- detta väkivaltaiselle medialle altistumisen ja alitajuisten minäkuvien muodos- tumisen välillä. Tutkimukseen osallistui 121 aloittelevaa psykologian opiskeli- jaa (54 miestä, 65 naista, 2 ei tiedossa) vapaaehtoisesti kurssipisteitä vastaan. Kaikki osallistujat olivat 18 vuotta täyttäneitä. Osallistujat pelasivat kymmenen minuutin ajan joko väkivaltaista 3D ammuskelupeliä *Doomia* tai väkivallatonta puzzle-peliä nimeltään *Mahjongg: Clicks*. Pelaamisen jälkeen osallistujat suorit- tivat IAT-testin (eng. *Implicit Association Test*), jonka tarkoituksena on mitata ali- tajuista aggressiivista minäkäsitystä. IAT-testissä järjestellään nopeasti katego- rioihin useita erilaisia implisiittisiä assosiaatioita (kuvia, sanoja) niin, että nope- ammat objektien pariutukset tulkitaan olevan vahvempina henkilön muistissa kuin vaikeammat ja hitaammat objektien pariutukset. Tässä tutkimuksessa osallistujat järjestelivät sanoja kuten aggressiivinen, taistelu, rauhallinen tai hel- lä kategorioihin kuten minä, muut, aggressiivinen ja rauhallinen. Testin jälkeen osallistujat suorittavat myös erilaisia kyselyitä, joissa selvitettiin mm. heidän käsityksiään omasta aggressiivisuudestaan, videopelien pelaamisen määrää ja väkivaltaisten videopelien pelaamisen määrää.

### **4.2.1. Tutkimuksen tulokset**

Tutkimuksessa havaittiin, että osallistujat, jotka pelasivat väkivaltaista *Doom*- ammuskelupeliä 10 minuutin ajan, yhdistivät minuutensa (kategoria: minä) IAT-testissä helpommin aggressiivisiin piirteisiin ja toimintoihin. Kyseiset hen- kilöt eivät kuitenkaan yhdistäneet itseään aggressiivisiin piirteisiin erilaisissa tutkimuksen itsearviointitehtävissä. Väkivaltaista *Doom*-peliä pelanneet henki- löt eivät myöskään arvioineet itseään aggressiiviseksi pelin pelaamisen jälkeen. Näiden löytöjen perusteella tutkimus väittää, että vahvimmat vaikutukset ly- hytaikaisesta altistumisesta väkivaltaiselle videopelille aiheutuivat juuri henki- lön alitajuiseen minäkuvaan.

Tutkijoiden mukaan väkivaltaisille videopeleille altistumisen pitkäaikaisia vaikutuksia alitajuiseen minäkäsitykseen tuki myös se, että osallistujat, jotka olivat kertoneet pelanneensa aikaisemmin väkivaltaisia videopelejä, yhdistivät IAT-testissä minuutensa useammin aggressiivisuuteen.

#### **4.2.2. Tutkimuksen analysointia**

Mielestäni tutkimuksessa käytetty IAT-testi ja saadut tulokset eivät vielä yksinään todista, että väkivaltaiset videopelit voisivat vaikuttaa suoraan henkilöiden minäkuvaan ja alitajuisiin mieltymyksiin. Tutkimus pystyi osoittamaan sen, että väkivaltaisia videopelejä aikaisemmin tai tutkimuksen aikana pelannut henkilö yhdistää IAT-testissä käsitteen minä nopeammin aggressiivisiin sanoihin kuin rauhallisiin sanoihin. Käsitteiden ja sanojen yhdistämisen nopeudella ei kuitenkaan vielä ole välttämättä mitään tekemistä todellisen aggressiivisen käyttäytymisen kanssa.

IAT-testiä on arvosteltu siitä, ettei pelkkien testissä laskettujen millisekuntien avulla pystytä vielä arvioimaan niinkin havaitsematonta ulottuvuutta kuin henkilöiden alitajuisia mieltymyksiä [Blanton and Jaccard, 2006]. Blanton ja Jaccard [2006] väittävät, ettei IAT-testillä ole tukenaan tarpeeksi empiiristä tutkimusta todistamaan sen kelvollisuutta kunnollisten diagnoosien tekemisessä. Heidän mukaansa kukaan ei pysty määrittelemään IAT-testin nollapisteen sijoittuvan juuri todelliselle neutraalille mieltymyksen kohdalle, eikä myöskään sitä, kuinka eroavaisuudet nollapisteestä voidaan muuttaa vastaamaan asteikkoja todellisista mieltymyksien eroista.

#### **4.3. Ferguson ja muut**

Ferguson ja muut [2007] selvittivät kahden eri menetelmän avulla, aiheuttavatko väkivaltaiset videopelit suoranaisesti aggressiota pelaajissaan vai onko suhde videopelien ja aggression välillä paremmin selitettävissä sivutuotteena ”kolmannen osapuolen” muuttujista kuten altistumisesta perheväkivallalle tai synnynäisistä väkivaltaisista piirteistä.

Tutkimuksen ensimmäisessä osassa tutkittiin väkivaltaisten videopelien vaikutusta pelaajiin laboratorio-olosuhteissa. Tutkimukseen osallistui yhteensä 101 vapaaehtoista opiskelijaa Texasin ja Wisconsinin yliopistoista. Osallistujien keski-ikä oli 20,9 vuotta ja heistä 46 oli miehiä ja loput 55 naisia. Osallistujien tuli täyttää aluksi heidän luoteenomaista aggressiota mittaava Aggression Questionnaire -lomake. Tämän jälkeen osallistujat arvioivat heidän omia videopelitottumuksiaan, pelaamisen määrää sekä luettelemaan viisi eniten pelaamaansa peliä ja arvostelemaan niiden sisällön väkivaltaisuutta. Osallistujilta



kysyttiin nykyisen pelaamisen määrän lisäksi myös sitä, kuinka paljon he olivat arviolta pelanneet videopelejä ollessaan ala- ja yläasteella.

Seuraavaksi osallistujat jaettiin satunnaisesti kolmeen eri ryhmään. Yksi ryhmä pelasi väkivaltaista ensimmäisen persoonan ammuskelupeliä nimeltään *Medal of Honor: Allied Assault*, toinen ryhmä pelasi väkivallatonta ensimmäisen persoonan seikkailupeliä nimeltään *Myst III: Exile* ja kolmas ryhmä sai päättää pelaamansa pelin lyhyen peliesittelyn perusteella. Jokainen osallistuja sai pelata peliänsä 45 minuutin ajan, jonka jälkeen he siirtyivät suorittamaan reaktiotestiä.

Tutkimuksessa mitattiin pelatun videopelin vaikutusta osallistujien aggressiivisuuteen käyttämällä standardoitua ja näin ollen luotettavampaa versiota Bartholowin ja muiden [2006] tutkimuksessa käytetystä äänipamausreaktiotestistä. Osallistujille kerrottiin, että he kilpailivat 25 erää toista tutkimukseen osallistuvaa henkilöä vastaan. Osallistujien tarkoituksena oli painaa reaktiotestin painiketta äänimerkin jälkeen toista osallistujaa nopeammin. Heille kerrottiin myös, että testin hävinnyt osapuoli tulisi saamaan kuulokkeisiinsa ”äänipamauksen”, jonka äänenvoimakkuuden he olivat itse saaneet määrittää ennen jokaista reaktiotestin erää. Äänipamauksen äänenvoimakkuudella mitattiin osallistujien aggressiota testin aikana. Äänipamauksen keston määrittäminen oli jätetty tästä tutkimuksesta kokonaan pois, koska se ei ollut osoittautunut hyväksi aggression mittariksi muissa vastaavissa tutkimuksissa. Reaktiotestin voittojen ja häviöiden määrä oli ennalta asetettu ja standardoitu jokaiselle osallistujalle.

Lopuksi osallistujat vastasivat vielä kysymyksiin heidän tutkimuksessa pelaamastaan videopelistä.

Toisen tutkimuksen tarkoituksena oli tarkastella ennustavaa suhdetta perheväkivallalle altistumisen, persoonallisuuden ja väkivaltaisen luoteen, videopelitottumuksien ja tehtyjen väkivaltaisten rikosten välillä. Tutkimus oli erityisesti suunniteltu paljastamaan, ennustavatko videopelitottumukset väkivaltaista käyttäytymistä, kun perheväkivallalle altistuminen oli otettu huomioon.

Tähän tutkimuksen osaan osallistui 428 opiskelijaa Floridan yliopistosta. Osallistujien keski-ikä oli 20,68 vuotta ja heistä 173 oli miehiä ja loput 255 oli naisia.

Tässäkin tutkimuksen osassa osallistujien tuli aluksi täyttää aluksi heidän luonteenomaista aggressiota mittaava Aggression Questionnaire -lomake. Tämän jälkeen osallistujat arvioivat omia videopelitottumuksiaan ja väkivaltaisille videopeleille altistumisen määrää. Osallistujat vastasivat myös Family Conflict Scale (FCS)-kyselyyn, jonka 49 kysymyksellä selvitetään altistumista perheväkivallalle sen eri muodoissa. Lopulta osallistujia pyydettiin vielä arvioimaan,

kuinka monta väkivallatonta ja väkivaltaista rikosta he olivat suorittaneet viimeisen vuoden aikana.

#### **4.3.1. Tutkimuksen tulokset**

Tutkimuksen ensimmäisessä osassa havaittiin, että väkivaltaiset pelit eivät aiheuta laboratorio-olosuhteissa lyhytkestoista aggressiota sen enempää kuin väkivallattomat pelit. Osallistujille tehtyjen videopelihistoriaa kartoittavien kyselyiden perusteella pitkäaikainen väkivaltaisille videopeleille altistuminen ei myöskään tuottanut eroavaisuuksia mitatuissa aggression tasoissa.

