

# Sisältöperusteinen kuvahaku ja piirteiden assosioiminen itseorganisoituvien karttojen avulla

Vuokko Aarnio

Pro gradu -tutkielma



ITÄ-SUOMEN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tietojenkäsittelytiede

Huhtikuu 2014

ITÄ-SUOMEN YLIOPISTO, Luonnontieteiden ja metsätieteiden tiedekunta, Kuopio  
Tietojenkäsittelytieteen laitos  
Tietojenkäsittelytiede

Aarnio, Vuokko Maria: Sisältöperusteinen kuvahaku ja piirteiden assosioiminen itseorganisoituvien karttojen avulla  
Pro gradu -tutkielma, 60 s.  
Pro gradu -tutkielman ohjaaja: FT Pekka Toivanen  
Huhtikuu 2014

Sisältöperusteisessa kuvahaussa haetaan kuvia niiden sisältöä edustavien, automaattisesti laskettavien piirteiden kuten värin, tekstuurin tai muodon avulla. Piirteistä voidaan muodostaa tehokkat haut mahdollistava tietorakenne käyttäen esimerkiksi itseorganisoituvia karttoja. Kirjallisuudessa esiintyy muutamia järjestelmiä, jotka oppivat assosioimaan yhdessä esiintyviä piirteitä itseorganisoituvien karttojen välisten yhteyksien avulla.

Tässä työssä toteutettiin järjestelmä, joka hakee esimerkkikuvalle samanlaisia kuvia väri- ja tekstuuripiirteiden perusteella ja lisäksi assosioi yhdessä esiintyviä värejä ja tekstuureja. Koulutusvaiheessa muodostetaan kaksi kasvavaa hierarkkista itseorganisoituvaa karttaa, joista toinen koulutetaan annetusta kuvajoukosta laskettavilla väripiirteillä ja toinen samasta tai eri kuvajoukosta laskettavilla tekstuuripiirteillä. Seuraavaksi muodostetaan kolmannella kuvajoukolla yhteyksiä aina eri karttojen niiden neuronien välille, jotka vastaavat parhaiten kutakin kuvaa toinen värin ja toinen tekstuurin puolesta. Koulutetulla järjestelmällä voi hakea annetulle esimerkkikuvalle samanlaisia kuvia värin tai tekstuurin tai molempien mukaan. Lisäksi on mahdollista hakea kuvia, joiden tekstuuri on yhdistetty esimerkkikuvan väriin tai päinvastoin.

Järjestelmällä löytyi esimerkkikuville värin ja tekstuurin puolesta melko samanlaisia kuvia. Värihaut toimivat erityisen hyvin yhtenäisen värisille kuville. Järjestelmä oppi hyvin assosioimaan koulutuskuvajoukossa esiintyviä värin ja tekstuurin yhdistelmiä. Haettaessa samanlaisia kuvia koulutuskuvajoukon ulkopuoliselle kuvalle haut suoraan ja yhteyksien kautta tarjosivat hieman toisistaan poikkeavia mutta toisiaan täydentäviä tuloksia. Järjestelmään olisi mahdollista vielä ottaa mukaan esimerkiksi avainsana-annotaatioita, jolloin yhteyksien avulla voisi yhdistää visuaalista ja semanttista tietoa toisiinsa.

Avainsanat: itseorganisoituva kartta, kasvava hierarkkinen itseorganisoituva kartta, sisältöperusteinen kuvahaku, väripiirteet, tekstuuripiirteet

ACM-luokat (ACM Computing Classification System, 2012 version): K.3.8 [Artificial intelligence]: Computer vision --- Computer vision tasks --- Visual content-based indexing and retrieval; H.5.5 [Information retrieval]: Retrieval tasks and goals --- Clustering and classification; K.4.1 [Machine learning]: Learning paradigms --- Unsupervised learning --- Cluster analysis, Dimensionality reduction and manifold learning

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Kuopio  
School of Computing  
Computer Science

Aarnio, Vuokko Maria: Content-based image retrieval and association of features using self-organizing maps  
Master's Thesis, 60 p.  
Supervisor of the Master's Thesis: PhD Pekka Toivanen  
April 2014

In content-based image retrieval, images are retrieved based on computational features such as color, texture or shape that represent their content. The features can be organized into a data structure that enables efficient searches using e.g. self-organizing maps. In literature, there are a few systems that learn to associate co-occurring features using connections between self-organizing maps.

In the present study, a system was implemented that finds similar images for an example image based on color and texture features, and also associates co-occurring colors and textures. In the training phase, two growing hierarchical self-organizing maps are formed, one of which is trained with color features calculated from a given set of images and another with texture features calculated from the same or different set of images. Then, connections are formed with a third set of images always between the neurons on different maps that best correspond to each image, one based on color and the other on texture. The trained system can be used to find similar images for a given example image based on color or texture or both. In addition, it is possible to search images whose texture is connected to the color of the example image or *vice versa*.

The system found images with rather similar colors or textures as in example images. Color searches worked especially well for images that had a uniform color. The system learned well to associate the color-texture combinations occurring in the training set of images. When searching similar images for an image outside of the training set, the direct searches and the searches through the connections produced slightly different but complementing results. Further, e.g. keyword annotations could be incorporated in the system after which the connections could be used to associate visual and semantic information.

Keywords: self-organizing map, growing self-organizing map, content-based image retrieval, color features, texture features

CR Categories (ACM Computing Classification System, 1998 version): K.3.8 [Artificial intelligence]: Computer vision --- Computer vision tasks --- Visual content-based indexing and retrieval; H.5.5 [Information retrieval]: Retrieval tasks and goals --- Clustering and classification; K.4.1 [Machine learning]: Learning paradigms --- Unsupervised learning --- Cluster analysis, Dimensionality reduction and manifold learning

## **Esipuhe**

Pro gradu -työ tehtiin Itä-Suomen yliopiston tietojenkäsittelytieteen laitokselle alkaneen syksystä 2010. Työ joutui kilpailemaan ajankäytöstäni samaan aikaan toisaalla tekemäni väitöskirjatyön kanssa ja jäi usein tappiolle. Kiitän ohjaajaani Pekka Toivasta sekä myös Keijo Haatajaa ideoinnista ja ohjauksesta.

Kuopiossa 7.4.2014

Vuokko Aarnio

## Lyhenneluettelo

A-SOM	assosiatiivinen itseorganisoituva kartta (Associative Self-Organizing Map, Johnsson & Balkenius, 2008)
BMU	voittajaneuroni (Best Matching Unit), itseorganisoituvan kartan neuroni, jonka painovektori on lähimpänä annettua syötevektoria (Koskela, 2003)
GHSOM	kasvava hierarkkinen itseorganisoituva kartta (Growing Hierarchical Self-Organizing Map, Rauber & <i>al.</i> , 2002)
HTML	Hyper Text Markup Language, rakenteellinen merkkäuskieli, jolla web-sivut kirjoitetaan (World Wide Web Consortium, 2013)
L*a*b	väriavaruus, jossa väri esitetään kolmella arvolla siten, että yksi kuvaa valoisuusastetta, toinen paikkaa vihreä-magenta-akselilla ja kolmas paikkaa sininen-keltainen-akselilla (Wikipedia, 2013a)
MQE	keskimääräinen kvantisointivirhe (Mean Quantization Error) GHSOM:ssa (Rauber & <i>al.</i> , 2002)
QBIC	Query By Image Content, eräs sisältöperusteinen kuvahakujärjestelmä (Niblack & <i>al.</i> , 1993)
qe	kvantisointivirhe (Mean Quantization Error) GHSOM:ssa (Rauber & <i>al.</i> , 2002)
RGB	”Red-Green-Blue”, väriavaruus, jossa väri esitetään kolmella arvolla siten että yksi kuvaa punaista, toinen vihreää ja kolmas sinistä (Wikipedia, 2013b)
SOM	itseorganisoituva kartta (self-organising map) , neuroverkkosovellus (Kohonen, 1982)
URL	Uniform Resource Locator, Internetissä olevan tiedoston yksilöivä teksti (Mitchell, 2013)

# Sisällysluettelo

1	Johdanto .....	1
2	Taustaa sisältöperusteisesta kuvahausta .....	2
2.1	Kuvien esittäminen piirteiden avulla .....	3
2.2	Kuvien indeksointi.....	6
2.3	Haun käsittely .....	7
3	Taustaa itseorganisoituvista kartoista .....	9
3.1	Itseorganisoituvan kartan (SOM) teoria.....	9
3.2	Kasvava hierarkkinen itseorganisoituva kartta (GHSOM) .....	10
4	Muita järjestelmiä, jotka assosioivat piirteitä SOM:ien avulla .....	15
4.1	Neuronille parhaan parin toiselta SOM:lta etsivä järjestelmä.....	16
4.2	SOM:eja Hebbin neuroverkolla yhdistävä järjestelmä .....	18
4.3	Sivupainovektoreita käyttävä assosiatiiivinen SOM .....	21
5	Piirteitä assosioivan sisältöperusteisen kuvahakujärjestelmän kuvaus.....	24
5.1	Piirteiden laskeminen.....	24
5.2	GHSOM:t ja niiden koulutus.....	26
5.3	Kuvan kartoitus GHSOM:lle .....	26
5.4	Yhteyksien luominen GHSOM:iien välille .....	27
5.5	Hakutoimintojen kuvaukset.....	28
5.5.1	Haku yhden piirrelajin mukaan yhdeltä GHSOM:lta .....	28
5.5.2	Haku yhden piirrelajin mukaan yhteysrakenteesta.....	29
5.5.3	Haku sekä värin että tekstuurin mukaan .....	30
5.5.4	Kuvan väriin liittyviä tekstuureja tai tekstuuriin liittyviä värejä edustavien kuvien haku .....	31
6	Järjestelmän toiminnan testaus ja arviointi .....	33
6.1	Toiminta yhtenäisen värisillä ja eri tekstuureja edustavilla kuvilla..	33
6.2	Toiminta erivärisiä alueita sisältävillä esittäville kuvilla.....	41
7	Yhteenveto .....	48
	Viitteet .....	53

# 1 Johdanto

*Sisältöperusteinen kuvahaku* on tieteenala, jossa pyritään järjestämään ja hakemaan digitaalisia kuvia automaattisesti niiden sisällön perustella. Kuvat esitetään laskennallisten *piirteiden* avulla, joista tavallisimpia ovat väri, tekstuuri ja muoto. Lisäksi kuvat on järjestettävä tällaisten piirteiden mukaan jonkinlaiseen tietorakenteeseen siten, että niiden haku on tehokasta. Tähän tarkoitukseen on käytetty monia menetelmiä, mm. itseorganisoituvia karttoja.

Jonkin piirteen suhteen samanlaiset kuvat saattavat olla erilaisia jonkin toisen piirteen mukaan. Monissa järjestelmissä kuvat luokitellaankin erikseen eri piirteiden mukaan, esimerkiksi muodostamalla jokaisesta erillinen itseorganisoituva kartta. Toisaalta on kehitetty myös järjestelmiä, jotka pyrkivät assosioimaan yhdessä esiintyviä piirteitä toisiinsa. Tämä mahdollistaa esimerkiksi sen, että jonkin piirrelajin tiedon puuttuessa se voidaan johtaa toisen piirrelajin tiedosta assosiaatioiden avulla.

Tässä työssä toteutettiin järjestelmä, joka järjestää annetun kuvajoukon värin ja tekstuurin mukaan kahdella eri itseorganisoituvalla kartalla ja rakentaa assosiaatioita yhdessä esiintyvien värien ja tekstuurien välille. Järjestelmällä on mahdollista hakea annetulle esimerkkikuvalle samanlaisia kuvia erikseen värin tai tekstuurin mukaan tai samanaikaisesti molempien mukaan. Lisäksi voidaan hakea esimerkkikuvan väriin liittyviä tekstuureja tai tekstuuriin liittyviä värejä edustavia kuvia. Työssä haluttiin nähdä, luokittelevatko itseorganisoituneet kartat kuvia hyvin värin ja tekstuurin perusteella ja vastaavatko hakutulokset esimerkkikuvia. Lisäksi haluttiin nähdä, miten järjestelmä oppii yhdistämään kuvajoukossa yhdessä esiintyviä värejä ja tekstuureja toisiinsa.

Tässä tutkielmassa kerrotaan aluksi taustaa toisaalta sovellusalueesta eli sisältöperusteisesta kuvahausta luvussa 2 ja toisaalta käytetystä tietorakenteesta eli itseorganisoituvista kartoista luvussa 3. Kirjallisuudesta löytyy myös muutamia järjestelmiä, jotka assosioivat piirteitä itseorganisoivien karttojen avulla, ja myös näistä kerrotaan luvussa 4. Luvussa 5 esitellään tässä työssä toteutetun järjestelmän toiminta pääpiirteissään, ja luvussa 6 testataan sen toimintaa eri tilanteissa ja pohditaan saatuja tuloksia. Lopussa on yhteenveto luvussa 7.