Tutkimuksen jälkimmäisessä osassa havaittiin, että miespuolisilla osallistujilla oli suurimmat mitatut aggression määrät. Fyysinen ja verbaalinen pahoinpitely perheessä havaittiin ennustaviksi tekijöiksi aggression määrän kasvuille. Muut muuttujat, kuten altistuminen väkivaltaisille videopeleille, eivät vaikuttaneet lisäävästi aggression määrään. Kokonaisuudessa tutkimuksen tulokset osoittavat sen, että luonteenomaiset väkivaltaiset piirteet sekä altistuminen fyysiselle pahoinpitelylle ovat parhaat ennustavat tekijät henkilöiden tekemille väkivaltaisille rikoksille. Tutkimuksen perusteella voidaan sanoa, että väkivaltaisen käyttäytymisen ja väkivaltaisten videopelien pelaamisen välillä ei ole kausaalista eikä korreloivaa yhteyttä.

#### **4.3.2. Tutkimuksen analysointi**

Fergusonin ja muiden [2007] tekemässä tutkimuksessa näytetään toteen ainakin se, ettei äänipamaus-reaktiotestiä käyttämällä saada eroavia tuloksia väkivaltaista ja väkivallatonta peliä pelanneiden ryhmien välillä. Tutkimukseen osallistui tasapuolisesti sekä miehiä että naisia, ja reaktiotesti-osuudella käytössä oli selkeä kontrolliryhmä (henkilöt, jotka pelasivat arvotusti väkivallatonta peliä). Tutkimuksen ensimmäisessä osassa on myös suurempi osallistujamäärä kuin Bartolowin ja muiden [2006] vastaavilla menetelmillä suorittamassa tutkimuksessa. Tutkimuksen toinen osio myös tukee tuloksillaan tutkimuksen ensimmäisen osan havaintoja. Tutkimuksen toiseen osioon on osallistunut tarpeeksi suuri määrä testihenkilöitä ja sen tuloksia voidaan pitää luotettavina.

#### **4.4. Ermi ja muut**

Ermin ja muiden [2004] tutkimuksen tarkoituksena oli tarkentaa kuvaamme peleistä, pelaajista, pelaamisen syistä, merkityksistä ja pelikokemuksen luonteesta ja samalla kartoittaa tavallisten suomalaiskotien arkea modernin pelikulttuurin parissa. Tutkimuksen lähtökohtana on näkemys lapsista aktiivisina pelikulttuurien toimijoina ja siinä on pyritty selvittämään, mitä ja miten lapset ja nuoret pelaavat, sekä miten he omaa pelaamistaan jäsentävät. Tutkimuksessa on eri-

tyisesti pyritty saamaan esille pelaajien oma ääni sekä henkilökohtainen mediasuhde.

Tutkimukseen osallistui 284 lasta ja heidän vanhempansa. Osallistuneista lapsista tyttöjä oli 156 ja poikia 128. Noin puolet osallistuneista lapsista oli iältään 11-vuotiaita ja loput 10- tai 12-vuotiaita. Tutkimukseen osallistuneet lapset ja heidän vanhempansa täyttivät pelaamista ja siihen liittyviä asioita käsittelevän kyselylomakkeen. Tämän jälkeen kyselyihin vastanneista valittiin 15 perhettä, joihin otettiin yhteyttä ja käytiin tekemässä haastattelu. Käytännössä haastatteluihin osallistui yhteensä 16 lasta ja 16 vanhempaa, koska yhdessä perheessä kohderyhmään kuuluvia lapsia oli kaksi ja yhdessä perheessä haastatteluun osallistuivat molemmat vanhemmat. Haastatelluista vanhemmista valtaosa oli äitejä, sillä haastatteluihin osallistui yhteensä 13 äitiä ja kolme isää. Myös kyselyaineistossa suuri osa vastanneista vanhemmista oli lasten äitejä.

Haastattelut toteutettiin teemahaastatteluina perheiden kodeissa. Haastatteluissa keskusteltiin muun muassa suosikkipeleistä, yhdessä ja yksin pelaamisesta, pelaamisen hyödyistä ja haitoista, perheen säännöistä ja käytännöistä sekä peliväkivallasta. Lasten ja vanhempien haastatteluissa käsiteltiin pääosin samoja teema-alueita, mutta vanhempien haastatteluissa nostettiin lisäksi esiin yleisempiä mediakasvatukseen liittyviä aiheita sekä selvitettiin vanhempien näkemyksiä ja tietämystä digitaalisista peleistä.

#### **4.4.1. Tutkimuksen tulokset**

Tutkimuksen ja haastattelun perusteella saatiin selville hyvin paljon peleihin liittyviä merkityksiä ja käytäntöjä osallistuneiden lasten ja perheiden arjessa. Tässä kappaleessa kuitenkin tarkastellaan vain tutkimuksen alueita jotka käsittelevät videopelien vaikutuksia lapsiin heidän omista ja vanhempien näkökulmista sekä väkivaltaisten videopelien pelaamista.

Pelaamiseen usein liittyvä nopeatempoisuus sai kritiikkiä vanhemmilta: pelaamisesta noussut jännitys ja innostus herättivät vanhemmissa huolta ja ärtymistä, kun lapsi vaikutti levottomalta ja tämä levottomuus tuntui jatkuvan pitkään vielä pelaamisen loputtuakin. Varsinkin kilpailulliset moninpelit ja toiminnalliset autoilu- sekä taistelupelit koettiin intensiteettinsä vuoksi monissa perheissä rasittaviksi. Osaltaan tästäkin syystä rauhallisempaan puuhasteluun keskittyneet pelit, kuten opetuspelit, olivat vanhemmista houkuttelevampia.

Haastattelun aikana vanhemmat toivat esille huolestumisensa peliväkivallan vaikutuksista lapsiin. Tämä huolestuminen ei välttämättä pohjautu vanhempien omiin havaintoihin ja kokemuksiin siitä, että väkivaltaiset pelit olisivat lapsille haitallisia, vaan esimerkiksi huhupuheena leviäviin varoitustarinoihin tai mediassa esiintyneisiin näyttävästi otsikoituihin

juttuihin. Tunnetasolla pelkät mahdolliset yhteydet peliväkivallan ja väkivaltaisen käyttäytymisen välillä riittävät siihen, että väkivaltaa sisältävät pelit koetaan ilmiönä vaaralliseksi ja uhkaavaksi.

Keskeisin huolenaihe useilla vanhemmilla oli pelko väkivaltaan turtumisesta ja jonkinlaisesta mediavälitteisestä mallioppimisesta, jossa pelistä saadut aggressiivisen käyttäytymisen mallit siirtyisivät suoraan lasten käytökseen arjessa. Kysyttäessä vanhemmilta, olivatko he huomanneet lapsissaan piirteitä väkivaltaisuudesta aggressiivisten tai kovin toiminnallisten pelien pelaamisen jälkeen, ei viitteitä tästä kuitenkaan löytynyt.

Vanhemmat saattoivat olla myös huolestuneita väkivaltaisten pelien pelaamisesta, vaikka konkreettisia ongelmia tai edes halua väkivaltaisten pelien pelaamiseen ei lasten parissa perheessä olisikaan ollut. Lisäksi se mikä ulkopuolisen tarkkailijan, kuten lapsen vanhempien, silmin voi näyttää väkivallalta, voi lasten kokemuksessa olla lähinnä esteiden poistamista pelissä etenemisen tieltä eikä siten vertaudu esimerkiksi oikeaan tappamiseen. Etenkään erilaisten sarjakuvahahmojen tai vaikkapa kilpikonnien eliminoiminen ei lasten mielissä yhdistynyt tappamiseen tai väkivaltaan.

Aggressiivinen ja väkivaltainen toiminta peleissä sijoittui lasten mielissä enemmän fiktiivisen kuin realistisen maailman piiriin. Siten pelottavimmiksi ja väkivaltaisimmiksi yleensä koettiin ne pelit, joissa väkivalta oli toteutettu realistisimmin, esimerkiksi niin, että hahmosta vuosi verta sen loukkaantuessa tai se kuoli yhdestä ainoasta laukauksesta. Harvat haastatelluista lapsista olivat kiinnostuneita väkivallasta sinänsä, vaan haastatteluissa nousi nimenomaan esiin, miten fiktiivinen väkivalta tekee pelikokemuksesta jännittävämmän. Väkivaltaisia pelejä pelaavat lapset kertoivat, etteivät he halua niinkään nähdä järjetöntä, ihmisiä kohtaan suunnattua väkivaltaa, vaan lähinnä kamppailla hirviöitä tai muita ei-ihmishahmoisia vastustajia vastaan. Pelin tarinalla oli väkivallan tulkinnassa myös merkittävä rooli – lapset halusivat sellaisia pelejä, joissa kamppailu ja väkivalta olivat kiinteä ja perusteltu osa pelin maailmaa ja seikkailua, eikä niistä irrallinen tai päälle liimatulta tuntuva elementti.

Haastatteluiden perusteella lapset eivät olleet kiinnostuneita näkemään mieletöntä ja rajatonta väkivaltaa eivätkä yleensä nimenomaisesti tahtoneet nähdä pelissä verta tai ihmistä muistuttavien hahmojen tappamista. Lasten puheessa kuolema pelissä ei muutenkaan vertautunut todelliseen kuolemaan vaan lähinnä "elämän", yhden pelin sisäisistä resursseista, menettämiseen. Tämä viittaa siihen realiteetteja koskevaan erotteluun, jonka lapset ovat kykenevät tekemään kuvitteellisen pelimaailman ja todellisuuden välille.

Lasten puheessa pelaamiseen keskeisimmin liittyvät hallintatilanteet liittyivät turhautumisen kokemuksiin pelien parissa. Lapset olivat poikkeuksetta

kokeneet tilanteita, joissa he olivat menettäneet hermonsa digitaalisen pelin parissa. Usein nämä tilanteet liittyivät pelin hankaliin tai lähes mahdottomilta tuntuviin kohtiin tai pelilaitteen toimimattomuuteen. Varsinaisen hermostumisen sattuessaan kukaan lapsista ei ollut rikkonut pelikoneitaan, vaikka he saattoivatkin kertoa, että mieli olisi tehnyt. Kun peli tuntui liian vaikealta ja lapsia alkoi suututtaa, he yleensä sulkiivat pelikoneen ja lähtivät tekemään jotakin muuta. Samoin he toimivat myös tilanteissa, joissa peli tuntui liian ahdistavalta tai pelottavalta.

#### **4.4.2. Tutkimuksen analysointi**

Ermin ja muiden [2004] tutkimuksessa ei ole lähdetty käyttämään keinotekoisia tai todellisesta elämästä irrotettuja tutkimusasetelmia, vaan aineisto on kerätty selkeillä kyselylomakkeilla ja haastatteluilla. Lapsien haastatteluista saadut vastaukset antavat hyvän käsityksen siitä, miten lapset näkevät väkivaltaisten videopelien maailman. Lapsia voidaan pitää myös rehellisinä haastateltavina, toisin kuin aikuisia. Aikuista haastateltaessa vastaus voi olla hyvinkin kaunisteltu eikä välttämättä vastaa todellista tilannetta. Aikuisten vastausten perusteella he kokivat väkivaltaiset pelit huolestuttavina, mutta kertoivat silti, etteivät lapset käyttäytyneet väkivaltaisesti näiden pelien pelaamisen jälkeen. Lapsien vastausten perusteella syy aggressioon videopelien pelaamisen yhteydessä ei niinkään johtunut pelin väkivaltaisesta sisällöstä vaan ennemminkin pelin turhauttavasta vaikeudesta tai pelikonsolin toimimattomuudesta. Näiden tulosten perusteella voidaan siis sanoa, että tutkimuksessa ei löydetty linkkiä väkivaltaisten videopelien ja väkivaltaisen käyttäytymisen välillä.

Mielestäni haastattelututkimuksen otos olisi voinut olla ainakin kaksinkertainen nykyiseen verrattuna. Tutkimuksessa otoksen pieni koko tiedostetaan ja siitä on kommentoitu seuraavasti: "On täysin mahdollista, että otoksemme ulkopuolelle on jäänyt jokin sellainen joukko, jossa pelit ovat vakavien sosiaalisten tai psykologisten vaikeuksien polttopisteessä".

Kaiken kaikkiaan tutkimus käsittelee aihetta hyvin monesta näkökulmasta, tarpeeksi selkeästi ja puolueettomasti.

## **5. Yhteenveto**

Vaikka väkivaltaisten videopelien vaikutuksista on tehty lukuisia tutkimuksia, on silti epäselvää, mitkä niiden todelliset vaikutuksen pelaajiin ovat. Aiheuttavatko väkivaltaiset videopelit pelaajissaan aggressiota ja väkivaltaista käyttäytymistä, vai onko vain niin, että jo ennestään väkivaltaan taipuvaiset henkilöt pelaavat mielellään väkivaltaisia pelejä? Tutkimustulosten ristiriitaisuus johtuu alan tutkijoiden jakautumisesta kahteen leiriin, jotka vuorotellen julkaisevat

tutkimuksiaan aiheen puolesta ja sitä vastaan. Vaikuttaa myös siltä, että pieni joukko tutkijoita näyttää tuottavan suurimman osan niistä tutkimuksista, jotka väittävät löytäneensä linkin väkivaltaisten videopelien ja aggressiivisen käyttäytymisen välillä. Tarkasteltaessa näitä tutkimuksia on syytä pitää mielessä tämänlainen tutkimusten ”sisäsiittoisuus”.

Tutkimuksia, jotka väittävät löytäneensä linkin väkivaltaisten videopelien ja aggressiivisen käyttäytymisen välillä, on kritisoitu myös niiden käyttämistä metodeista ja tutkimustulosten oikeellisuudesta. Byron [2008] väittää, että tutkijat yleistävät tutkimuksen tuloksia liikaa ja tekevät päätelmiä aggression aiheuttajista väärin perustein. Hänen mielestään laboratorio-olosuhteissa tehtyjen tutkimusten tuloksista ei voi varmuudella päätellä, ovatko vaikutukset samat myös todellisuudessa, jossa pelejä oikeasti pelataan. Byron [2008] myös korostaa, että löydetty lyhytaikaiset vaikutukset eivät vielä todista, että väkivaltaisilla videopeleillä olisi todellisia pitkäaikaisia vaikutuksia pelaajiin.

Ermi ja muut [2004] ottavat myös kantaa väkivaltaisten videopelien tutkimukseen kertomalla, kuinka vaikutuskeskustelu on kokonaisuudessaan ongelmallinen: tiukka syy-seuraussuhteen todistusvaade johtaa helposti keinotekoisiiin ja todellisesta elämästä irrotettuihin tutkimusasetelmiin sekä äärimmäisyydestä toiseen heilahteleviin väittämiin, joissa toisaalta oletetaan fiktiivisen väkivallan vaikuttavan kaikkiin samalla tavalla tai toisaalta yhtä lailla totalisoiviin ”mediavaikutuksettomuuden” väittämiin.

Tutkimuksissa käytettyjen eri metodien ollessa kiistelyn kohteena sekä niiden tulosten ollessa hyvinkin ristiriitaisia keskenään voimme kuitenkin löytää yhden tekijän, joka kyseenalaistaa linkin väkivaltaisten videopelien ja aggressiivisen sekä väkivaltaisen käyttäytymisen välillä: tilastot. Videopelien kasvattaessa suosiotaan on nuorison tekemien rikosten määrä jatkuvasti vähentymässä. Tilastojen mukaan Yhdysvalloissa nuorison tekemien rikosten määrä saavutti huippunsa vuonna 1993 ja on ollut sen jälkeen jatkuvasti laskemassa [Puzzanchera, 2009]. Tilastojen mukaan vuosien 1994 ja 2004 välillä tehtyjen murhien, ryöstöjen ja törkeiden pahoinpitelyiden määrä laski 49 % johtaen alhaisimpiin nuorten tekemien väkivaltaisten rikosten pidätyslukuihin sitten vuoden 1980. Tilastojen mukaan myös kouluväkivalta on Yhdysvalloissa ollut jatkuvassa laskussa vuoden 1993 jälkeen [Dinkes et al., 2009].

## Viiteluettelo

[Apperley, 2006] Thomas H. Apperley, Genre and game studies: Toward a critical approach to video game genres, *Simulation & Gaming* 37, 1 (March 2006) 6-23.

- [Bartholow et al., 2006] Bruce D. Bartholow, Brad J. Bushman and Marc A. Sestir, Chronic violent video game exposure and desensitization to violence: Behavioral and event-related brain potential data. *Journal of Experimental Social Psychology* **42** (2006) 532–539.
- [Byron, 2008] Tanya Byron, Byron Review – Children and New Technology. Available: <http://publications.education.gov.uk/eOrderingDownload/DCSF-00334-2008.pdf>. Checked 25.11.2010.
- [Cumberbatch, 2004] Guy Cumberbatch, Video Violence: Villain or Victim?, A report prepared for The London Video Standards Council, 2004.
- [Dinkes et al., 2009] Rachel Dinkes, Jana Kemp, Katrina Baum, Indicators of School Crime and Safety: 2009. Available: <http://nces.ed.gov/pubs2010/2010012.pdf>. Checked 25.11.2010.
- [Ermi et al., 2004] Laura Ermi, Satu Heliö ja Frans Mäyrä, Pelien voima ja pelaamisen hallinta – Lapset ja nuoret pelikulttuurien toimijoina. Hypermedialaboratorion verkkojulkaisuja **6**, Saatavana digitaalisesti: <http://tampub.uta.fi/tup/951-44-5939-3.pdf>.
- [Ferguson et al., 2008] Christopher J. Ferguson, Stephanie M. Rueda, Amanda M. Cruz, Diana E. Ferguson, Stacey Fritz and Shawn M. Smith, Violent video games and aggression: causal relationship or by product of family violence and intrinsic violence motivation? *Criminal Justice and Behaviour* **35**, 3 (March 2008) 311-332.
- [Gamespy, 2004] Gamespy, Top 50 Games of All Time. Available: <http://archive.gamespy.com/articles/july01/top501aspe/index4.shtm>. Checked 10.11.2010.
- [Geddes, 2007] Ryan Geddes and Daemon Hatfield, IGN's top10 most influential games. Available: <http://games.ign.com/articles/840/840621p2.html>. Checked 17.11.2010.
- [Gentile, 2007] Douglas A. Gentile, Muniba Saleem and Craig A. Anderson, Public Policy and the Effects of Media Violence on Children. *Social Issues and Policy Review*, **1** (2007) 15-61.
- [Gonzalez, 2004] Lauren Gonzalez, When Two Tribes Go to War: A History of Video Game Controversy. Available: <http://www.gamespot.com/features/6090892/p-1.html> Checked 3.11.2010.
- [Guillory, 2010] Brant Guillory, Video Games, In: August E. Grant and Jennifer H. Meadows (eds.), *Communication Technology Update and Fundamentals Twelfth Edition* (2010), Focal Press, 202-213.