## 2 Taustaa sisältöperusteisesta kuvahausta

Sisältöperusteinen kuvahaku (content-based image retrieval) soveltaa konenäön menetelmiä digitaalisten kuvien hakuun suurista tietokannoista. Siinä kuvia järjestetään ja haetaan kuvista itsestään laskettavien visuaalisten piirteiden, ei kuviin manuaalisesti liitettyjen *avainsana-annotaatioiden* avulla (Koskela, 2003; Shete & Chavan, 2012). Tarve sisältöperusteisille kuvahakumenetelmille syntyi digitaalisten kuvien määrän kasvaessa rajusti 1990-luvulle tultaessa. Perinteiset avainsana-annotaatioihin perustuvat hakumenetelmät eivät enää sopineet suuriin kuvatietokantoihin, koska kuvan koko sisältöä kuvaavien annotaatioiden laatiminen manuaalisesti on erittäin aikaa vievää ja annotaatioista on vaikea saada riittävän yksityiskohtaisia ja kattavia. Visuaaliset piirteet taas voidaan laskea kuvista automaattisesti, jolloin se on tehokasta, systemaattista ja objektiivista.

Termiä sisältöperusteinen kuvahaku käytti todennäköisesti ensimmäisenä Kato (1992). Ensimmäiset sisältöperusteiset kuvahakujärjestelmät ilmestyivät pian sen jälkeen, ja niitä olivat mm. QBIC (Query By Image Content, Niblack & *al.*, 1993), Chabot (Ogle & Stonebraker, 1995) ja Photobook (Pentland & *al.*, 1996). Sisältöperusteisen kuvahaun tutkimus on lisääntynyt ainakin viime vuosiin asti (Rafiee & *al.*, 2010), ja järjestelmiä on paljon. Sisältöperusteisen kuvahaun menetelmiä voi periaatteessa käyttää missä tahansa kuvasovelluksissa, esim. taidekokeelmissa, diagnoosinteossa lääketieteellisistä kuvista, valvontakameran kuvien analysoinnissa, maantieteellisen informaation käsittelyssä, sodankäyntiin liittyvissä sovelluksissa tai ei-toivotun materiaalin suodatuksessa. Myös videoille ja multimedialle on kehitetty vastaavanlaisia järjestelmiä.

Sisältöperusteisessa kuvahaussa ei yleensä ole tarkoitus esittää kuvan jokaisen pikselin arvoa, vaan kuvasta lasketaan sen sisältöä edustavia piirteitä kuten väri, objektien muodot tai tekstuuri. Piirteistä kerrotaan lisää kohdassa 2.1. Piirteistä muodostetaan jokin *indeksi* eli tietorakenne, josta on tehokasta tehdä hakuja. Tästä kerrotaan kohdassa 2.2. Haussa haetaan tavallisimmin samanlaisia kuvia jollekin syötteenä annetulle esimerkkikuvulle. Kuvien samanlaisuutta mitataan piirteiden perusteella jollakin



etäisyysmitalla. Joissain järjestelmissä on lisäksi mahdollista muokata hakua käyttäjän antaman palautteen perusteella. Hakujen käsittelystä kerrotaan kohdassa 2.3.

## 2.1 Kuvien esittäminen piirteiden avulla

Digitaaliset kuvat koostuvat pikseleistä, joilla on numeroarvot. Värikuvien tapauksessa jokaista pikseliä kohti on erikseen tyypillisesti kolme eri väriarvoa. Kuvatiedostot ovat usein suhteellisen suuria mutta sisältävät redundanttia informaatiota. Tämän vuoksi sisältöperusteisessa kuvahaussa kuvia järjestellään, vertaillaan ja haetaan niiden sisältöä edustavien piirteiden avulla, jotka sisältävät vähemmän dataa kuin itse kuvat.

Piirteet esitetään yleensä *piirrevektoreina*. Yleisen määritelmän mukaan vektorit ovat vektoriavaruuden alkioita, joille on määritelty vektoriavaruuden laskutoimitukset. Piirrevektori on kuitenkin aina myös järjestetty joukko reaalilukuja. Vektoriin kuuluvista luvuista käytetään nimitystä *komponentti*. Jos piirrevektorissa on  $n$  kappaletta komponentteja, se on  $n$ -ulotteisen avaruuden vektori ja eli  $n$ -ulotteinen vektori. Eriyisesti ohjelmoinnin kannalta tämäntyyppisen vektorin voi myös hahmottaa matriisina, jonka leveys on 1 ja korkeus  $n$ . Piirrevektorien komponenttien määrä pyritään saamaan selvästi pienemmäksi kuin numeroarvojen määrä itse kuvissa. Piirteet lasketaan jollakin *piirreirrotusmenetelmällä*. Samasta kuvasta voidaan laskea useampia piirrevektoreita eri piirreirrotusmenetelmillä. Samalla piirreirrotusmenetelmällä saaduilla vektoreilla on aina sama vektoriavaruus (samat ulottuvuudet), ja vain samalla piirreirrotusmenetelmällä saatuja vektoreita voi verrata keskenään ko. vektoriavaruuden laskutoimituksilla.

Piirteet voidaan laskea kuvista globaalisti tai alueittain. Globaali piirre edustaa koko kuvaa, alueesta laskettu piirre ko. aluetta. Helpoin tapa jakaa kuva alueisiin on käyttää ennalta määriteltyjä alueita kaikissa kuvissa, kuten on tehty esim. PicSOM-järjestelmän varhaisessa versiossa (Laaksonen & al., 2000). Tällaiset alueet eivät kuitenkaan vastaa kuvassa olevia sisällöltään erilaisia alueita. Toinen mahdollisuus on jakaa kuvat jonkin piirteen tai piirteiden suhteen homogeenisiin alueisiin esim. *segmentointi-* tai *reunantunnistusmenetelmillä*. Tällaiset menetelmät ovat kuitenkin käytännössä vaikeita toteuttaa täysin automaattisesti ja vaativat käyttäjän osallistu-

mista esim. aloituskohtien valitsemiseen. Näin löytyneet homogeeniset alueet eivät myöskään vielä ole sama asia kuin objektit, koska objektit yleensä koostuvat useista erilaisista sisäisesti homogeenisistä alueista.

Tavallisimmat visuaaliset *piirrelajit*, joita voi laskea kuvista kuvankäsittely- ja kokenäkötekniikoilla, ovat väri, tekstuuri ja muoto. Joka piirrelajista voi tehdä eri piirreirrotusmenetelmillä erilaisia *esityksiä*. Värien esitysmenetelmät perustuvat usein *värihistogrammiin*, jossa esitetään pikselien osuudet ennalta määriteltyjen eri väriarvoalueiden sisällä. Indeksoinnin värihistogrammin mukaan esittelivät ensimmäisinä Swain & Ballard (1991). Koskela (2003) listaa käytettyinä menetelminä *värimomentit* (Stricker & Orengo, 1995), edustavien värien etsimisen ja niiden spatiaalisen informaation esittämisen (Hsu & al., 1995; Smith & Chang, 1995), *värikoherenssivektorit* (Pass & al., 1996) ja *värikorrelogrammit* (Huang & al., 1997).

*Tekstuuri* on ominaisuus, joka aiheutuu alueen värien tai intensiteetin epätasaisuudesta. Tekstuurin esittämiseen on olemassa syntaktisia ja tilastollisia menetelmiä. Syntaktisissa menetelmissä pyritään kuvaamaan sääntöjä, joiden mukaan tekstuurin alkeisyksiköt ovat järjestyneet. Tämä lähestymistapa sopii säännöllisille keinotekoisille tekstuureille. Tilastollisissa esitysmenetelmissä taas kuvataan intensiteettien asettumista jollekin alueelle jollakin kvantitatiivisella arvolla, ja ne sopivat hyvin epäsäännöllisille luonnollisille tekstuureille. Sisältöperusteisessa kuvahaussa tekstuuria on esitetty lähinnä tilastollisilla menetelmillä, joista Koskela (2003) mainitsee *yhteismatriisin* (co-occurrence matrix, Haralick & al., 1973), Tamuran ja kollegojen (1978) esittämät *karkeuden* (coarseness), *kontrastin*, *suunnallisuuden* (directionality), *viivamaisuuden* (line-likeness), *säännöllisyyden* ja *karheuden* (roughness); *multi-resoluutio-simultaaniautoregressiiviset* tekstuurimallit (Mao & Jain, 1992), *Wold-hajotelmaan* perustuvat tekstuuripiirteet (Liu & Picard, 1996), *Gabor-funktiot* (Turner, 1986) ja *aallokkeet* (wavelets, Daubechies, 1990).

Kuvassa olevan objektin muoto kertoo potentiaalisesti paljon kuvan sisällöstä, mutta muoto on vaikeampi esittää kuin väri tai tekstuuri ja siksi vähemmän käytetty sisältöperusteisessa kuvahaussa (Koskela, 2003). Kuvista olisi ensin löydettävä objekteja tai ainakin yhtenäisiä alueita, joiden muotoja esitetään. Kuten edellä kerrottiin, objektien löytäminen kuvista automaattisesti on monimutkainen ongelma. Tämän kier-

tämiseksi on kehitetty myös globaaleja tilastollisia muotopiirteitä, jotka eivät vaadi objektin löytämistä tai edes segmentointia vaan perustuvat koko kuvasta tunnistettuihin reunoihin ja esimerkiksi niiden suuntiin (Brandt & al., 2002). Objektien tunnistamista kuvista muodon perusteella vaikeuttaa myös se, että eri asennoissa olevat objektit näyttävät erimuotoisilta kaksiulotteisessa kuvassa. Muotopiirteitä on pääasiassa kahdenlaisia: ääriviivoihin ja alueisiin perustuvia. Ääriviivoihin perustuvia muotopiirteitä ovat esim. *ketjukoodit* (chain codes, Freeman, 1974), *Fourier-deskriptorit* (Zahn & Roskies, 1972) ja *Wavelet-deskriptorit* (Chuang & Kuo, 1996). Alueisiin perustuvia muotopiirteitä taas ovat esim. *momentti-invariantit* (Hu, 1962), *Zernike-momentit* (Khotanzad & Hong, 1990), sekä yksinkertaiset kuten pinta-ala, *Eulerin luku*, *pyöreys* (circularity), *eksentrisyys*, *pitkulaisuus* (elongatedness) ja *suorakulmaisuus* (rectangularity).

Sisältöperusteisen kuvahaun perustavimpia ongelmia on, että edellä kuvatut laskennalliset visuaaliset piirteet eivät useinkaan vastaa kovin hyvin kuvien semanttista sisältöä, vaikka käyttäjä yleensä ensisijaisesti etsii kuvia juuri tietyistä objekteista tai jopa käsitteistä. Tätä kutsutaan *semanttiseksi kuiluksi*. Sisältöperusteisessa kuvahaussa halutaan usein esittää visuaalisen informaation ohella jonkinlaista semanttista informaatiota. Jos kuville on olemassa avainsana-annotaatioita, niistä voidaan muodostaa joko tekstidokumentteja, joita haetaan perinteisillä tekstihauilla, tai vaihtoehtoisesti binääriarvoisia piirrevektoreita, jossa kunkin mahdollisen avainsanan arvo on joko 1 tai 0 sen mukaan, liittyykö sana kuvaan vai ei.

Toisaalta sisältöperusteisessa kuvahaussa pyritään eroon kuvien työläästä manuaalisesta annotoinnista kehittämällä automaattisia tai puoliautomaattisia menetelmiä semanttisen informaation luomiseksi. Internetistä saaduille kuville voidaan poimia avainsanoja HTML-dokumentista esim. kuvan URL-osoitteesta, kuvan vaihtoehtoisesta tekstistä, kuvaan osoittavan linkin tekstistä tai kuvan kanssa samalla sivulla esiintyvistä tekstistä. Kuvahakujärjestelmiä, jotka käyttävät sekä näin saatua tekstinformaatiota että visuaalisia piirteitä, esittivät ensimmäisten joukossa mm. Smith & Chang (1997), Sclaroff & al. (1999), Aslandogan & Yu (2000) ja Newsam & al. (2001). Semanttista informaatiota on pyritty luomaan esimerkiksi luokittelemalla kuvia automaattisesti muutamaan yksinkertaiseen semanttiseen luokkaan (Frankel & al., 1996; Szummer & Picard, 1998; Vailaya & al., 1998, 2001; Wang & al., 2001).