- [Kohler, 2009] Chris Kohler, July 29, 1994: Videogame Makers Propose Ratings Board to Congress. Available: [http://www.wired.com/thisdayintech/2009/07/dayintech\\_0729/](http://www.wired.com/thisdayintech/2009/07/dayintech_0729/). Checked 9.11.2010.
- [PEGI, 2010] PEGI - Pan European Game Information. Saatavilla: <http://www.pegi.info/fi>. Tarkastettu 17.11.2010.
- [Puzzanchera, 2009] Charles Puzzanchera, U.S. Department of Justice - Juvenile Arrests 2008. Available: <http://www.ncjrs.gov/pdffiles1/ojdp/228479.pdf>. Checked 25.11.2010.
- [Uhlmann and Swanson, 2003] Eric Uhlmann and Jane Swanson, Exposure to violent video games increases automatic aggressiveness. Available: [http://www.lionlamb.org/research\\_articles/study%203.pdf](http://www.lionlamb.org/research_articles/study%203.pdf). Checked 22.9.2009.
- [Vgchartz.com, 2010] VGChartz, Hardware Totals. [http://www.vgchartz.com/hardware\\_totals.php](http://www.vgchartz.com/hardware_totals.php). 2010. Checked 1.11.2010.
- [Wikipedia, 2010a] Wikipedia, Videopeli. Saatavilla: <http://fi.wikipedia.org/wiki/Videopeli>. Tarkistettu 2.11.2010.
- [Wikipedia, 2010b] Wikipedia, Wolfenstein 3D. Available: [http://en.wikipedia.org/wiki/Wolfenstein\\_3D](http://en.wikipedia.org/wiki/Wolfenstein_3D). Checked 8.11.2010.
- [Wikipedia, 2010c] Wikipedia, Doom. Available: [http://en.wikipedia.org/wiki/Doom\\_%28video\\_game%29](http://en.wikipedia.org/wiki/Doom_%28video_game%29). Checked 10.11.2010.



# Tietokoneavusteinen psykoterapia addiktioiden hoidossa

**Hannu Sievänen**

## **Tiivistelmä.**

Tämä tutkielma on katsaus niihin tapoihin, joilla tietokoneilla on saatu lisäarvoa psykoterapiakäytäntöihin addiktioiden hoidossa. Tutkielma pyrkii esittelemään laajasti menetelmiä, joilla tietokoneavusteista psykoterapiaa on toteutettu.

**Avainsanat ja -sanonnat:** Tietokoneavusteinen psykoterapia, addiktiot, hoitokeinot.

**CR-luokat:** J.3, J.4

## **Johdanto**

Tutkielman kohteena ovat erinäisten riippuvuussairauksien hoitoon jo käytetyt sekä myös mahdolliset tietokoneita hyödyntävät menetelmät. Riippuvuussairauksia tässä tutkielmassa käsitellään melko väljässä merkityksessä. En ota kantaa eri addiktioiden mahdollisiin biologisiin, geneettisiin tai yhteiskunnallisiin kaan juonteisiin, vaan pyrin käsittelemään niitä yksilöpsykologisesta näkökulmasta.

Määrittelen riippuvuussairauksiksi useat addiktiot, joille yhteistä on nautintoa tuottavien toimintojen toistaminen, sekä kykenemättömyys lopettaa toistamista, vaikka sillä olisi nähtäviä negatiivisia vaikutuksia henkilön elämään ja terveyteen [Kuoppasalmi et al., 2007]. Addiktioiden yhteydessä toistamiseen liittyy toistetun asian euforisuus sekä siitä seuraavat vieroitusoireet, jos toimintaa ei voida toteuttaa [Thomps, 2006].

Käsittelen myös syömishäiriöitä riippuvuuksina, vaikka voidaankin olla monta mieltä siitä, ovatko ne lopulta niitä. Kuitenkin syömishäiriöiden hermostolliset mekanismit ovat samankaltaisia kuin addiktioissa [Riva et al., 2006]. Muita keskeisiä riippuvuuksia ovat muun muassa päihteisiin, alkoholiin, tupakkaan, internetiin sekä pelaamiseen, niin uhkapelien kuin videopelienkin osalta, liittyvät oireistot. Useiden addiktioiden taustalla vaikuttavat samankaltaiset aivotoiminnan prosessit [Ruth et al., 2010]. Psykiatristen sairauksien diagnosoinnin taustalla vaikuttava tautiluokitusjärjestelmä DSM (Diagnostic and Statistical Manual of Mental Disorders) siirtyy uusimmassa julkaistavassa

tautiluokituksessaan (DSM-V) erinäisten riippuvuuksien luokittelun sijasta käyttämään yläkäsitettä addiktiot, joten määrittelylleni on myös olemassa perusta.

Riippuvuuksia on pidetty aikakautemme ongelmallisina sairauksina, sillä useat henkilöt yhteiskunnassa kärsivät addiktioista ja täten niillä on merkittävä kansantaloudellinen merkitys. Kuitenkin kuormittuneen sosiaali- ja terveydenhuolto sektorin vuoksi tarvitaan uusia tehokkaampia menetelmiä vanhojen menetelmien tueksi, tai jopa niiden tilalle. Toisaalta addiktiot ovat myös tulleet osaksi jokapäiväistä elämää ja niiden vaikeudessa on monia tasoja. Kaikki eivät ole ongelmakäyttäjiä, vaikka saattavat toisinaan esimerkiksi juoda ongelmallisesti.

Tarvitaan myös menetelmiä, jotka eivät leimaa käyttäjänsä siinä määrin kuin esimerkiksi vieroitusasemalle hoitoon kirjautuminen tai a-klinikalla vierailu. Tähän tietokoneistetulla psykoterapialla voi olla jotakin tarjottavaa, koska ajan myötä se saattaa tehostaa terapia- ja kuntoutuskäytäntöjä. Tietokoneistettu psykoterapia myös mahdollistaa uudenlaista anonyymiutta päihdeongelmien kanssa painiville henkilöille.

Usein ongelmallisimpana addiktioiden hoidossa on pidetty asiakkaan motivoimista ja motivoitumista hoitoon [Miller, 2008]. Tässä voidaan olettaa menetelmien perustuvan sille, että niiden käyttäjillä on jonkinlainen oma motivaatio saada hoitoa.

## **Addiktioiden hoito ja psykoterapia**

Polut hoitoon addiktiosta kärsivän ihmisen elämässä ovat moninaisia. Kuusisto [2009] on löytänyt addiktioista toipumisessa kolme eri reittiä. Ne ovat spontaani, ammattiapua tai vertaistukea hyödyntävä tie. Usein myös eri tason ongelmat vaativat erilaatuista hoitoa, joten ongelmien hankaluus saattaa selittää osan eri hoitokanavien käytöstä.

Psykoterapia käsitteenä on varsin laaja. Psykologiatieteen erityislaatuisuudesta johtuen useat teoriasuuntaukset ovat kehittäneet omiin periaatteisiinsa painottuvia terapiamalleja. Tunnetuimpia terapian teoreettisia viitekehyksiä lienevät psykoanalyttinen, humanistinen, kognitiivinen sekä behavioristinen.

Suomessa sanalla psykoterapia usein viitataan terapiaan, joka on KELAn hyväksymä hoito. Psykoterapeuteilta vaaditaan ammattipätevyys. Tässä tutkielmassa psykoterapia käsitettä käytetään enemmänkin merkityksessä terapeutin hoito. Joissain menetelmissä on todella kyse psykoterapiasta, mutta toisissa vain hoidosta, joka ei välttämättä vaadi psykoterapeutin pätevyyttä. Ensimmäisessä tutkielmassa keskitytään kuitenkin menetelmiin, joissa tietokoneilla ja

teknologialla voidaan voimistaa ja edistää addiktioiden nykyistä hoitoa psykologisesta viitekehystä huolimatta.

Viitataan usein kognitiivis-behavioristisen terapiaan (CBT), koska useimmat tietokoneavusteista terapiaa hyödyntävät menetelmät käyttävät sitä viitekehysenään. Tämän vuoksi avaan käsitettä hieman. Kognitiivisella psykoterapialla on hyvä maine uusimpien tutkimustulosten sekä uusien kokeellisten menetelmien soveltamisessa käytäntöön [Hakanen, 2008]. Kognitiivisen psykologian asenne siis tekee siitä kokeellisesti orientoituvampaa, eikä voida väittää, että sen perusteissa olisi jotain, mikä tekisi siitä toimivampaa uusien menetelmien hyödyntämisessä.

Kognitiivinen psykoterapia perustuu ajatukselle kognitioiden tarkastelemisesta. Kognitiot ovat yksilön tiedonkäsittelyllisiä prosesseja [Hakanen, 2008]. Käytännössä kyseessä on siis yksilön sisäisten ajattelumallien sekä käsitysten, skeemojen ja skriptien, tarkastelemisesta ja muuttamisesta. Toimimattomat mallit pyritään käsittelemään sekä muuttamaan. Ajatustapaa muuttamalla ajatellaan myös toiminnan muuttuvan.

Kognitiivis-behavioristisen terapian perustana on ajatus siitä, että ihmisen ajatukset, tunteet, käytös sekä fysiologiset seikat muodostavat kokonaisuuden [Curven, 2000]. Kokonaisuuden osat ovat sidoksissa toisiinsa. Tällöin muuttamalla jotain kokonaisuuden osista tuotetaan muutosta myös toisella alueella [Curven, 2000]. Perusajatuksena on, että tunteet ja ajatukset johtavat tekoihin, mikä esimerkiksi addiktien kohdalla usein aiheuttaa ongelmia. Esimerkiksi muuttamalla käytöstä tai ajatusta, pyritään vaikuttamaan myös tunteisiin ja fysiologiaan. Ahdistavassa tilanteessa voisi, soveltamalla esimerkiksi hengitysharjoitusta tai jotain muuta sovittua toimintatapaa, välttää niitä ”luonnollisia” tapoja – paniikkia tai muuta negatiivista toimintaa – reagoida tilanteeseen.

## **Tietokoneistetut terapiat**

Tässä pyritään esittelemään niitä keinoja ja menetelmiä, joilla tietokoneita on jo hyödynnetty psykoterapian välineinä. Jaottelen menetelmät kolmeen osaan niiden luonteen kautta. Osa menetelmistä perustuu enemmänkin viestinnän parantamiseen toimien eräänlaisina yhteyden pitämisen välineistönä. Usein terapiaksi kutsutuissa menetelmissä terapeutilla on jonkinlainen osuus hoidossa. Osuus voi olla viesteihin vastaamista tai niiden lähettämistä asiakkaalle. Toiset menetelmät taas ovat itsehoitoon ja itsekontrollin avuksi kehitettyjä ohjelmistoja. Nämä tiettyjen algoritmien pohjalta toimivat menetelmät voivat olla esimerkiksi verkossa suoritettavia itsearviointilomakkeita. Keskeinen ero

edelliseen on, ettei niitä käytettäessä terapeutilla ole keskeistä roolia. Vuonna 2010 terveydenhuollon ohjelmia jo yli 100 [Carroll and Rounsaville, 2010].