Lisäksi esimerkiksi Zhang & Chen (2002) ovat kehittäneet järjestelmän, jossa käyttäjä luo annotaatioita osalle kuvista ja järjestelmä ennustaa niiden perusteella annotaatioita muille kuville interaktiivisessa prosessissa. Semanttista informaatiota on pyritty muodostamaan automaattisesti myös käyttäjän antamasta palautteesta avainsanahakujen yhteydessä (Minka & Picard, 1997; Sychay & al., 2002; Koskela & Laaksonen, 2003) tai käyttämällä apuna kuvien esiintymistä samalla web-sivulla tai niiden välisiä linkkejä (Laakso & al., 2001).

## 2.2 Kuvien indeksointi

Kuvatietokannan indeksi tarkoittaa mitä tahansa alkuperäisestä datasta muodostettua tietorakennetta, joka mahdollistaa tehokkaan hakemisen (Koskela, 2003). Indeksini muodostetaan kuvista lasketuista piirrevektoreista. Sisältöperusteisen kuvahaun indeksoinnin ongelma on, että piirrevektorit ovat usein suhteellisen moniulotteisia mutta käyttäjäystävällisyyden kannalta hakujen pitäisi toimia nopeasti. Tehokkuutta voidaan parantaa joko vähentämällä piirrevektorien ulottuvuuksia tai jakamalla datajoukkoa osiin, jotta välttyttäisiin prosessoimasta koko joukkoa yhden haun yhteydessä.

Piirrevektorien ulottuvuuksia voidaan vähentää projisoimalla moniulotteinen vektoriavaruus vähempiulotteiseen vektoriavaruuteen. Tällaisia menetelmiä ovat esimerkiksi *pääkomponenttianalyysi* (Hotelling, 1933), *yksittäisarvohajotelma* (singular value decomposition, Golub & Kahan, 1965), *satunnaisprojektio* (random projection, Kaski, 1998), *moniuulotteinen skaalaus* (multidimensional scaling, Kruskal, 1964) ja *itsenäinen komponenttianalyysi* (independent component analysis, Comon, 1994). Ulottuvuuksien vähentyessä dataa tietenkin aina menetetään jonkin verran. Joskus riittää, että datasta vain vähennetään ulottuvuuksia ja sen jälkeen kaikkia vektoreita voidaan verrata keskenään. Yleensä kuitenkin ulottuvuuksien vähentämisen jälkeen data vielä järjestetään jonkinlaiseen tietorakenteeseen.

Indeksoinnissa kuvajoukkoa tai piirreavaruutta jaetaan usein osiin, jotta haku voitaisiin kohdistaa vain osaan datayksiköistä. Tähän käytettyjä menetelmiä ovat esim. rekursiiviset ositusmenetelmät (recursive partitioning), klusterointi ja ”näköalapaikat” (vantage points) -menetelmä (Koskela, 2003). *Rekursiivisissa ositusmenetel-*

*missä* piirreavaruutta tai datasettiä jaetaan aina kahtia, jolloin siitä syntyy puurakenne. *Klusteroinnissa* tietokannan kuvien vektorit jaetaan ryhmiin eli klustereihin, jotta haussa esimerkkikuvasta laskettua vektoria voitaisiin ensin verrata kunkin klusterin yhteen edustaja-arvoon ja sitten yhden valitun klusterin vektoreihin. ”*Näköalapaikat*”-menetelmässä kuvat järjestetään sen mukaan, kuinka lähellä ne ovat mitäkin valittua kiintopistettä eli ”näköalapaikkaa” (Natsev & Smith, 2002; Vleugels & Veltkamp, 2002).

Yksi sisältöperusteisessa kuvahaussa käytetty indeksointimenetelmä on itseorganisoituva kartta. Sen avulla voidaan muodostaa kaksiulotteinen esitys hyvinkin moniulotteisesta datasta, eli tässä mielessä se on datan ulottuvuuksia vähentävä menetelmä. Itseorganisoituvalla kartalla voi myös jakaa datayksiköt klustereihin. Muista klusterointimenetelmistä poiketen itseorganisoituva kartta kuitenkin myös sijoittaa samanlaiset klusterit lähekkäin. Itseorganisoituvan kartan teoriasta kerrotaan tarkemmin luvussa 3.

### **2.3 Haun käsittely**

Kun kuvat on indeksoitu piirteiden avulla, voidaan tehdä varsinaisia kuvahakuja tietokannasta. Kuvahaun tulosten pitäisi toisaalta vastata sitä mitä haetaan, ja toisaalta haun olisi toimittava suhteellisen nopeasti, jotta järjestelmä olisi käyttäjäväläinen (Koskela, 2003). Sisältöperusteisessa kuvahaussa ei niinkään haeta yhtä tiettyä ennalta määriteltyä kuvaa vaan useita kuvia tiettyjen kriteerien mukaan. Kuvahaku eroaa myös *selauksesta*, jossa haettava kohde ei ole tarkkaan määritelty ja jossa painottuu erityisesti tietokannan kuvien visualisointi käyttäjälle järkevällä tavalla. Jotkut sisältöperusteiset kuvahakujärjestelmät tukevat myös selausta, mutta tässä raportissa keskitytään hakuihin.

Kuvahaku alkaa tyypillisesti käyttäjän antamalla *esimerkkikuvalla*, jolle haetaan samanlaisia kuvia. Esimerkkikuva voi järjestelmästä riippuen olla tietokannasta tai sen ulkopuolelta, joko valmis kuva tai käyttöjärjestelmässä luotava hahmotelma. Joissain järjestelmissä esimerkkikuvia voi myös olla useita ja haussa pyritään keskittymään niiden yhteisiin piirteisiin. Toinen vaihtoehto on esimerkkikuvan sijasta määrittellä suoraan piirteiden arvoja tai valita kuville haluttuja ominaisuuksia kuten tiettyjä vä-

reja. Kolmas mahdollisuus on haku hakusanoilla, joka on usein mielekäs käyttäjälle mutta onnistuu vain järjestelmissä, joissa käytetään visuaalisten piirteiden lisäksi teksti-informaatiota.

Haussa, joka alkaa esimerkkikuvalla, kuvasta lasketaan piirrevektori haun aikana samalla piirreirrotusmenetelmällä kuin tietokannan kuvista laskettiin indeksoinnin alussa, ja esimerkkikuvan piirrevektoria verrataan tietokannan kuvien piirrevektoreihin jollakin etäisyys- tai samanlaisuusmitalla. Tuloksena voidaan antaa esimerkiksi haluttu määrä esimerkkikuvaa lähimpänä olevia kuvia käytetyn etäisyysmitan mukaan (*k nearest neighbors -haku*) tai kaikki korkeintaan tietyn etäisyyden päässä olevat kuvat. Yleisesti käytetyt etäisyysmitat kuten euklidinen etäisyys laskevat etäisyyden aina kerrallaan kahdelle vektorille, joilla on oltava sama vektoriavaruus. Jos järjestelmässä kuvat esitetään erikseen usean eri piirreavaruuden vektoreilla, tulostajoukkoa voidaan esimerkiksi rajata vuorotellen eri piirreavaruuksien mukaan. Toinen mahdollisuus on laskea eri piirreavaruuksissa saaduista etäisyyksistä yhdistetty etäisyysmitta, jossa eri piirteitä voi painottaa eri tavoin (Koskela, 2003).

Monissa järjestelmissä käytetään myös käyttäjän antamaa palautetta tulosten oikeellisuudesta (*relevance feedback*). Siinä käyttäjä merkitsee ensimmäisessä haussa saaduista kuvista relevantit (tai mahdollisesti jakaa ne eri luokkiin relevanssin mukaan), ja tämän tiedon perusteella hakua muokataan parempien tulosten saamiseksi. Hakua ja palautetta voidaan toistaa useita iteraatiokertoja. Haun muokkaamiseen on monia menetelmiä, joita ovat koonneet yhteen esim. Koskela (2003) ja Patil & Kokare (2011). Esimerkkejä menetelmistä ovat esimerkkikuvan piirrevektorin muuttaminen lähemmäs relevantteja hakutuloksia, piirrevektorin komponenttien painotusten muuttaminen etäisyysmittaa laskettaessa, etäisyysmitan vaihtaminen, piirrelajin painotuksen muuttaminen yhdistettyä etäisyysmittaa laskettaessa ja tulosten antaminen klustereista, joissa on eniten relevantteja kuvia.

### 3 Taustaa itseorganisoituvista kartoista

*Itseorganisoituva kartta* (self-organising map, SOM) on Kohosen (1982) kehittämä neuroverkkosovellus. Sen tarkoitus on luokitella dataa ei-ohjatun oppimisen avulla ja esittää moniulotteinen data kaksiulotteisena karttana, jolloin datan väliset yhteydet on helpompi havaita ja ymmärtää. SOM:ejä on käytetty sisältöperusteisessa kuva-haussa datan indeksoimiseen, koska ne sekä vähentävät datan ulottuvuuksia että klusteroivat sitä. Muista klusterointimenetelmistä poiketen SOM myös säilyttää samantyyppiset klusterit lähekkäin. Tämä luku keskittyy SOM:eihin: Kohdassa 3.1 esitellään tarkemmin SOM:n teoria. SOM:ista on myös tehty erilaisia muunnelmia eri sovelluksia varten. Kohdassa 3.2 kerrotaan eräästä tällaisesta SOM:in muunnelmasta eli kasvavasta hierarkkisesta SOM:sta.

#### 3.1 Itseorganisoituvan kartan (SOM) teoria

SOM koostuu neuroneista, jotka on järjestetty yleensä kaksiulotteiseksi esim. neli-kulmion tai heksagonin muotoiseksi ruudukoksi. Aluksi SOM *koulutetaan* jollakin datajoukolla, jonka jokainen *datayksikkö* (esim. kuva) esitetään *syötevektorin* avulla (ks. kohta 2.1). SOM:n jokaisessa neuronissa on *painovektori*, jolla on samat ulottuvuudet kuin syötevektoreilla. Painovektoreille alustetaan ensin esim. satunnaisesti määrätty arvot. Seuraavaksi esitetään SOM:n koulutukseen liittyvät kaavat nykyisessä vakiintuneessa muodossa samoin kuin esim. Rauber & al. (2002) ovat ne esittäneet. Koulutusdatajoukosta lasketuista syötevektoreista valitaan aina satunnaisesti yksi, joka esitellään SOM:lle. Syötevektorin etäisyys lasketaan kaikista SOM:n painovektoreista. Tavallisesti käytetään *euklidista etäisyyttä* (kaava 1)

$$\|x - m_i\| = \sqrt{\sum_k (x_k - m_{ik})^2}, \quad (1)$$

missä  $x$  on syötevektori,  $m_i$  on neuronin  $i$  painovektori ja  $k=1\dots n$  on  $n$ -ulotteisen vektorin komponentin indeksi. Painovektoreista valitaan se, jonka etäisyys syötevektoriin on pienin. Sitä neuronina, jonka painovektori on lähimpänä syötevektoria, kutsutaan *voittajaneuroniksi* (best matching unit, BMU). Tämän jälkeen kutakin SOM:n painovektoria  $m_i$  päivitetään kaavan (2) mukaan

$$m_i(t + 1) = m_i(t) + \alpha(t)h_{ci}(x(t) - m_i(t)), \quad (2)$$

missä  $\alpha(t)$  on ajan funktiona pienenevä *oppimismopeuskerroin* ja  $h_{ci}$  on *naapurustofunktio*. Naapurustofunktiona voi käyttää esim. *Gaussin funktiota* (kaava 3):

$$G_{ci}(t) = \exp - (||r_c - r_i||^2 / (2\delta(t)^2)), \quad (3)$$

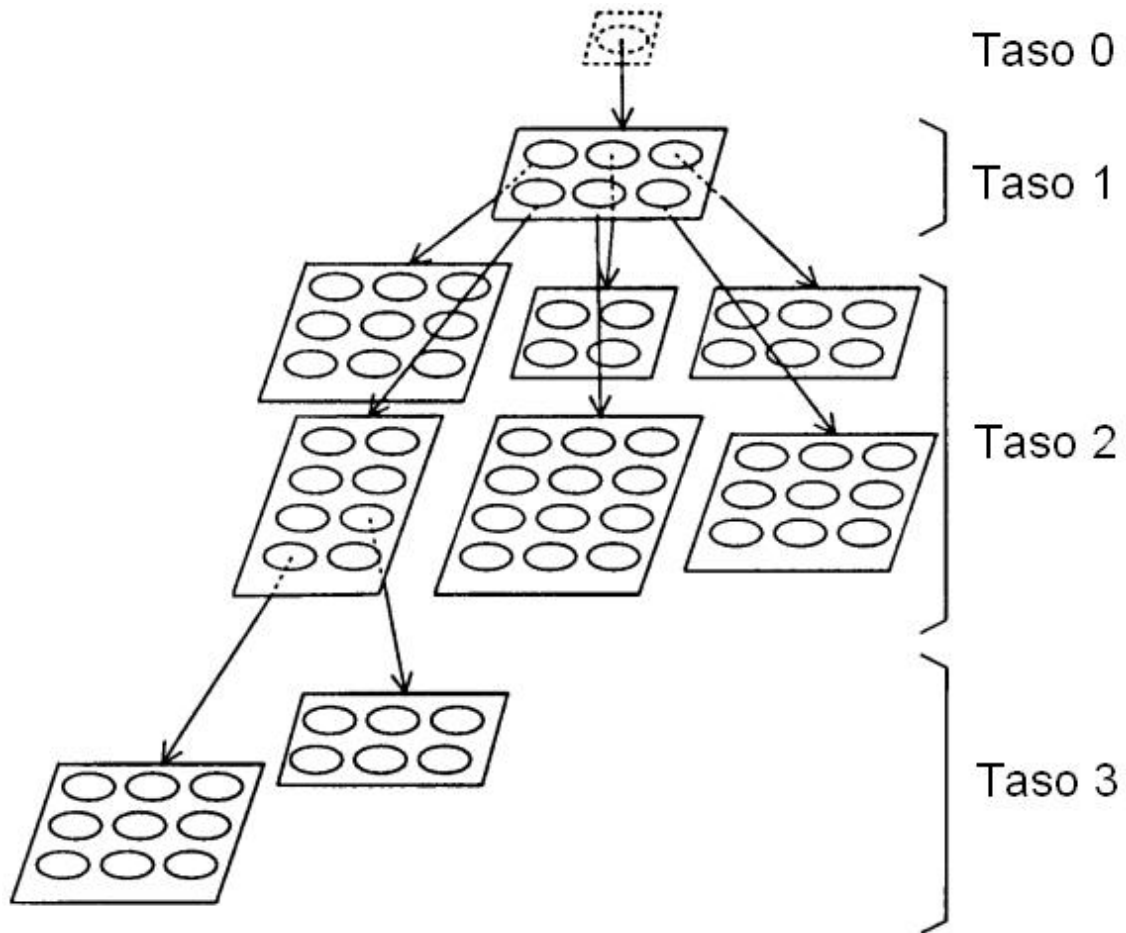
missä  $||r_c - r_i||^2$  kuvaa voittajaneuronin (merkitty kaavassa symbolilla  $c$ ) ja neuronin  $i$  etäisyyttä SOM-ruudukossa ja  $\delta(t)$  on ajan mukaan pienenevä parametri. Kaavat 2 ja 3 voi ymmärtää siten, että neuronin painovektoria päivitetään lähemmäs syötevektoria sitä enemmän, mitä lähempänä ruudukossa se sijaitsee voittajaneuronia. Ajan suhteen pienenevä oppimismopeuskerroin  $\alpha$  (kaava 2) määrää, että koulutuksen alussa painovektoreita päivitetään voimakkaammin kuin koulutuksen lopussa. Ajan suhteen pienenevä parametri  $\delta$  (kaava 3) taas määrää, että koulutuksen alussa päivitetään suuremman alueen painovektoreita ja koulutuksen lopussa taas vain voittajaneuronin tai aivan sen lähiympäristön neuronien painovektoreita. Näitä vaiheita toistetaan koulutuksessa useita iteraatiokertoja.

*Koulutetussa* SOM:ssa samankaltaiset datayksiköt sijoittuvat kartassa lähekkäin oleviin neuroneihin ja erilaiset datayksiköt vastaavasti kaukana toisistaan oleviin neuroneihin. *Kartoitusvaiheessa* katsotaan, mihin kohtaan karttaa uusi datayksikkö asettuu, ja nähdään, mitä koulutusdatayksiköitä se muistuttaa.

### 3.2 Kasvava hierarkkinen itseorganisoituva kartta (GHSOM)

*Kasvava hierarkkinen SOM* (growing hierarchical SOM, GHSOM, Rauber & al., 2002) koostuu useasta kasvavasta SOM:sta eli *alikärtasta*, jotka muodostavat hierarkian kuvan 1 mukaisesti. Se perustuu aikaisempiin kasvaviin SOM-malleihin (Blackmore & Miikkulainen, 1993; Fritzke, 1995; Alahakoon & al., 2000), joissa SOM:n lopullinen neuronien määrä muotoutuu koulutusdatan mukaan. GHSOM muistuttaa myös hierarkkisia SOM:eja (Koikkalainen & Oja, 1990), joissa voittajaneuronin löytäminen nopeutuu, kun sen haku voidaan kohdistaa tiettyyn haaraan.





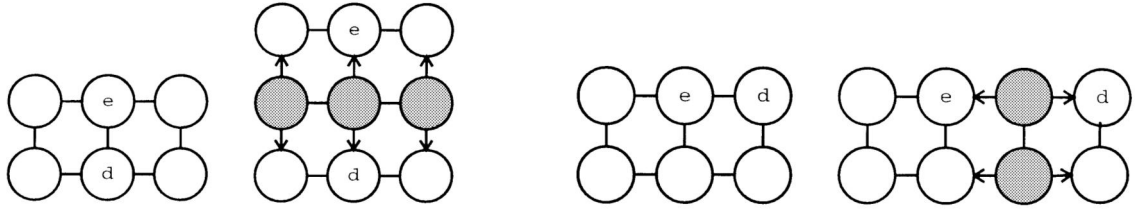
Kuva 1: GHSOM Rauber & al. (2002) mukaillen.

GHSOM:n koulutuksen alussa luodaan yksi  $2 \times 2$  neuronia sisältävä neliömäinen SOM ja alustetaan sen painovektoreille satunnaiset arvot (kuvassa 1 tason 1 alikartta). Tätä SOM:ia koulutetaan kuten tavallista SOM:ia. Aina tietyn määrän  $\lambda$  iteraatiokertoja jälkeen lasketaan kullekin neuronille  $i$  kvantisointivirhe (quantization error,  $q_{e_i}$ ), joka on neuronin  $i$  kartoittuvien syötevektorien  $x_j$  euklidiset etäisyydet neuronin  $i$  painovektorista  $m_i$  yhteen laskettuina:

$$q_{e_i} = \sum \|m_i - x_j\|.$$

Kvantisointivirhe  $q_e$  kertoo, miten vaihtelevaa dataa neuronin kartoittuu. Neuroneista etsitään se, jonka  $q_e$  on suurin. Tässä kohdassa karttaa on siis vaihtelevaa dataa ja tarvitaan lisää neuroneita. Suurimman  $q_e$ :n neuronin painovektoria verrataan sen naapureiden painovektoreihin ja valitaan naapureista se, jonka painovektori on siitä kauimpana. Näiden kahden neuronin väliin luodaan uusi rivi tai sarake neuroneita

kuvan 2 esittämällä tavalla. Uusien neuronien mallivektorit alustetaan niiden naapuri-keskiarvoilla.



**Kuva 2: GHSOM:n alikartan kasvattaminen (Rauber & al., 2002). Suurimman kvantisointivirheen ( $q_e$ ) neuroni on merkitty symbolilla  $e$  ja sen eniten erilainen naapuri symbolilla  $d$ . Uudet neuronit on kuvattu harmaalla ja niiden painovektorien alustuksessa käytetyt naapurit on osoitettu nuolilla.**

SOM:n painovektorien päivityskaavan (kaava 2) oppimisnopeuskerroin  $\alpha$  ja naapurustofunktion ajan suhteen muuttuva parametri  $\delta$  (kaava 3) palautetaan alkuperäisiin arvoihinsa, ja koulutusta jatketaan taas seuraavat  $\lambda$  iteraatiokertaa. Kun karttaan tulee lisää neuroneja, ne sisältävät vähemmän vaihtelua ja koko alikartan *keskimääräinen kvantisointivirhe* (mean quantization error, MQE) pienenee. Alikartan  $s$  MQE lasketaan seuraavasti:

$$\text{MQE}_s = 1/n_U \sum q_{e_i},$$

missä  $n_U$  on kartan niiden neuronien lukumäärä, joihin kartoittuu datayksiköitä. Alikartan kasvatus lopetetaan, kun sen MQE:n arvo alittaa tietyn osuuden  $\tau_1$  sen oman vanhempineuronin  $q_{e_u}$ :stä  $q_{e_u}$ . Tämä *kasvun lopetuskriteeri* on kaavassa 4:

$$\text{MQE}_s < \tau_1 q_{e_u}. \quad (4)$$

Jos kyseessä on tason 1 alikartta, sillä ei ole vanhempineuronia ja  $q_{e_u}$ :n tilalla käytetään koko syötedatan  $q_{e_0}$ :ta  $q_{e_0}$ , joka lasketaan kaavalla (5):

$$q_{e_0} = \sum \|m_0 - x_i\|, \quad (5)$$

missä  $m_0$  on kaikkien syötevektorien keskiarvo. Kaikki koulutusdatan syötevektorit voidaan myös ajatella GHSOM:n ylimmäksi tasoksi 0 ja  $q_{e_0}$  tämän tason ainoan neuronin  $q_e$ :ksi (ks. kuva 1).

Kun kaavan (4) ehto täyttyy, alikartan kasvatus lopetetaan ja tarkastetaan jokaisen neuronin *globaalit lopetuskriteerit* (global termination criteria, kaava 6):

$$q_{e_i} < \tau_2 q_{e_0}.$$

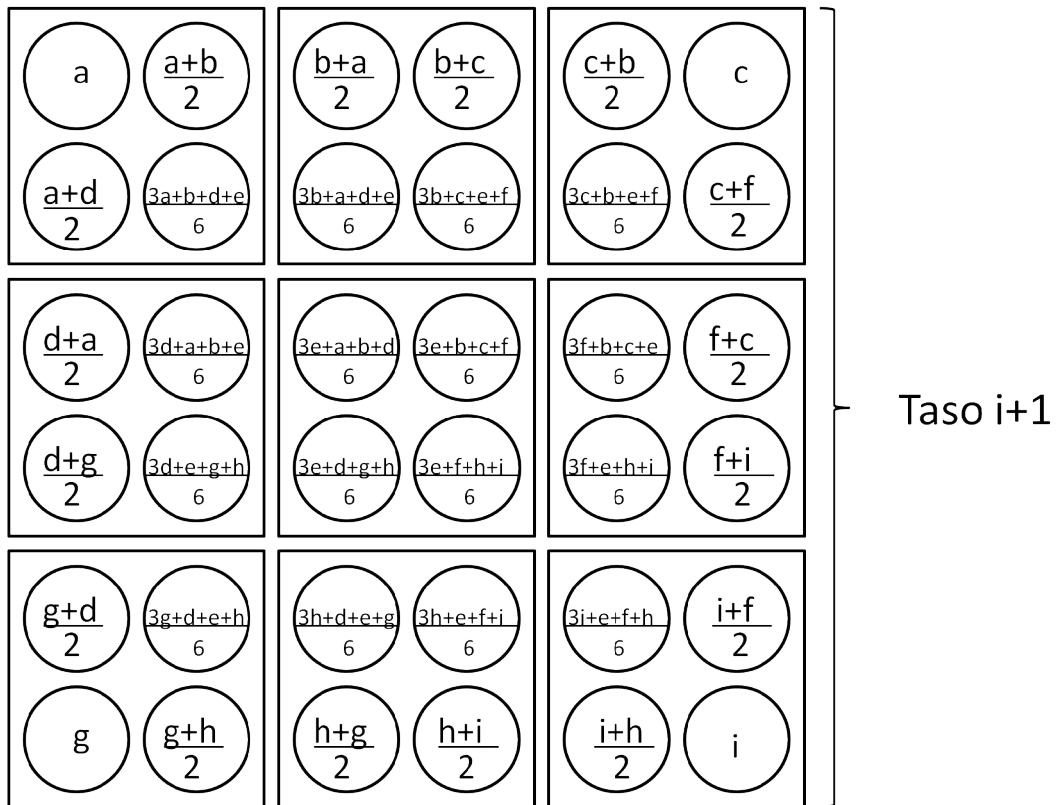
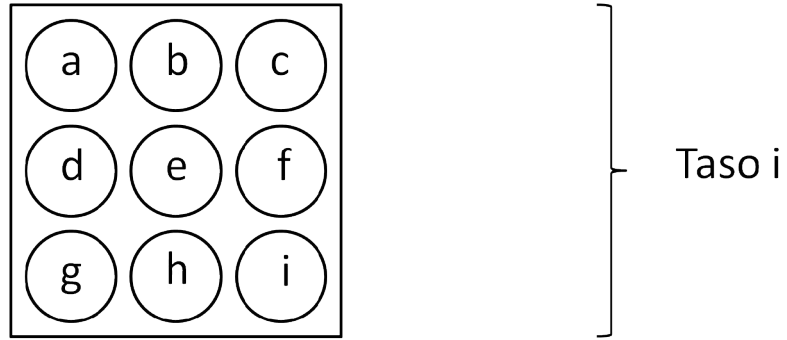
(6)

Kriteerin mukaan neuronin  $q_e$ :n kuvaama vaihtelu saa olla korkeintaan jokin määrätty osuus  $\tau_2$  koko syötedatan kvantisointivirheestä  $q_{e_0}$ :sta (kaava 5). Kullekin neuronille, jossa globaali lopetuskriteeri ei täyty eli  $q_e$  ylittää sallitun arvon, luodaan uusi  $2 \times 2$  lapsikartta uudella tasolla. Uuden kartan neljän neuronin painovektorit alustetaan aina vanhempineuronin ja sen naapureiden painovektorien mukaan kaavoilla, jotka on esitetty kuvassa 3. Tällä pyritään varmistamaan, että myös alemman tason vierekkäisten karttojen lähellä toisiaan oleviin reunoihin kartoittuu samanlaisia datayksiköitä.