Osa tietokoneen hyödyntämiskohteista on uusia avauksia, joiden avulla terapiassa voidaan esimerkiksi kohdata ongelmia, joiden kohtaaminen normaalielämässä olisi joko mahdotonta, hankalaa tai jopa vaarallista. Tästä oivana esimerkkinä toimivat virtuaalitodellisuutta hyödyntävät teknologiat, joilla esimerkiksi virtuaalisessa todellisuudessa esitettyjen ärsykkeiden avulla on yritetty vähentää addiktiivisten elementtien himoa.

### **Ohjelmistopohjaiset työkalut tietokoneistetussa psykoterapiassa**

Länsimaissa internet on nykyisin lähes jokaisen ulottuvilla. Langattomien- ja mobiiliverkkojen ansiosta nettiin pääsee lähes missä vain ja milloin vain. Myös julkiset laitokset, kuten kirjastot ja koulut, tarjoavat mahdollisuuden vierailta internetissä. Toisaalta myös tietokoneet ja matkapuhelimet, jotka usein nykyisin vastaavat tietokoneita, ovat lähes jokaisen ulottuvilla länsimaalaisessa maailmassa. Tämä antaa tietenkin uusia mahdollisuuksia myös tietokoneistetulle psykoterapialle. Aiemmin aikaan ja paikkaan sidotut perinteiset terapiat eivät kykene vastaamaan ihmisten hätään kuin tiettyinä aikoina terapeutin työpäivän aikana. Tähän verrattuna ohjelmistopohjaiset terapiat ovat todella joustavia, sillä niitä on mahdollista hyödyntää silloin kun niitä tarvitaan, olipa sitten yö tai päivä ja olipa sijaintisi tämä tai tuo [Copeland and Martin, 2004]. Ne ovat aina saatavilla palvelinten ja laitteiden ollessa toiminnassa.

Internetissä toimivilla palveluille keskeistä on myös niiden anonyymius, joka varsinkin laittomien huumeiden käyttäjille voi olla todella tärkeää, sillä pelko rikosseuraamustoimien kohteeksi joutumisesta on suuri. Nimettömissä internet-interventioissa ei myöskään tarvitse pelätä leimautuvansa päihteiden tai muiden addiktioiden kanssa painivaksi [Koski-Jännes et al., 2009; Marks and Cavanagh, 2009], mikä useilla henkilöillä on esteenä a-klinikalle kävelemisessä. Internetissä et myöskään joudu kohtaamaan toisia ”ongelmakäyttäjiä” tahtomattasi, mitä mahdollisesti et voi välttää asioidessasi julkisen terveydenhuollon palveluissa.

Ohjelmisto- ja internetpohjaiset interventiot ovat usein verkkosivustoja, joilta on mahdollista ladata tai suorittaa ohjelmia. Usein ohjelmat liittyvät itsearviointiin, jonka avulla hoitoa kaipaava henkilö voi joko arvioida hoidon tarvettaan tai hoidossa edistymistään sekä muita addiktion hoitamiseen liittyviä seikkoja. Itsearviointi hyödyntää teoreettisena viitekehystenään usein behavioristista näkökulmaa, joissa riippuvuuden hoitoon hyödynnetään käyttäytymisen muutokseen tähtääviä toimia. On myös todettu, että

kognitiivis-behavioristista terapiaa soveltamalla tietokoneiden välityksellä on saatu kannustavia tuloksia riippuvuuksien hoidossa [Copeland and Martin, 2004].

Carroll ja muut [2008] ovat tutkineet kognitiivis-behavioristisen terapiaan pohjautuvan ohjelman tehoa addiktioiden hoidossa. He tarkastelivat tietokone-ohjelmalla toteutettua kognitiivis-behavioristista terapiasovellusta (CBT4CBT) lisänä tavalliseen päihteiden käyttäjän hoitoon. Ohjelmisto piti sisällään videoita, graafisia kuvauksia sekä kertomuksia, joissa päihteiden käyttäjiä koulutettiin tulevien ongelmatilanteiden varalle.

Koehenkilöinä heillä oli 77 eri päihteiden käyttäjää, jotka oli siivilöity 155 henkilön joukosta. Valitulla joukolla ei ollut todettu muita vaikeita mielenterveyden ongelmia ja he täyttivät DSM-IV -ehdot päihderiippuvuudesta. Puolet koehenkilöistä määrättiin käyttämään CBT4CBT-ohjelmaa (n=39) normaalin hoidon lisäksi. Loput (n=38) osallistuivat vain tavalliseen hoitoon. Jokaiselta koehenkilöltä otettiin säännöllisesti huumeensa virtsasta sekä mitattiin hengityksen alkoholipitoisuus.

Tutkimus kesti kahdeksan viikon ajan. Tänä aikana huomattiin, että niillä koehenkilöillä, jotka osallistuivat CBT4CBT-ohjelman käyttöön, oli vähemmän positiivisia huumeensaurojen tuloksia tuon kahdeksan viikon aikana.

Tutkimus siis antaa hyviä viitteitä CBT4CBT-ohjelman mahdollisuuksista, mutta tämänkään tutkimuksen tulokset eivät ole vielä riittäviä. Tutkimus osoitti, että ohjelmalla oli vaikutusta positiivisten huumeensaurojen esiintyvyyteen. Kuitenkaan tutkimuksen avulla ei voida arvioida tavallisen hoidon ja tietokoneistetun hoidon suhdetta [Carroll et al., 2008], mikä mielestäni olisi todellakin keskeistä saada selville.

Tutkimuksessa ei myöskään hyödynnetty mitään varsinaista kontrolliryhmää, koska molemmat ryhmät saivat tavallista hoitoa. Klassisessa koeasetelmassa tutkittava joukko on usein jaettu koe- sekä kontrolliryhmään satunnaisesti. Tällöin koeryhmälle annetaan esimerkiksi jotain hoitoa ja kontrolliryhmälle taas ei. Psykologisissa tutkimuksissa tosin kaikki saavat lopulta hoitoa, koska eettisistä syistä ketään ei voida jättää hoidotta. Kaivattaisiin lisää tutkimuksia, jotka olisi suoritettu varsinaista kontrolliryhmää hyödyntäen. Tällöin voitaisiin arvioida CBT4CBT-ohjelman vaikutusta suhteessa muihin terapioihin.

Alkoholistien hoidossa Yhdysvalloissa on hyödynnetty tietokonepohjaista itsearviointilomaketta. BSCPW (Behavioural Self-Control Program for Windows) on ohjelma, jonka avulla alkoholisti voi arvioida sen hetkistä tilannettaan. Ohjelma tarjoaa mahdollisuuden tarkastella ja asettaa tavoitteita. Tavoit-

teisiin pääsemistä myös voi kontrolloida ja asettaa sanktioita, jos niitä ei saavutettu.

Hester ja Delaney tutkivat ohjelman tehokkuutta noin 40 alkoholistin otoksella, joista noin puolet käytti BSCPW:tä ja loput olivat 10 viikon ajan odotuslistalla, minkä jälkeen heille tarjottiin hoitoa. Koehenkilöillä oli mahdollisuus käyttää ohjelmaa 8 kertaa viikossa 15-45 minuutin ajan. Tutkimukset osoittivat, että menetelmä auttoi vähentämään juomista puolella. Koehenkilöt myös saivat menetelmän käytöstä pitkäaikaista hyötyä, sillä vähentyneen juomisen huomattiin kestävän jopa vuoden ajan.

On tietenkin huomattava, että lisää kokeita vaaditaan, sillä osa kokeen tuloksista voi johtua eräänlaisesta plasebovaikutuksesta kuten Marks ja muut [2007b] huomauttavat. Plasebovaikutuksella tarkoitetaan vaikutuksen ilmene mistä, vaikka varsinaista hoitoa ei ole annettu. Edellä mainitussa tapauksessa voisi jo se, että koehenkilöt saivat huomiota, vähentää heidän juomistaan.

Eräs sovellus ohjelmistopohjaisessa psykoterapiassa ovat internetistä löytyvät itsearviointilomakkeet. Koski-Jännes ja muut [2009] tutkivat suomalaisen internet-sivuston ([www.paihdelinkki.fi](http://www.paihdelinkki.fi)) juomatapoja mittaavan lomakkeen vaikutusta juomiseen. Sivustolla sijaitseva itsearviointilomake sisältää kysymyksiä henkilön juomatavoista, määristä sekä juomiseen liittyvistä kokemuksista. Koski-Jännes ja muut seurasivat 343 vapaaehtoisen päihdelinkin käyttäjän lomakkeeseen syötettyjä tietoja vuoden ajan.

Vapaaehtoiset osallistujat antoivat palveluun sähköpostiosoitteen, johon heille lähetettiin pyyntö käydä täyttämässä lomake 3, 6 ja 12 kk kohdalla ensimmäisestä itsearvioinnista. He vertasivat tutkimuksessa käytettyä koeryhmää ja satunnaisotannalla valitun joukon (n=538) täytettyjä itsearviointeja. Mielenkiintoinen seikka aineistossa on, että koeryhmästä oli naisia enemmän kuin satunnaisotannalla nostetussa joukossa ja koehenkilöt olivat myös hieman vanhempia sekä juomatavoiltaan ongelmallisempia [Koski-Jännes et al., 2009]. Naisten hieman suurempi määrä itsearviointilomakkeiden käyttäjinä on huomattu myös muiden tutkimusten osalta [Sinadinovic et al., 2010].