Uutta karttaa koulutetaan sen vanhempineuroniin kartoittuneilla syötevektoreilla ja siihen lisätään neuroneita samoin kuin edellä kuvattiin. Näin jatketaan, kunnes kasvun lopetuskriteeri (kaava 4) täyttyy kaikissa alikartoissa ja globaali lopetuskriteeri (kaava 6) täyttyy jokaisessa neuronissa, jolla ei ole lapsikarttoja.

Parametrin  $\tau_1$  valinta vaikuttaa siihen, muodostuuko suurempia alikarttoja vai enemmän hierarkiatasoja. Jos  $\tau_1$  on pieni, alikartta saa kasvaa enemmän ennen kuin sen neuroneille luodaan uusia lapsikarttoja. Parametri  $\tau_2$  puolestaan vaikuttaa koko GHSOM:n lopulliseen kokoon määräämällä, kuinka paljon vaihtelua neuronin sisällä saa olla. Jos arvo on pieni, neuronin  $q_e$  ei helposti alita lopetuskriteerin (kaava 6) määräämää arvoa, vaan lapsikarttoja muodostetaan helposti.

Hierarkkisissa SOM:eissa datan käsittely nopeutuu, koska neuroneita luodaan vain tarvittava määrä, kutakin alemman tason karttaa koulutetaan vain sen vanhempineuroniin kartoittuvilla syötevektoreilla ja voittajaneuronin hakeminen voidaan kohdistaa aina tiettyyn haaraan. GHSOM:n kaikki haarat eivät ole yhtä syviä, vaan tiheämmillä alueilla on enemmän tasoja. Siinä pyritään siihen, että hierarkia muotoutuu datan vaatimusten mukaan, ei tasaisen puurakenteen optimaaliseen tehokkuuteen. Tekijöiden testissä GHSOM:iin tuli vähemmän neuroneja kuin vastaavaan tavalliseen SOM:iin, mutta siitä näki myös datan hierarkkisen rakenteen, eli mitä ylätasoin ryhmiä luokiteltavat asiat muodostivat, ja näiden yksityiskohtaisemman luokittelun. Uusien alikarttojen neuronien alustuksella kuitenkin säilytettiin vierekkäisten alikarttojen vierekkäiset reunat samanlaisina.



**Kuva 3: Uusien alikarttojen neuronien painovektorien alustus Dittenbach & al. (2001) mukailen. Kaavat on muutettu tässä työssä käytetyn GHSOM Toolbox -kirjaston version 5 lähdekoodin (Pampalk & Chan, 2002) mukaiseksi. Kirjainsymbolit merkitsevät tason i neuronien painovektoreita.**

## 4 Muita järjestelmiä, jotka assosioivat piirteitä SOM:ien avulla

*Assosiaatio* tarkoittaa yleensä mielessä olevaa yhteyttä joidenkin asioiden välillä. Aivot muodostavat assosiaatioita mm. eri lajin aistiärsykkeiden (esim. näkö, kuulo, tunto jne.) välille, joita prosessoidaan eri aivoalueilla. Tällaisista assosiaatioista on apua tilanteissa, joissa jonkin aistilajin tietoa on saatavilla rajallisesti tai ei ollenkaan. Esimerkiksi esineen tunnustelu taskussa voi herättää näköaistimuksen simulointia, joka näkyy aktivaationa näköaivokuoressa, ja mieleen tulee yleensä jokseenkin oikeanlainen kuva tunnusteltavasta esineestä, vaikka siitä ei saada varsinaista näköinformaatiota. Saatavilla olevan aistilajin avulla voidaan siis ennakoida puuttuvan piirrelajin tietoa.

Sisältöperusteisissa kuvahakujärjestelmissä käytetään usein erillisiä alijärjestelmiä eri piirrelajeille. Näistä lasketaan usein yhteinen tuloste jonkin funktion avulla, mutta niiden välille on harvemmin muodostettu varsinaisia assosiaatioita. Myös sisältöperusteisissa kuvahakujärjestelmissä jonkin piirrelajin tieto voi olla puutteellista, mutta järjestelmä voisi oppia tuottamaan sitä assosiaatioiden avulla muunlajisesta saatavilla olevasta tiedosta. Kuville on esimerkiksi helppo laskea visuaalisia sisältöä kuvaavia piirteitä, mutta nämä kuvaavat huonosti kuvan semanttista sisältöä. Avainsana-annotaatiot taas kertovat hyvin kuvan sisällöstä, mutta niitä on raskas tuottaa. Assosiaatioiden avulla visuaalista ja semanttista tietoa olisi mahdollista yhdistää toisiinsa ja esimerkiksi avainsana-annotaatioita voitaisiin tuottaa automaattisesti visuaalisten piirteiden avulla.

Seuraavassa esitellään kolme lähestymistapaa, joissa kaikissa assosioidaan yhdessä esiintyviä eri piirrelajeja SOM:ien välisillä yhteyksillä. Niissä luokiteltavat datayksiköt (esim. kuvat) esitetään kahdella tai useammalla eri lajin piirrevektorilla, ja kullakin piirrelajilla koulutetaan erikseen oma SOM:nsa. Yhteyksien kautta datayksiköiden pitäisi olla tunnistettavissa myös sen piirrelajin avulla, jolla SOM:ia ei alun perin koulutettu. Esiteltyjen järjestelmien alkuperäiset sovellusalueet vaihtelevat, mutta ne ovat periaatteessa sovellettavissa myös sisältöperusteiseen kuvahakuun. Assosiaatiot on toteutettu eri järjestelmissä eri tavoin. Kohdassa 4.1 esitellään järjestelmä, jossa

yhden SOM:n kullekin neuronille määritetään yksi yhteys johonkin toisen SOM:n neuroniin (Yang & al., 2009). Kohdassa 4.2 on vastaavanlainen järjestelmä, mutta siinä kahden SOM:n neuroneista muodostetaan Hebbin neuroverkko (Ahmad & al., 2003). Lopuksi kohdassa 4.3 tutustutaan järjestelmään, jossa SOM:ien assosiaatiot on toteutettu käyttämällä joka neuronissa pääasiallisen painovektorin lisäksi yhtä tai useampaa sivupainovektoria (Johnsson & Balkenius, 2008, 2009).

#### **4.1 Neuronille parhaan parin toiselta SOM:lta etsivä järjestelmä**

Yang & al. (2009) kehittivät järjestelmän, jossa piirteiden väliset assosiaatiot toteutetaan yhdistämällä yhden piirrelajin SOM:n jokainen neuronin sitä parhaiten vastaavaan neuroniin toisen piirrelajin SOM:lla. Valmiilla järjestelmällä voidaan kartoittaa datayksikkö, kuten kuva, yhden piirrelajin mukaan yhdelle SOM:lle ja seurata sieltä yhteyttä toiselle SOM:lle, josta saadaan datayksikköön liittyvää toisen piirrelajin tietoa. Järjestelmän alkuperäisenä sovelluksena oli luoda kuville avainsana-annotaatioita automaattisesti assosioimalla kuvien visuaalisia piirteitä avainsanoihin, joita usein esiintyy niitä ympäröivässä tekstissä. Järjestelmän SOM:t ovat GHSOM:eja.

Järjestelmässä muodostetaan ensin kaksi eri piirrelajin GHSOM:ia, jotka koulutetaan samalla joukolla datayksiköitä, esim. kuvia. Sen jälkeen muodostetaan yhteydet toisesta GHSOM:sta toiseen. Piirrelajit voisivat tietenkin olla mitä tahansa, ja yhteydet voitaisiin periaatteessa muodostaa ensin yhdeltä SOM:lta toiselle ja sitten toisin päin. Seuraavassa kuvataan kuitenkin selvyuden vuoksi yhteyksien muodostaminen Yangin ja kollegojen (2009) alkuperäisen esimerkin mukaisesti visuaalisesta GHSOM:sta annotaatio-GHSOM:iin.

- 1) Käsitellään kaikki visuaalisen GHSOM:n neuronit yksi kerrallaan.
- 2) Aina kun siirrytään käsittelemään uutta neuronin, annetaan kullekin annotaatio-GHSOM:n neuronille vertailuluvuksi 0.
- 3) Otetaan visuaalisen GHSOM:n käsiteltävästä neuronista kaikki siihen kartoituneet koulutuskuvajoukon kuvat ja muodostetaan niistä kaikki mahdolliset parit.

- 4) Katsotaan kustakin kuvaparista, mihin ne kartoittuvat annotaatio-GHSOM:lla. Kummallekin kuvalle löytyy annotaatio-GHSOM:lta voittajaneuroni (joka voi erikoitapauksessa olla sama molemmille kuville).
- 5) Molempien voittajaneuronien (tai yhteisen voittajaneuronin) vertailulukua kasvatetaan yhdellä.
- 6) Lisäksi etsitään lyhin polku näiden kahden neuronin välillä ja kasvatetaan kaikkien muiden polulla olevien neuronien vertailulukua arvolla  $1/(|P|-1)$ , missä  $|P|$  on polun pituus.
- 7) Kun kaikki visuaalisen GHSOM:n käsiteltävänä olevan neuronin kuvaparit on käyty läpi, etsitään se annotaatio-GHSOM:n neuroni, jonka vertailuluku on suurin. Tämä yhdistetään siihen visuaalisen GHSOM:n neuroniin, jota olttiin käsittelemässä.
- 8) Sitten siirrytään käsittelemään seuraavaa visuaalisen GHSOM:n neuronია, ja tehdään sille vaiheet 2-7. Näin jatketaan, kunnes kaikki visuaalisen GHSOM:n neuronit on käyty läpi.

Yang ja kollegat (2009) eivät ota kantaa siihen, että käytettäessä GHSOM:eja sama kuva kartoittuu usein moneen eri tason neuroniin kummallakin GHSOM:lla. Todennäköisesti yhteyksiä ei käytännössä muodostetakaan kaikille visuaalisen GHSOM:n neuroneille vaan ainoastaan kaikille sellaisille, joilla ei ole lapsikarttoja. Kohdassa 4 voittajaneuroniksi valitaan oletettavasti alimman tason voittajaneuroni annotaatio-GHSOM:lla.

Alkuperäisessä (Yang & *al.*, 2009) esimerkissä jokaiselle visuaalisen GHSOM:n neuronille halutaan vain yksi yhteys annotaatio-GHSOM:n neuroniin. Joskus suurin vertailuluku voi olla useammalla kuin yhdellä annotaatio-GHSOM:n neuronilla, mutta ”tasapeli” halutaan ratkaista valitsemalla näistä yksi esim. viereisten neuronien vertailulukujen mukaan tai satunnaisesti. Poikkeuksen muodostavat sellaiset visuaalisen GHSOM:n neuronit, joihin ei kartoitu yhtään kuvaa. Tällaiselle neuronille ei teorian mukaan voi tehdä yhteyksiä. Siinä erikoistapauksessa, että visuaalisen GHSOM:n neuroniin liittyy vain yksi kuva, ei myöskään voida muodostaa teorian mukaisesti kuvapareja mutta yhteys voidaan muodostaa siihen annotaatio-GHSOM:n

neuroniin, johon tämä ainoa kuva liittyy. Tehtäessä yhteyksiä visuaalisesta GHSOM:sta annotaatio-GHSOM:iin annotaatio-GHSOM:n yksittäinen neuronin voi liittyä useaan tai ei yhteenkään visuaalisen GHSOM:n neuronin.