Koski-Jännes ja muut [2009] huomasivat juomisen vähentyneen koeryhmässä 12 kk aikajaksolla. Tulokset ovat lupaavia, mutta on huomioitava, että tulokset perustuvat koehenkilöiden itsearvioinneille, joten ne eivät ole täysin luotettavia, koska käytännössä tutkijoilla ei ollut mitään tietoa tutkittavistaan. Tutkitut saattoivat antaa väärää tietoa juomatavoistaan sekä identiteetistään, syystä tai toisesta. Toisaalta on myös huomattava, ettei itsearviointilomakkeen täyttäminen aiheuttanut tarvetta hakeutua hoitoon, vaikka juominen olisikin havaittu ongelmalliseksi.

Ehkä arviointilomakkeen täyttäminen on yhtä helppoa kuin kirkosta eroaminen tai tykkään-napin painaminen Facebookissa, mistä seuraa, että arviointeja kyllä tehdään, mutta kun pitäisi toimia, siis hakeutua hoitoon ihan oikeasti arkimaailmassa, niin toiminta syystä tai toisesta estyy. Toisaalta ei myöskään tiedetä, mitä ja miksi ihmiset todella täyttivät itsearviointilomakkeita: oliko heillä halu oikeasti saada selvyys omasta päihteiden käytöstä vai kenties jokin muu syy? Entä jos ihmiset vain halusivat osallistua tutkimukseen tai muutoin saada tietoa juomisestaan. Ovathan monet muutkin testit, esimerkiksi sosiaalisessa mediassa, todella suosittuja. Moni tekee internetissä testejä, mutta ei silti ota niiden tuloksia tosissaan.

Toisaalta mietin myös, kuinka hyvin hoitoonohjaus testin tekemisen jälkeen oli hoidettu. Oliko tulossivulla vain linkki a-klinikan yhteystietoihin? Kaipaisin kuitenkin jotain konkreettisempaa lomakkeiden lisäksi. Koemielessä tein Päihdelinkin juomatapatestin. Annoin aina mahdollisimman huonon vastauksen nähdäkseni palautteen, jonka ongelmajuoja saa. Ihmetyksekseni palautesivuilla ei ollut hoitoonohjausta lainkaan tarjolla. Sivulla oli linkki, josta kyllä pääsi yhteydenottosivulle. Olisiko mahdollisesti auttanut, jos kynnystä saada hoitoa olisi jotenkin madallettu, vaikka lisäämällä ajanvarausmahdollisuus testin tuloksien yhteyteen ja kerrottu enemmän mahdollisista hoitomenetelmistä? Mielestäni, jos ongelma on niin suuri kuin koemielessä antamieni vastausten tulokset (34/40) antoivat osoittaa, olisi keskeistä pyrkiä tarjoamaan madallettu polku yhteydenottoon ja ajanvaraukseen asti. Toisaalta on myös mahdollista, että itsearviointitestin tekijät kuuluivat Kuusiston [2009] esittelemässä jaotellussa niihin spontaanisti toipuneisiin, jotka ovat tehneet päätöksen lopettaa juomisensa itse, vaikkakin juomatapatestin innostamana, eivätkä he täten hakeudu hoitoon.

## **E-terapia**

Edellä esiteltiin tietokonetta sekä teknologioita hyödyntäviä terapian keinoja ja apuvälineitä, joissa terapeutilla oli minimaalinen tai olematon rooli. Tässä luvussa tuodaan esille niitä keinoja, joissa terapeutin ja klassisen terapian reper-tuaaria pyritään laajentamaan tietokoneiden ja viestintäteknologian avulla. Barak ja muut [2009] toteavat internetissä tapahtuvalla terapialla olevan neljä viestintätapaa: asynkroninen, synkroninen yksilöllinen ja ryhmäviestintä. Asynkroninen viittaa ei-reaaliaikaiseen viestintään, jolloin kaikki jäsenet eivät ole samanaikaisesti verkossa. Synkroninen taas tapahtuu reaaliajassa.

Viestintäkeinona toimivat verkkoteknologioita hyödyntävät menetelmät kuten videoneuvottelut sekä sähköpostin välityksellä käytävät keskustelut tera-

peutin kanssa. Lammilla toimivassa Mainiemen kuntoutumiskeskuksessa ollaan kehittelemässä hanketta ("Viidakko"), jossa avohoitoon siirtyville potilaille asennetaan kotiin kosketusnäyttöllinen näyttölaite, jolla he voivat olla kotoaan videoyhteydessä Mainiemen työntekijöihin. Hankkeen on tarkoitus edistää avohoitopalveluiden saatavuutta [Viidakko hankekortti]. Menetelmä saattaisi olla toimiva varsinkin jos välimatka asiantuntijoiden ja asiakkaan välillä on pitkä.

Internetissä on useita sivustoja, joilla päihdeongelmasta jaetaan tietoa, mutta joillain näistä sivustoista on myös mahdollista saada terapiaa, pikaviestinnän ja video neuvottelu yhteyden kautta [Griffiths, 2005]. Esimerkkinä esitellään Yhdysvaltalaisista sivustoja e-Getgoing ([www.egetgoing.com](http://www.egetgoing.com)). Sivusto tarjoaa apua erinäisiin päihteisiin liittyviin ongelmiin. Sivusto perustuu yhteisöhoitolisille periaatteille. Sivustolle rekisteröidytään ja luodaan tili, minkä jälkeen on mahdollista osallistua sivustolla tarjottaviin istuntoihin.

Sivusto tarjoaa mm. chat-alustalla toimivaa ryhmäterapiaistuntoa (Live Group Sessions) sekä videoneuvottelutekniikkaa hyödyntävää vertaisryhmäistuntoa (Video-based Peer Group). Asiakkaan on myös mahdollista perustaa sivustolle eräänlainen oma profiili -sivu, jonka avulla sivustolla voi navigoida tehokkaammin sekä seurata omaa hoidossa edistymistään. Myös yhteydenpito toisiin asiakkaisiin sekä henkilökuntaan on mahdollista. Mielestäni alusta muistuttaa opetuskäyttöön kehitettyä Moodlea. Ongelmana ainakin chat-pohjaisessa viestinnässä on osapuolten ymmärrys, kun viestinä käytetään pelkkää tekstiä [Barak et al., 2009]. Varsinkin, jos ottaa huomioon, että päihteitä käyttävillä henkilöillä kognitiiviset kyvyt, kuten tarkkaavaisuus ja keskittyminen, voivat olla heikentyneet [Carroll et al., 2008]. Tällöin voisi ajatella, että kirjoittaminen, vieläpä niin että tulee ymmärretyksi, voisi olla hankalaa.

Jotkin päihdehoidot tarjoavat asiakkailleen päihteitä korvaavia lääkeaineita, jotta nämä pysyisivät irti laittomista huumeista. Tästä käytetään nimitystä korvaushoito. Usein korvaushoidossa muu kuin lääkkeellinen hoito on vähäistä, mikä tosin riittää osalle potilaista [King et al., 2009]. Toiset potilaat kuitenkin kaipaavat lisää tukea.

King ja muut [2008] tutkivat videoneuvottelun toimivuutta korvaushoidossa olevien päihteidenkäyttäjien hoidon tukena. Koehenkilöinä heillä oli metadon-korvaushoidossa olevia addikteja, joilta otettiin huumeeseulat virtsasta viikoittain. Tutkimus antoi viitteitä videoneuvottelun tehokkuudesta korvaushoidon tukena. Tutkimus osoitti myös sen, että asiakkaat kokevat internetiin perustuvan internet-palvelun mielekkääksi [King et al., 2009], koska se on



kätevämpi käyttää ja siinä yksityisyyden suoja koettiin paremmaksi verrattuna tavalliseen hoitoon, jossa joutui kohtaamaan muita käyttäjiä.

## **Virtuaalitodellisuudessa tapahtuva terapia**

Virtuaalisten todellisuuksien hyödyntäminen psykoterapiassa tulee lisääntymään. Kysyttäessä psykoterapeuteilta, minkä menetelmien he uskoivat säilyvän tai kasvavan tulevaisuudessa, sijoittui virtuaalista ympäristöä hyödyntävä psykoterapia viiden parhaan joukkoon. Voidaankin sanoa, että menetelmillä nähdään potentiaalia eksperttien keskuudessa [Norcross et al., 2002]. Kysely toteutetaan aina kymmenen vuoden välein. Tosin on huomioitava, että kysely toteutettiin vuosisadan alussa lähes kymmenen vuotta sitten, jolloin nyt pitäisi olla nähtävissä, kuinka hyvin nämä terapiat ovat yleistyneet. Emme vielä ole saaneet tuloksia uudesta 2010 järjestettävästä kyselystä, mutta olisi mielenkiintoista nähdä, onko virtuaalinen ja tietokoneistettu psykoterapia säilyttänyt maineensa tulevaisuuden keskeisenä terapia keinona.

Mitä sitten tarkoitetaan virtuaalisella todellisuudella tai sille pohjautuvalla psykoterapialla? Virtuaalisella todellisuudella tarkoitetaan todellisuutta, joka on luotu koneellisesti, jossa ihminen voi kokoa olevansa läsnä [Riva, 2005]. Läsnä olemisella tarkoitetaan tässä sitä samaa tunnetta, joka ihmisillä on heidän jokapäiväisessä elämässään. Tunne siitä että olen osa minun ulkopuolistani maailmaa. Virtuaalinen todellisuus pyrkii tuottamaan ihmiselle juuri edellä kuvatuunlaiseen tunteen siitä, että hän on osallinen tuohon maailmaan. Virtuaalinen todellisuus siis syntyy kun aisteille tarjotaan vaihtelevassa määrin keinotekoisesti tuotettuja ärsykeitä.