Annotaatioiden muodostaminen uusille kuville visuaalisten piirteiden perusteella tapahtuu seuraavasti: Kuvasta lasketaan visuaaliset piirteet ja se kartoitetaan visuaaliselle GHSOM:lle. Visuaalisen GHSOM:n voittajaneuronista seurataan yhteyttä annotaatio-GHSOM:n yhteen neuronin, ja siitä otetaan annotaatiot. Jos yhteydet tehtäisiin toiseen suuntaan, kuvia voitaisiin periaattessa hakea avainsanoilla visuaaliselta GHSOM:lta.

Yang & al. (2009) testasivat järjestelmää University of Washington's Groundtruth image annotation database -tietokannan manuaalisesti annotoiduilla kuvilla. Kuvia oli 1109, ja niistä 80 % käytettiin koulutukseen ja 20 % testaukseen. Kuvista lasketut visuaaliset piirteet ja annotaatiot luokiteltiin kumpikin omalla GHSOM:llaan, ja yhteydet muodostettiin visuaalisesta GHSOM:sta annotaatio-GHSOM:iin. Kun testaukseen valituille kuville tuotettiin järjestelmän avulla annotaatiot automaattisesti edellä kuvatulla tavalla, saatiin noin 70 % niistä annotaatiosta, joita kuville oli annettu manuaalisesti. Assosiaatiot näyttävät siis toimivan kuvatulaisessa sovelluksessa melko hyvin. Lisäksi kun Yang & al. (2009) hakivat automaattisesti annotoitujen kuvien joukosta kuvia 21:llä satunnaisesti valitulla avainsanalla, hakutuloksista 42 % oli oikeanlaisia, ja kaikista oikeanlaisista haettavista kuvista löytyi 53 %.

## **4.2 SOM:eja Hebbin neuroverkolla yhdistävä järjestelmä**

Ahmad & al. (2003) esittelivät järjestelmän, jossa myös luokitellaan datayksiköt (esim. kuvat) kahdella eri piirrelajin SOM:lla ja SOM:ien neuroneista muodostetaan yhteyksiä toisen SOM:n neuroneihin. Yhteydet kuitenkin poikkeavat edellisestä: Ne toteutetaan *Hebbin neuroverkon* (Hebb, 1949) muunnelmalla. Yhden SOM:n kaikista neuroneista on yhteys kaikkiin toisen SOM:n neuroneihin, mutta eri yhteyksillä on eri vahvuudet, jotka muotoutuvat SOM:ien koulutuksen aikana. Valmiissa järjestelmässä syöte aktivoi oman piirrelajinsa SOM:n neuroneja, jotka taas aktivoivat yhteyksien kautta toisen SOM:n neuroneja. Periaatteessa tätäkin järjestelmää olisi mahdollista soveltaa samoin kuin Yangin ja kollegojen (2009) vastaavaa. Tässä esiteltä-



vän järjestelmän tarkoituksena oli kuitenkin alun perin lähinnä mahdollistaa hakuja useammalla kuin yhdellä tavalla ja siten parantaa hakutuloksia.

Järjestelmään muodostetaan kaksi SOM:ia, joiden neuronien määrä ja muoto määrätään alussa. SOM:t voivat olla keskenään erikokoisia mutta ne eivät saa olla hierarkisia. Molempien SOM:ien koulutus ja yhteyksien vahvuuksien oppiminen tapahtuvat limittäin samanaikaisesti. Kustakin datayksiköstä (esim. kuvasta) muodostetaan kaksi erilajista piirvektoria (esim. laskennalliset visuaaliset piirteet ja ympäristöstä poimittu teksti), joista toinen esitetään toiselle ja toinen toiselle SOM:lle. Molemmista SOM:eista etsitään voittajaneuroni ja niiden painovektoreita päivitetään vastaavasti kuin kaavassa (2).

Joka kerran kun kuva kartoittuu jonnekin kummallakin SOM:lla, neuroneille tulee aktiivisuuksia. Neuronin  $i$  aktiivisuus  $y$  lasketaan kaavasta

$$y_i = \exp(-\|r_c - r_i\|^2),$$

missä  $\|r_c - r_i\|^2$  on neuronin  $i$  euklidinen etäisyys voittajaneuronista ( $c$ ). Suurin aktiivisuuden arvo 1 on kummankin SOM:n voittajaneuronilla, ja aktiivisuus pienenee mentäessä kauemmas voittajaneuroneista. Kuten edellä mainittiin, yhden SOM:n kaikki neuronit ovat yhteydessä kaikkiin toisen SOM:n neuroneihin. Kunkin yhteyden painoa muutetaan kummankin neuronin aktiivisuuden mukaan Hebbin (1949) teoriaan perustuen (Ahmad & al. eivät esitä tarkkaa kaavaa). Neuronien välille muodostuu vahva yhteys, jos ne ovat aktiivisia samaan aikaan. Kullakin iteraatiokerralla eniten vahvistuvat siis voittajaneuronien yhteydet, mutta myös ympäristön neuronien yhteyksiä toisen SOM:n voittajaneuronin naapureihin vahvistetaan jonkin verran. SOM:ien ja yhteyksien koulutusta jatketaan yhtä kauan.

Kahden eri piirrelajin SOM:n ja niiden välisten yhteyksien käytöllä on tarkoitus saada aikaan tarkempia hakutuloksia kuin jos käytettäisiin vain yhtä SOM:ia. Esimerkiksi uusi kuva halutaan saada osumaan SOM:lla samanlaisten kuvien joukkoon. Tunnistusvaiheessa on mahdollista käyttää ensin yhtä SOM:ia. Ne datayksiköt, esim. kuvat, joiden luokittelu epäonnistuu tällä tavalla, voidaan kartoittaa toiselle SOM:lle ja seurata sieltä yhteyksiä ensimmäiselle SOM:lle. Yhteyksien seuraaminen tapahtuu siten, että esim. kuvan esittäminen toiselle SOM:lle aiheuttaa sille tietynlaisen akti-

vaatiokuvion, joka taas aiheuttaa yhteyksien kautta tietynlaisen aktivaatiokuvion toiselle SOM:lle. Eniten aktivoituvista neuroneista pitäisi löytyä eniten samanlaisia kuvia.

Ahmad & al. (2003) testasivat järjestelmää 66 rikospaikkakuvalla. He jakoivat kuvat manuaalisesti kahdeksaan luokkaan ja valitsivat joka luokasta yhden kuvan testausta varten. Järjestelmä koulutettiin jäljelle jääneillä 58 kuvalla: yksi SOM visuaalisilla piirteillä ja toinen kuvaa ympäröivästä tekstistä poimituista avainsanoista muodostetuilla vektoreilla. SOM:ien välille muodostettiin yhteydet edellä kuvatulla tavalla. Ahmad & al. (2003) testasivat, osuvatko testaukseen valitut kuvat SOM:lla samaan kohtaan kuin samaan luokkaan sijoitetut koulutuskuvaajoukon kuvat. Yksittäisillä SOM:eilla noin puolet kuvista (4/8 tai 5/8) osuivat oikeaan kohtaan. Kun väärin osuneet kuvat kartoitettiin ensin toiselle SOM:lle ja yhteyksien kautta takaisin ensimmäiselle SOM:lle, suurin osa kuvista (3/4 tai 2/3) osui oikeisiin kohtiin. Jos taas visuaalinen ja avainsanatieto esitettiin yhdessä vektorissa ja luokiteltiin vain yhdellä SOM:lla, haku toimi testissä vielä huonommin kuin kumpikaan yksittäisistä SOM:eista, sillä vain 3/8 kuvaa osui oikeisiin kohtiin. Kahden erilajisen SOM:n käyttö siis paransi hakutuloksia, kuten oli tarkoituskin.

Tekijöiden (Ahmad & al., 2003) testissä kaikki koulutuskuvaajoukon kuvat osuivat suunnilleen samaan kohtaan seurattaessa yhteyttä avainsana-SOM:sta visuaaliseen SOM:iin kuin kartoitettaessa suoraan visuaaliselle SOM:lle. Testattaessa toiseen suuntaan kaikki paitsi yksi kuva osuivat lähekkäin. Tässä mielessä muodostuneet assosiaatiot todella kuvaavat hyvin datassa esiintyvien piirteiden assosiaatioita. Edellä esitetystä testeistä voi päätellä, että koulutuskuvaajoukon ulkopuoliset kuvat puolestaan kartoituivat usein eri kohtiin SOM:ia suoraan SOM:n omalla piirrelajilla kuin toisella piirrelajilla yhteyksien kautta. Esitetyn sovelluksen kannalta se oli kuitenkin hyödyllinen ominaisuus, koska näin saatiin kaksi vaihtoehtoista osumaa, joista edes toinen saattoi olla oikea. Toisaalta yhteyksien rooli hakutulosten parantamisessa on hieman epäselvä, sillä haut olisi periaatteessa mahdollista tehdä erikseen kummankin SOM:n avulla myös ilman yhteyksiä. Tuloksista käy kuitenkin ilmi, että yhteyksiä on ainakin mahdollista käyttää järjestelmässä ilman että ne huonontavat hakutuloksia merkittävästi.

### 4.3 Sivupainovektoreita käyttävä assosiatiivinen SOM

Johnssonin ja Balkeniuksen (2008) *assosiatiivinen SOM* (A-SOM) muistuttaa edellä esitettyjä järjestelmiä siinä, että sekin assosioi yhden piirrelajin syötteen aiheuttaman aktivaation toisen piirrelajin syötteen aiheuttamaan aktivaatioon. Tässä järjestelmässä ei kuitenkaan ole konkreettisia eri SOM:iien neuronien välisiä yhteyksiä, vaan assosiaatiot toteutetaan käyttämällä neuroneissa pääasiallisten painovektorien lisäksi *sivupainovektoreita*, joita myös päivitetään syötteen mukaan. A-SOM:n on tarkoitus tunnistaa datayksiköt sekä *pää-* että *sivusyötteellä* ja myös niiden yhdistelmällä. Järjestelmällä ei ole yhtä ilmeistä sovellustarkoitusta kuin kahdella edellä esitetyllä, vaan siinä on lähinnä tarkoitus mallintaa aivoissa tapahtuvaa eri aistien yhteistoimintaa.

Tässä esitellään A-SOM:n alkuperäinen teoria (Johnsson & Balkenius, 2008). A-SOM on kaksiulotteinen ruudukko, jonka koko ja muoto määrätään alussa. Yksittäinen A-SOM ei myöskään voi olla hierarkkinen. Jokaisessa A-SOM:n neuronissa on *pääpainovektorin* lisäksi sivupainovektori. Koulutuksen jokaisella iteraatiokerralla jokaiselle neuronille esitetään *pääsyötevektori* ja *sivusyötevektori*. Kuten tavallisessa SOM:ssa, A-SOM:n neuroneista valitaan se, jonka pääpainovektori on lähimpänä pääsyötevektoria jonkin etäisyysmitan mukaan. Neuronien pääpainovektoreita päivitetään samoin kuin tavallisessa SOM:ssa kaavan (2) mukaan siten, että syötevektorina käytetään pääsyötevektoria ja naapurustofunktiona Gaussin funktiota (kaava 3).

Assosiaatiot toteutetaan sivupainovektorien päivityksen avulla. Sivupainovektoreita päivitetään kaavan (7) mukaan:

$$m_i^b(t+1) = m_i^b(t) + \beta x^b(t)[y_i^a(t) - y_i^b(t)], \quad (7)$$

missä  $m_i^b$  on neuronin  $i$  sivupainovektori,  $\beta$  on päivityksen voimakkuutta kuvaava vakio,  $x^b$  on sivusyötevektori,  $y_i^a(t)$  on neuronin  $i$  pääaktiivisuus ja  $y_i^b(t)$  on neuronin  $i$  sivuaktiivisuus. Sivupainovektoreita päivitetään sitä voimakkaammin, mitä kauempana pää- ja sivuaktiivisuus ovat toisistaan. *Pääaktiivisuus*  $y_i^a(t)$  lasketaan Gaussin funktiosta (kaava 3). Suurin pääaktiivisuuden arvo on siis voittajaneuronilla, ja pääaktiivisuus pienenee vähitellen siirryttäessä SOM:n rakenteessa kauemmas voittajaneuronista. *Sivuaktiivisuus*  $y_i^b(t)$  lasketaan kaavasta

$$y_i^b(t) = x^b(t) \cdot m_i^b(t),$$

eli se on sivusyötevektorin ja sivupainovektorin *pistetulo*. Pistetulo on sitä suurempi, mitä suurempia arvoja vektorien komponenteilla on. Jos pää- ja sivuaktiivisuus ovat lähellä toisiaan, sivupainovektoria ei juurikaan muuteta kaavassa (7). Jos pääaktiivisuus on suuri (eli neuronin on pääsyötteen mukaan voittajaneuronin tai lähellä sitä) mutta sivuaktiivisuus on pieni, kaavan (7) vähennyslaskun tulos on suuri ja sivupainovektorin komponenttien arvoja kasvatetaan suhteellisen voimakkaasti. Tämän seurauksena neuronin sivuaktiivisuudet kasvavat. Jos taas pääaktiivisuus on pieni (eli neuronin on kaukana pääsyötteen mukaisesta voittajaneuronista), kaavan (7) vähennyslaskun tulos on negatiivinen ja sivupainovektorin komponenttien arvoja pienennetään. Samalla neuronin sivuaktiivisuudet pienenevät. Neuronille  $i$  voidaan myös laskea *kokonaisaktiivisuus*  $y_i$ , joka on pää- ja sivuaktiivisuuksien keskiarvo:

$$y_i = ([y_i^a(t) + y_i^b(t)]/2).$$

A-SOM:n uudemmassa muunnelmassa (Johnsson & Balkenius, 2009) sivusyötteitä voi olla useita ja jokaista vastaa oma sivupainovektorinsa. Tässä muunnelmassa pää- ja sivuaktiivisuuksien kaavat myös poikkeavat hieman edellä esitetyistä.