Usein virtuaalinen todellisuus rakennetaan pukemalla asiakkaan päälle laitteisto, joka sisältää silmiä lähelle tulevan näytön. Tämä luo täysin erilaisen tunteen kuin esimerkiksi tietokoneen näyttö, joka havaitaan kuuluvaksi ulkopuoliseen maailmaan eikä itsen jatkeeksi. Välineistöön kuuluvat usein myös datahansikkaat, joilla keinotekoisesti luodussa todellisuudessa ollaan vuorovaikutuksessa ympäristön kanssa. Toisaalta datahansikkaat voidaan korvata eräänlaisilla robottikäsillä, jota reagoivat käyttäjän syötteisiin sekä antavat käyttäjälle myös palautetta esimerkiksi vastuksen muodossa. Myös liikekameraa on joskus käytetty mittaamaan käyttäjän liikkeitä.

Kypärän sijasta voidaan myös käyttää varta vasten suunniteltua huonetta CAVE (Computer Automatic Virtual Environment), jossa ympäristö tuotetaan heijastamalla seinille kuvaa luodusta ympäristöstä [Emmelkamp, 2005]. Myös

eräänlaisia paranneltuja todellisuuksia (augmented reality), joissa fyysiseen maailmaan istutetaan virtuaalisia objekteja, voidaan hyödyntää.

Virtuaalisessa ympäristössä on myös ongelmia, sillä se ei sovi kaikille. Osa henkilöistä alkaa pian kytkeydyttyään virtuaaliseen todellisuuteen kokea pahoinvointia, jota nimitetään kuvaavasti kybersairaudeksi (vrt. merisairaus). Se aiheutuu usein aistielinten ristiriitaisesta informaatiosta.

Myöskään kaikki henkilöt eivät kykene kokemaan virtuaalisia maailmoja niin intensiivisesti, että siitä olisi terapialle hyötyä [Emmelkamp, 2005]. Terapioiden kannalta keskeisenä pidetään usein tunteiden kanssa työskentelyä ja niiden läpikäymistä. Jos virtuaalinen maailma ei synnytä mitään tunnereaktioita, ei siitä ole iloa terapeutiselle instituutiolle.

Virtuaalista ympäristöä on hyödynnetty myös muidenkin kuin kognitiivis-behavioristisen terapian viitekehyksessä, mikä tekee siitä mielenkiintoisen alueen tietokoneistetussa psykoterapiassa. Lupaavia tuloksia on saatu psykodynaamisen terapian kentällä, jossa terapeuttien on ollut mahdollista rakentaa virtuaalisia ympäristöjä, joissa terapian kannalta keskeisiä tunnereaktioita on kyetty synnyttämään sekä käsittelemään [Riva, 2003]. Psykodynaaminen terapia on käytännössä asiakkaan fantasioiden, kokemusten ja unien tarkastelua. Nyt esimerkiksi unien sisältö voidaan asiakkaan ja terapeutin yhteistyöllä tuoda virtuaaliseen todellisuuteen. Tämä antaa uudenlaisia mahdollisuuksia asioiden tarkasteluun.

Virtuaalista todellisuutta on myös kokeiltu alkoholin ja tupakan himojen vähentämisessä [Moon and Lee, 2009]. Osa tähänastisista tutkimuksista on perustunut behavioristiseen teoriaan riippuvuudesta, jonka mukaan riippuvaisiksi tulevat ihmiset jotka tulevat riippuvaisiksi, ovat oppineet liittämään mielihyvän tiettyihin päihteisiin, jolloin jo näihin päihteisiin liittyvien asioiden ja vihjeiden (cue) näkeminen aiheuttaa halun päihteen nauttimiseen [Kolb and Whishaw, 2008]. Virtuaalista todellisuutta pyritty hyödyntämään riippuvuuskien behavioristisessa hoidossa. Edellä mainitulle teoreettiselle viitekehykselle pohjautuva CE-terapia (Cue-Exposure Therapy) on havaittu tehokkaaksi ainakin alkoholismiin liittyvän päihteen nauttimisen halun hoidossa [Wiederhold and Wiederhold, 2008; Lee et al., 2007]. On tutkittu virtuaalisessa ympäristössä esitettyjen vihjeiden vaikutusta ongelmakäyttäjiiin. Lopulta on saatu viitteitä siitä, että päihteisiin liittyvien virtuaalisten representaatioiden on huomattu herättävän käyttäjissä haluja käyttää päihdettä [Bordnick et al., 2005].

Moon ja muut [2009] pyrkivät tarkastelemaan virtuaalisen ympäristön vaikutusta tupakointiin liittyvään himoon. Tämä toteutettiin näyttämällä koehenkilöille tupakointiin liittyviä vihjeitä virtuaalisessa baarissa. Koehenkilöt liik-

kuivat baarissa avatar-hahmojen avulla sekä viettivät iltaa, tupakoivat ja joivat alkoholia. Koehenkilöt liikkuvat baarissa päälle puetun HMD-laitteen avulla. Kokeeseen kuului myös haastattelu, jossa heidän kokemuksiaan ja mielitekojaan tuotiin esille.

Tutkimuksen aluksi koehenkilöille oli esitetty sekä tupakointiin liittyviä valokuvia että neutraaleja kuvia, minkä aikana heidän aivojaan kuvannettiin fMRI-tekniikalla. Functional Magnetic Resonance Imaging (fMRI) tarkastelee magneettisia vaihteluita aivoissa, eli käytännössä veren virtausta aivoissa. Tällöin päästään havaitsemaan, millä alueilla aivoissa milloinkin on aktivaatiota esimerkiksi tietyn tehtävän aikana.

Neutraalien kuvien aikana riippuvuuksiin yhdistetyillä aivoalueilla oli vähemmän aktivaatiota kuin tupakointiin liittyvien kuvien kohdalla. Asetelma toistettiin myös virtuaalisessa baarissa liikkumisen jälkeen, jolloin aktivaatio tupakointiin liittyviin kuviin väheni. Kyseinen tutkimus on vasta alulla, joten joudumme odottamaan vielä varsinaisia tuloksia käytettäessä kokeellista asetelmaa verrokkiryhmällä.

Mielestäni edellä esitetyllä tutkimuksella voi todellakin olla hyötyä eräänlaisena tukitoimena addiktioiden hoidossa. Joskin on otettava huomioon, että behavioristiset teoriat riippuvuudesta eivät tavoita kaikkia niitä aspekteja riippuvuudesta, joita ehkä muut addiktiota selittävät teoriat tavoittavat. Addiktio ei ole pelkästään opittua vaan siinä on monia eri tasoja. Tämän vuoksi rinnalle tarvitaan muunkinlaista terapiaa. Muutoin näkisin kyllä virtuaalisten maailmojen hyödyntämiseen olevan todella keskeinen alue, kun tulevaisuuden terveydenhuollon menetelmiä kehitetään. Mietin, voisiko virtuaalisia ympäristöjä hyödyntää ehkä toisella tavalla addiktioiden hoidossa. Ehkä sillä voisi valmentaa henkilöitä niihin todellisen elämän tilanteisiin, joissa heidän päihitteettömyyttään koetellaan. Ehkä keskeistä voisi olla liikkuminen virtuaalisessa baarissa, jossa henkilöt suostuttelisivat potilasta nauttimaan päihitteitä.

Virtuaaliset ympäristöt ovat toisaalta tässä vaiheessa vielä aivan aluillaan. Ainakin itse en vakuuttunut [www.virtuallybetter.com](http://www.virtuallybetter.com) sivustolla esitetyistä esimerkeistä, sillä niiden graafinen ulkoasu on niin kankea ja palikkamainen. Myöskään avoimeen lähdekoodiin perustuva neuroVR ei vakuuta minua, vaikka graafisilta ominaisuuksiltaan se onkin jo kohtalainen. Kuten Emmelkamp [2005] huomauttaa, virtuaalisten ympäristöjen rajat liikkuvat kun tietokoneiden prosessointi tehot kasvavat.

Toisaalta, vaikka teknologia on edistynyt paljon, kaivataan kokemuksellisuuden tuottamiseen mielestäni vielä enemmän. Mielestäni näiden ympäristöjen käyttöliittymissä on tapahduttava paljon muutosta, jotta kokemukselli-

suutta saataisiin parannettua. Mielenkiintoista voisi olla multimodaalisuuden tuominen virtuaaliseen todellisuuden käyttöliittymiin, jos sillä saataisiin aikaan välittömämpää vuorovaikutusta ympäristön kanssa. NeuroVR hyödyntää muun muassa HMD-laitetta ja hiirtä, mikä mielestäni ei ole käytettävyydeltään välttämättä paras ratkaisu. En ihmettele, jos joku hiiren avulla ei voi kokea virtuaalista maailmaa niin ”todellisena”, että se herättäisi hänessä affekteja. Itse asiassa ihmettelen enemmän, jos joku voi hiiren avulla liikkua kokea olevansa osa virtuaalista ympäristöään. Toisaalta henkilöt ovat jo nyt hyötynyt näistä ympäristöistä.

Myös näiden alustojen hinta sekä muokattavuus voivat olla esteitä niiden integroitumiselle jokapäiväiseen terapian käytäntöön. Osa laitteista saattaa myös tarvita toimiakseen asiantuntevan asentajan tai operaattorin, mikä lisää kuluja edelleen ja tuo mielestäni ongelmallisia kysymyksiä terapian yksityisyydestä. Toisaalta osa näistä ongelmista voi poistua ajan myötä teknologioiden kehittyessä ja algoritmien kehittyessä.

## **Yhteenveto**

Tietokoneavusteisen terapian kenttä on moninainen kuten aiemmasta voimme huomata. On myös hankala vastata kysymykseen, onko tietokoneistettu psykoterapia tehokasta, koska käyttökohteet ja toteutustavat ovat niin moninaisia [Marks and Cavanagh, 2009]. Marks ja muut [2007a] näkevät, ettei tietokoneavusteisen psykoterapian tehokkuutta voi kiistää, vaan on pikemminkin löydettävä ne kohteet ja potilaat, joihin se vaikuttaa tehokkaimmin. Tulokset ovat myös antaneet viitteitä siitä, että se on mahdollisesti tehokas hoitomenetelmä addiktioiden hoidossa. Carroll ja muut [2010] ovat hieman varovaisempia arvioidessaan tietokoneistetun psykoterapian tehokkuutta. Lisätutkimuksia tarvitaan joka tapauksessa, koska useimmat tähänastiset tutkimukset ovat otokseltaan ja tutkimusasetelmiltaan vajavaisia, mistä seuraa että tarvitaan vielä paljon lisätutkimuksia, jotta menetelmien todellinen teho saataisiin selville.