Koulutuksen jälkeen A-SOM on luokitellut koulutusdatan pääsyötteen avulla ja lisäksi sivusyötteen avulla. Datayksiköiden pitäisi kartoittaa samassa A-SOM:ssa jokseenkin samoihin kohtiin pää- ja sivusyötteellä sekä näiden yhdistelmällä. Annettaessa A-SOM:lle sen pääsyötettä neuronien pääaktiivisuuksista muodostuu jonkinlainen jakauma. Annettaessa samasta datayksiköstä saatua sivusyötettä sivuaktiivisuuksista pitäisi muodostua samankaltainen jakauma. Vastaavasti kokonaisaktiivisuusjakaumasta pitäisi muodostua samankaltainen käytettäessä datayksiköstä pää- ja sivusyötteen yhdistelmää. Lisäksi eri datayksiköiden toivotaan aktivoivan A-SOM:lla erillisiä alueita, jotta A-SOM:n voisi sanoa erottavan datayksiköt toisistaan.

Johnsson & Balkenius (2008) testasivat järjestelmää mittaamalla kahdeksasta esineestä kovuus- ja karheustietoa erilaisten sensorien avulla. He muodostivat kaksi A-SOM:ia, joista toisessa oli pääsyötteenä tekstuuri ja sivusyötteenä kovuus, toisessa päinvastoin. SOM:t koulutettiin näistä kahdeksasta testiesineestä saaduilla useilla mittaustuloksilla. Järjestelmää testattiin samoilla esineillä, joilla se koulutettiin. Kun

sille A-SOM:lle, jonka pääsyöte oli tekstuuri, annettiin testiesineistä vain tekstuuritietoa, eri esineet sijoittuvat yleensä melko selvästi omiin kohtiinsa SOM:lla pääaktiivisuuksien mukaan. Kun samalle A-SOM:lle annettiin sekä teksturi- että kovuustietoa, esineet sijoittuivat jokseenkin samoihin kohtiin kuin annettaessa pelkkää tekstuuritietoa. Kun tälle A-SOM:lle annettiin pelkkää sivusyötettä eli kovuustietoa, esineet sijoittuivat edelleen melko samoihin kohtiin kuin annettaessa tekstuuritietoa. Tosin sivusyötteen mukaan esineet eivät erottuneet toisistaan yhtä selvästi kuin pääsyötteen mukaan. Kovuustietoa saaneesta A-SOM:sta saatiin vastaavanlaiset tulokset. Tosin esineet eivät erottuneet tällä A-SOM:lla aivan yhtä hyvin toisistaan kuin toisella edes pääsyötteen mukaan. Kaiken kaikkiaan A-SOM:t siis ”tunnistivat” esineet melko hyvin niin pääsyötteen, sivusyötteen kuin niiden yhdistelmänkin avulla, joten se oppi assosioimaan yhdessä esiintyviä piirteitä toisiinsa.

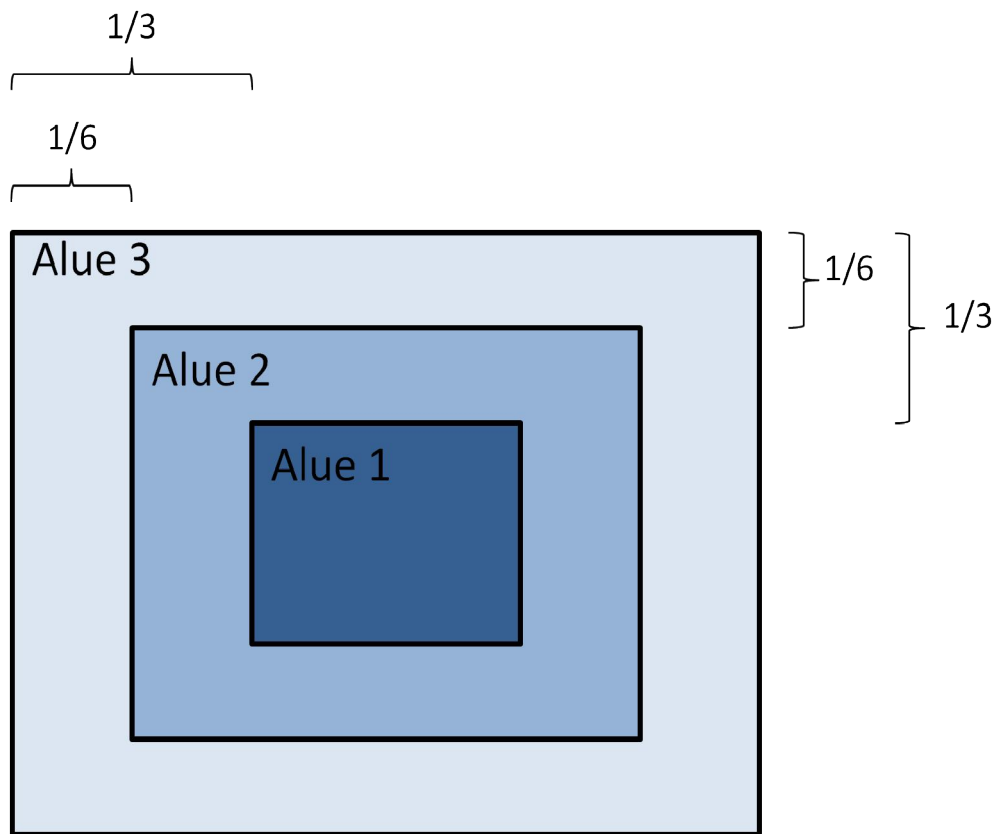
## **5 Piirteitä assosioivan sisältöperusteisen kuvahakujärjestelmän kuvaus**

Tässä luvussa on tarkoitus kuvata uusi sisältöperusteisen kuvahaun toteuttava järjestelmä, jossa käytetään kahta GHSOM:ia ja niiden välille muodostettavia yhteyksiä. Toinen GHSOM luokittelee kuvat värin ja toinen tekstuurin perusteella. Järjestelmä muodostaa GHSOM:ien neuronien välille yhteyksiä sen mukaan, mitkä värit ja tekstuurit esiintyvät yhdessä koulutusdatan kuvissa. Koulutettujen SOM:ien avulla voidaan hakea esimerkkikuvalla samanlaisia kuvia värin, tekstuurin tai molempien perusteella. Lisäksi voidaan hakea kuvan väreihin yleisesti liittyviä tekstuureja edustavia kuvia tai kuvan tekstuureihin yleisesti liittyviä värejä edustavia kuvia.

Tämä luku jakautuu kohtiin seuraavasti: Kohdassa 5.1 kerrotaan, miten järjestelmä laskee kuvista väri- ja tekstuuripiirteet. Kohdassa 5.2 puolestaan kerrotaan GHSOM:eista, joita järjestelmä käyttää kuvien luokitteluun näiden piirteiden avulla. Yhteyksien muodostaminen ja haut perustuvat kuvien kartoitukseen koulutetuille GHSOM:eille. Kohdassa 5.3 esitetään periaate, että kukin kuva kartoitetaan koulutetussa GHSOM-rakenteessa yhteen, alimman mahdollisen tason alikartan neuroniiin. Kohdassa 5.4 kerrotaan, miten järjestelmä muodostaa yhteydet GHSOM:ien välille. Lopuksi kohdassa 5.5 esitellään erilaiset kuvahaut, joita järjestelmällä voi tehdä.

### **5.1 Piirteiden laskeminen**

GHSOM:ien koulutuksien alussa järjestelmä laskee koulutuskuvajoukon kuvista piirteet. Myöhemmin haettaessa kuvia se laskee piirteet esimerkkikuvasta samalla tavoin. Piirteet lasketaan kolmesta kiinteästä alueesta, jotka näkyvät kuvassa 4. Ensimmäinen alue on suorakulmio, jonka leveys on  $1/3$  koko kuvan leveydestä ja korkeus  $1/3$  koko kuvan korkeudesta ja joka sijaitsee tarkalleen keskellä koko kuvaa. Toinen alue on hieman suurempi suorakulmio, johon sisältyy myös ensimmäinen alue (sisin suorakulmio). Toisen alueen leveys on  $4/6$  koko kuvan leveydestä ja korkeus  $4/6$  koko kuvan korkeudesta, ja sekin sijaitsee tarkalleen koko kuvan keskellä. Kolmas on koko kuva itse, ja siihen sisältyvät molemmat sisemmät alueet. Jako perustuu oletukseen, että kuvan tärkeimmät asiat sijaitsevat keskellä kuvaa.



**Kuva 4: Kuvan jako alueisiin piirteiden laskemista varten.**

Järjestelmä tarkistaa, että kuvan väriavaruus on aluksi RGB (Wikipedia, 2013b), ja vaihtaa sen  $L^*a^*b^*$ :ksi (Wikipedia, 2013a) ennen väripiirteiden laskemista. Järjestelmä laskee joka alueen pikseleistä valoisuutta kuvaavan L-komponentin keskiarvon, paikkaa vihreä-magenta-akselilla kuvaavan a-komponentin keskiarvon ja paikkaa sininen-keltainen-akselilla kuvaavan b-komponentin keskiarvon. Tuloksena syntyy kustakin kuvasta  $3 \times 3 = 9$ -ulotteinen vektori. Näin saadut vektorit ovat kuvien värejä edustavia piirrevektoreita ja niitä käytetään väri-GHSOM:n koulutukseen (kohta 5.2) sekä hauissa, joissa verrataan esimerkkikuvan väriä koulutus kuvajoukon väreihin (kohta 5.5).

Tekstuuripiirteet lasketaan myös kolmesta edellä kuvatusta alueesta (kuva 4). Järjestelmä muuttaa kuvan ensin harmaasävykuvaksi ja laskee joka alueesta *entropian* ja yhteismatriisin (co-occurrence matrix), josta saadaan arvot *kontrastille*, *energialle* ja *homogeenisuudelle*. Joka alueesta lasketaan siis yhteensä neljä tekstuuria kuvaavaa arvoa ja näin kustakin kuvasta saadaan  $3 \times 4 = 12$ -ulotteinen vektori. Tekstuuria edustavilla piirrevektoreilla koulutetaan tekstuuuri-GHSOM (kohta 5.2) ja niiden

avulla verrataan esimerkkikuvan tekstuuria koulutuskuvajoukon tekstuureihin haussa (kohta 5.5).

## 5.2 GHSOM:t ja niiden koulutus

Värille ja tekstuurille muodostetaan kummallekin oma GHSOM:nsa. GHSOM:t voidaan kouluttaa samalla tai eri kuvajoukoilla. Järjestelmä kouluttaa väri-GHSOM:n kuvista lasketuilla väripiirteillä ja tekstuuri-GHSOM:n tekstuuripiirteillä.

GHSOM on Pampalkin ja Chanin (2002) kehittämä tietorakenne, joka sisältää alikartat ja tiedot niiden välisistä suhteista. Kukin GHSOM:n alikartta on SOM Toolbox:n som\_map-rakenne (Alhoniemi & al., 2005), jossa on mm. neuronien painovektorit, tietoja SOM:n koulutuksesta ja topologiasta sekä ns. leimoina kuhunkin kartan neuronin kartoittuvat koulutusdatayksiköt. Tämä järjestelmä tallentaa leimoiksi koulutuskuvajoukon kuvien tiedostonimet polkuineen.