Toisaalta tietokoneet tuovat paljon uusia mahdollisuuksia addiktioiden hoitoon. Terapiaa tukevat ohjelmistot nähdään hyvänä lisänä terapeutin työkalupakkiin ja niitä on kehitetty jo useita. Internetissä ja ladattavina ohjelmina toimivat mini-interventiot ja itsearviointien välineet saattavat vastata niiden addiktien tarpeisiin, joiden addiktio ei ole muodostunut vielä niin vakavaksi, että he olisivat menettäneet esimerkiksi työpaikkansa. Ehkä Kuusiston toipumisreiteissä nämä ohjelmistopohjaiset psykoterapiat voisivat vastata spontaanisti toipuneiden henkilöiden tarpeisiin.

Myös virtuaalisella todellisuudella on varmasti tulevaisuudessa monia käyttökohteita. Varsinkin sitten, kun niiden hinnassa tapahtuu alenemista. Mielenkiintoisia sovelluksia voisivat olla esimerkiksi virtuaalisessa todellisuudessa tapahtuvat terapiaistunnot, joissa avatarien välityksellä voitaisiin olla vuorovaikutuksessa toisten kanssa. Ongelmaksi vielä tässä vaiheessa näiden lähestymistapojen kanssa nousevat tunteiden käsittelyyn ja teknologiaan liittyvät ongelmat [Glantz et al., 2003]. Kunhan tunteiden tunnistusteknologioissa edistytään, voidaan joskus mahdollisesti päästä tilanteeseen, jossa henkilö voi tavata terapeuttinsa vaikka toiselta puolelta maailmaa virtuaalisessa istunnossa. Erittäin kiintoisalta vaikuttaisi soveltaa CBT4CBT-ohjelmiston asetelmia sekä kognitiivis-behavioristisen terapian periaatteita virtuaalisessa todellisuudessa. Ehkä virtuaaliset todellisuudet voisivat tarjota hyvän ympäristön eräänlaiselle toimivampien toimintatapojen harjoittelulle.

Kuitenkin teknologian mennessä eteenpäin vain pienissä askelissa on hyvä keskittyä niihin nykyisiin ratkaisuihin, joita käytössämme jo on. Jatkossa olisi myös mielenkiintoista arvioida kuinka nykyisiä tietokoneistetun psykoterapian ratkaisuja voisi integroida olemassa oleviin sosiaalihuollon palveluihin täällä Suomessa.

## Viiteluettelo

- [Barak et al., 2009] Azy Barak, Britt Klein and Judith G. Proudfoot, Defining internet-supported therapeutic interventions. *Annals of Behavioral Medicine* **38**, 1 (2009), 4-17.
- [Bordnick et al., 2005] Patrick S. Bordnick, Amy C. Traylor, Ken M. Graap, Hilary L. Copp and Jeremy Brooks, Virtual reality cue reactivity assessment: a case study in a teen smoker. *Applied Psychology and Biofeedback* **30**, 3 (2005), 187-193.
- [Carroll et al., 2008] Cathleen M. Carroll, Samuel A. Ball, Steve Martino, Charla Nich, Theresa A. Babuscio, Kathryn F. Nuro, Melissa A. Gordon, Galina A. Portnoy and Bruce J. Rounsaville, Computer-assisted delivery of cognitive-behavioral therapy for addiction: a randomized trial of CBT4CBT. *Am. J Psychiatry* **165** (2008), 881-888.
- [Carroll and Rounsaville, 2010] Kathleen M. Carroll and Bruce J. Rounsaville, Computer-assisted psychotherapy in psychiatry: be brave – it's a new world. *Current Psychiatry Reports* **12**, 5 (2010), 426-432.

- [Copeland and Martin, 2004] Jan Copeland and Greg Martin, Web-based interventions for substance use disorders: A qualitative review. *Journal of Substance Abuse Treatment* **26**, 2 (March 2004), 109-116.
- [Curven, 2000] Berni Curven, Stephen Palmer, Peter Ruddell, *Brief Cognitive Behavior Therapy*. Thousand Oaks, 2000.
- [Emmelkamp, 2005] Paul M. G. Emmelkamp, Technological innovations in clinical assessment and psychotherapy. *Psychotherapy and Psychosomatics* **74**, 6 (2005), 336-343.
- [Glantz et al., 2003] Kalman Glantz, Albert Rizzo and Ken Graap, Virtual reality for psychotherapy: current reality and future possibilities. *Psychotherapy: Theory, Research, Practice, Training* **40**, 1-2 (2003), 55-67.
- [Griffiths, 2005] Mark Griffiths, Online therapy for addictive behaviors. *CyberPsychology & Behavior* **8**, 6 (December 2005), 555-561.
- [Hakanen, 2008] Armi Hakanen, Kognitiivisen psykoterapian luonne ja kehityshistoria. Teoksessa: Seppo Kähkönen, Irma Karila ja Nils Homberg (toim.), *Kognitiivinen psykoterapia*, Duodecim, 2008.
- [King et al., 2009] Van L. King, Kenneth B. Stoller, Michael Kidorf, Kori Kindbom, Steven Hursh, Thomas Brady and Robert K. Brooner, Assessing the effectiveness of an internet-based videoconferencing platform for delivering intensified substance abuse counseling. *Journal of Substance Abuse Treatment* **36** (2009), 331-338.
- [Kolb and Whishaw, 2008] Bryan Kolb and Ian Q. Whishaw, *Fundamentals of Human Neuropsychology*. Worth Publishers, 2008.
- [Koski-Jännes et al., 2009] Anja Koski-Jännes, John Cunningham and Kari Tolonen, Self-assessment of drinking on the internet – 3-, 6- and 12-month follow-ups. *Alcohol & Alcoholism* **44**, 3 (2009), 301-305.
- [Kuoppasalmi et al., 2007] Kimmo Kuoppasalmi, Pekka Heinälä ja Juoko Lönnqvist, Päihteisiin liittyvät häiriöt. Teoksessa: Jouko Lönnqvist, Veikko Aalberg ja muut (toim.), *Psykiatria*, Duodecim, 2007.
- [Kuusisto, 2009] Katja Kuusisto, Päihderiippuvuudesta toipumisen eri reitit. Teoksessa: Tuukka Tammi, Mauri Aalto ja Anja Koski-Jännes, *Irti päihdeongelmista*, Edita 2009.
- [Lee et al., 2007] J. H. Lee, H. Kwon and B. H. Yang, Cue-exposure therapy to decrease alcohol craving in virtual environment. *CyberPsychology & Behavior* **10**, 5 (2007), 617-623.
- [Marks and Cavanagh, 2009] Isaac M. Marks and Kate Cavanagh, Computer-aided psychological treatments: evolving issues. *Annual Review of Clinical Psychology* **5**, (2009), 121-141.

- [Marks et al., 2007a] I. M. Marks, K. Cavanagh and L. Gega, Computer-aided psychotherapy: revolution or bubble? *The British Journal of Psychiatry* **191**, (2007), 471-473.
- [Marks et al., 2007b] Isaac M. Marks, Kate Cavanagh and Lina Gega, *Hands-on Help: Computer-aided Psychotherapy*. Psychology Press, 2007.
- [Miller, 2008] William Miller, Motivaation ja muutoksen käsitteellistäminen. Teoksessa: Anja Koski-Jännes, Liisa Riittinen ja Pekka Saarnio, *Kohti muutosta: motivointimenetelmiä päihde- ja käyttäytymisongelmiin*. Tammi, 2008.
- [Moon and Lee, 2009] Jiyoung Moon and Jang-Han Lee, Cue exposure treatment in a virtual environment to reduce nicotine craving: a functional MRI study. *CyberPsychology & Behavior* **12**, 1 (2009), 43-45.
- [Norcross et al., Prochaska, 2002] John C. Norcross, Melissa Hedges and James O. Prochaska, The face of 2010: a Delphi poll on the future of psychotherapy. *Professional Psychology: Research and Practice* **33**, 3 (June 2002), 316-322.
- [Riva, 2005] Giuseppe Riva, Virtual reality in psychotherapy: review. *CyberPsychology & Behavior* **8**, 3 (June 2005), 220-230.
- [Riva et al., 2006] Giuseppe Riva, Monica Bacchetta, Gianluca Cesa, Sara Conti, Gianluca Castelnovo, Fabrizia Mantovani and Enrico Molinari, Is severe obesity a form of addiction?: rationale, clinical approach, and controlled clinical trial. *CyberPsychology & Behavior* **9**, 4 (2006), 457-479.
- [Ruth et al., 2010] Ruth J. van Holst, Wim van den Brink and Dick J. Veltman, Anna E. Goudriaan, Brain imaging studies in pathological gambling. *Current Psychiatry Rep* **12** (2010), 418-425.
- [Sinadinovic et al., 2010] Kristina Sinadinovic, Anne H. Berman, Dan Hasson and Peter Wennberg, Internet-based assessment and self-monitoring of alcohol and drug use. *Addictive Behaviors* **35** (2010), 464-470.
- [Thombs, 2006] Dennis L. Thombs, *Introduction to Addictive Behaviors*. Guilford Press, 2006.
- [Viidakko hankekortti] Silja Wahlsten, Viidakko hankekortti. Viitattu: 12.12.2010. url: [elgg.valittajahanke.fi/action/file/download?file\\_guid=2643](http://elgg.valittajahanke.fi/action/file/download?file_guid=2643).
- [Wiederhold and Wiederhold, 2008] Brenda K. Wiederhold and Mark D. Wiederhold, Virtual reality with fMRI: a breakthrough cognitive treatment tool. *Virtual Reality* **12**, 4 (2008), 259-267.