## 5.3 Kuvan kartoitus GHSOM:lle

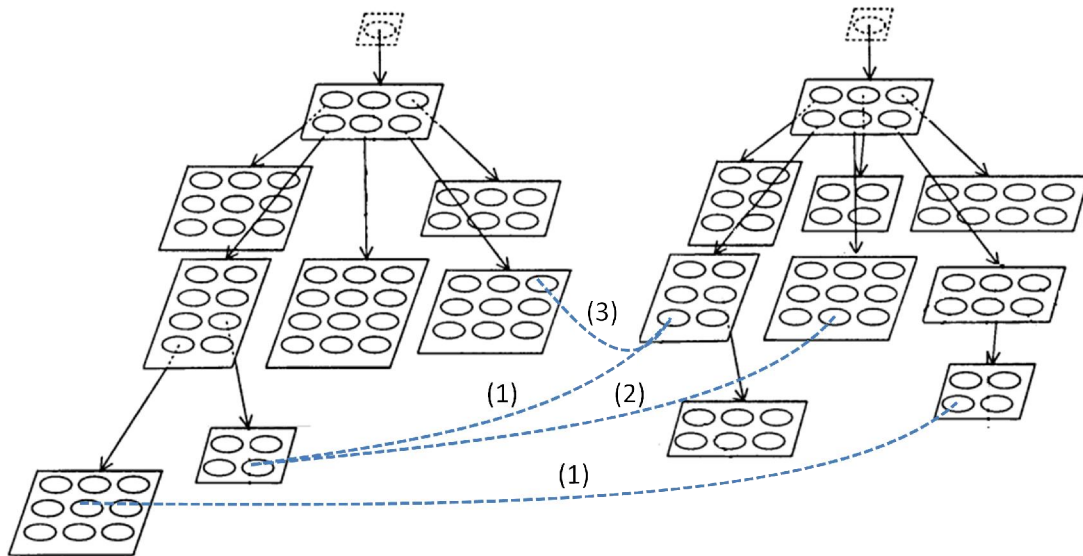
Kuvan kartoitus vain yhdestä kartasta muodostuvalle SOM:lle on yksiselitteistä: Kuvasta lasketaan piirrevektori ja SOM:lta etsitään se neuronin, jonka painovektori on lähimpänä tätä piirrevektoria valitun etäisyysmitan mukaan. GHSOM:lla kuva voi kuitenkin kartoittua usealle alikartalle eri tasoilla. Tässä järjestelmässä kuvalla halutaan löytää voittajaneuronin GHSOM:n alimmalta mahdolliselta alikartalta, ja tämä neuronin tulkitaan kuvan voittajaneuroniksi koko GHSOM:lla.

Järjestelmä vertaa kuvasta laskettua piirrevektoria ensin ylimmän tason alikartan neuronien painovektoreihin. Jokin niistä on lähimpänä kuvan piirrevektoria, ja tämä neuronin on voittajaneuronin ylimmän tason alikartalla. Jos ylimmän tason alikartan voittajaneuronilla on lapsikartta, tältä alikartalta etsitään piirrevektorille voittajaneuronin. Jos tälläkin neuronilla on lapsikartta, siltä etsitään edelleen voittajaneuronin. Näin jatketaan, kunnes löydetään voittajaneuronin, jolla ei enää ole lapsikarttoja. Tämä on kuvan voittajaneuronin koko GHSOM-rakenteessa.



## 5.4 Yhteyksien luominen GHSOM:ien välille

Sen jälkeen kun kumpikin GHSOM on koulutettu erikseen, niiden välille muodostetaan yhteydet. Yhteydet voidaan muodostaa kuvajoukolla, jota on käytetty toisen tai molempien GHSOM:ien koulutukseen, tai kokonaan eri kuvajoukolla. Kukin yhteyksien muodostuksessa käytetyn kuvajoukon kuva kartoittuu aina yhteen neuroniiin kummallakin GHSOM:lla (ks. kohta 5.3), ja yhteys muodostuu näiden kahden neuronin välille. Jos neuronien välillä on jo yhteys, sitä vahvistetaan. Koulutuksen jälkeen yksittäisellä neuronilla voi olla useampia kuin yksi yhteys, kukin eri neuroniiin toisella GHSOM:lla. On myös mahdollista, että neuronilla ei ole yhtään yhteyttä. Yhteyksillä on vahvuudet sen mukaan, kuinka moni kuva on kartoittunut yhteyden osoittamaan neuronipariin yhteyksiä muodostettaessa. Yhteyksien idea on esitetty kuvassa 5.



**Kuva 5: Yhteyksiä kahden GHSOM:n neuronien välillä. Suluissa olevat numerot kuvaavat yhteyksien vahvuuksia.**

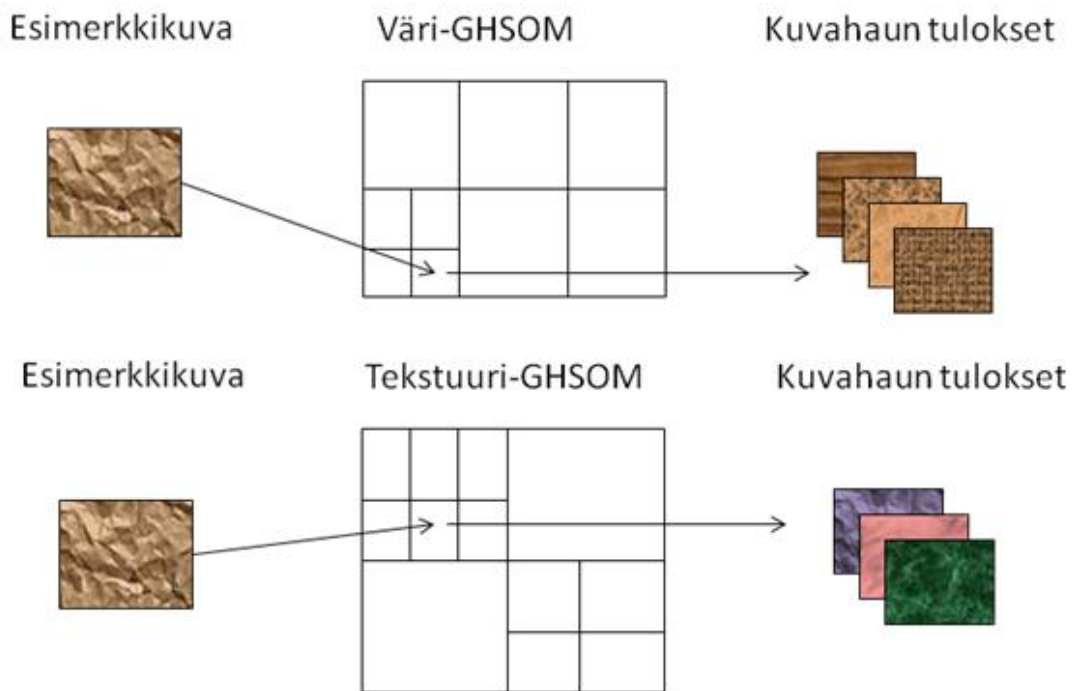
Yhteydet esitetään kahdessa eri tietorakenteessa: Toisessa on helppo navigoida väri-GHSOM:n neuroneista teksturi-GHSOM:n neuroneihin päin ja toisessa teksturi-GHSOM:n neuroneista väri-GHSOM:n neuroneihin päin. Yhteysrakenteessa GHSOM:n kullekin neuronille esitetään sen yhteydet. Kustakin yhteydestä kerrotaan, mihin neuroniiin toisella GHSOM:lla se kohdistuu, yhteyden vahvuus ja niiden kuvatiedostojen nimet, jotka ko. yhteyden ovat luoneet. Yhteydet esitetään tietorakenteessa järjestyksessä vahvimmasta heikoimpaan.

## **5.5 Hakutoimintojen kuvaukset**

Haku alkaa aina käyttäjän antamasta yhdestä esimerkkikuvasta, jolle haetaan tuloksia. Seuraavissa alikohdissa kuvataan erilaiset mahdolliset haut. Yksi mahdollisuus on hakea esimerkkikuvalle samanlaisia kuvia erikseen väripiirteiden tai tekstuuri-*piirteiden* mukaan joko vastaavan GHSOM:n koulutukseen tai yhteyksien muodostamiseen käytetystä kuvajoukosta (alikohtat 5.5.1 ja 5.5.2). Toinen mahdollisuus on hakea sekä väriltään että tekstuurltaan esimerkkikuvaa vastaavia kuvia yhteyksien muodostukseen käytetystä kuvajoukosta (alikohta 5.5.3). Kolmas ja monimutkaisin hakumahdollisuus on hakea esimerkkikuvan väripiirteisiin liittyviä tekstuureja edustavia kuvia tekstuuri-GHSOM:n koulutukseen käytetystä kuvajoukosta tai esimerkkikuvan tekstuuri-*piirteisiin* liittyviä värejä edustavia kuvia väri-GHSOM:n koulutukseen käytetystä kuvajoukosta (alikohta 5.5.4). Viimeksi mainitun hakutyypin tulokset kertovat yleisesti siitä, millaiset värit ja tekstuurit esiintyvät yhdessä yhteyksien muodostukseen käytetyssä kuvajoukossa.

### **5.5.1 Haku yhden piirrelajin mukaan yhdeltä GHSOM:lta**

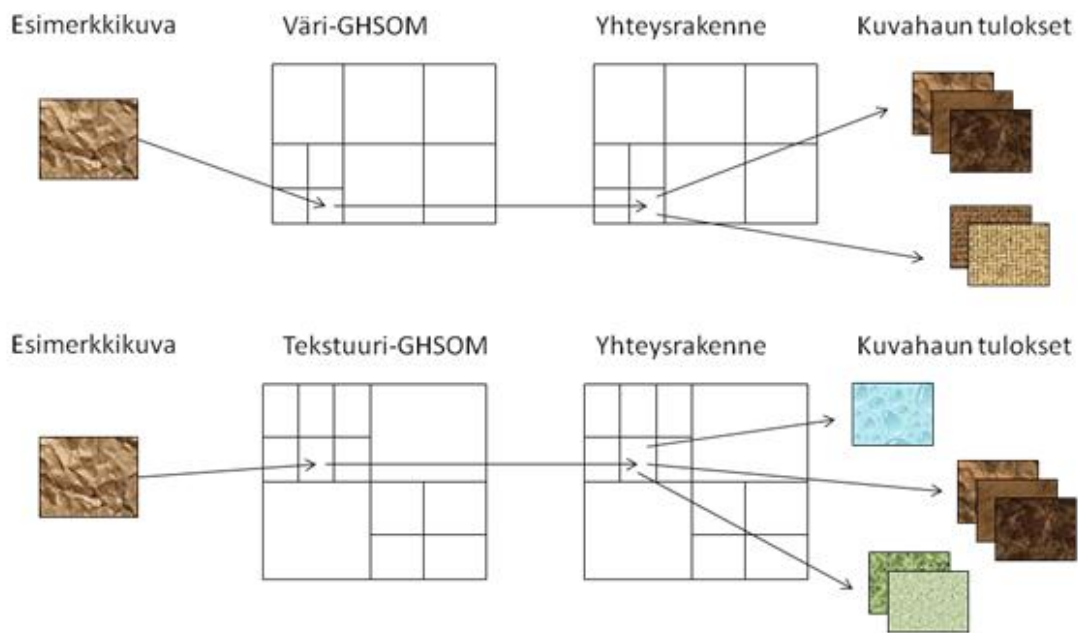
Tässä haussa haetaan esimerkkikuvalle samanvärisiä kuvia väri-GHSOM:n koulutuskuvajoukosta tai vastaavasti samantekstuurisia kuvia tekstuuri-GHSOM:n koulutuskuvajoukosta. Haettaessa samanvärisiä kuvia järjestelmä laskee esimerkkikuvasta väripiirteet samoin kuin koulutusdatan kuvista (kohta 5.1) ja kartoittaa sen niiden avulla väri-GHSOM:lle. Haun tuloksena annetaan väri-GHSOM:n voittajaneuroniin liittyvät koulutuskuvajoukon kuvat. Vastaavasti haettaessa esimerkkikuvalle tekstuurltaan samanlaisia kuvia järjestelmä laskee esimerkkikuvasta tekstuuri-*piirteet* samoin kuin koulutusdatan kuville (kohta 5.1) ja kartoittaa sen niiden avulla tekstuuri-GHSOM:lle. Haun tuloksena annetaan tekstuuri-GHSOM:n voittajaneuroniin kartoituneet koulutuskuvajoukon kuvat. Jos esimerkkikuvalle laskettuun voittajaneuroniin ei liity GHSOM:lla yhtään koulutuskuvajoukon kuvaa, tuloksia ei saada. Haut on kuvattu pääpiirteissään kuvassa 6.



**Kuva 6: Samanväristen tai -tekstuuristen kuvien haku esimerkkikuvalle vastaavan piirrelajin GHSOM:ilta.**

### 5.5.2 Haku yhden piirrelajin mukaan yhteysrakenteesta

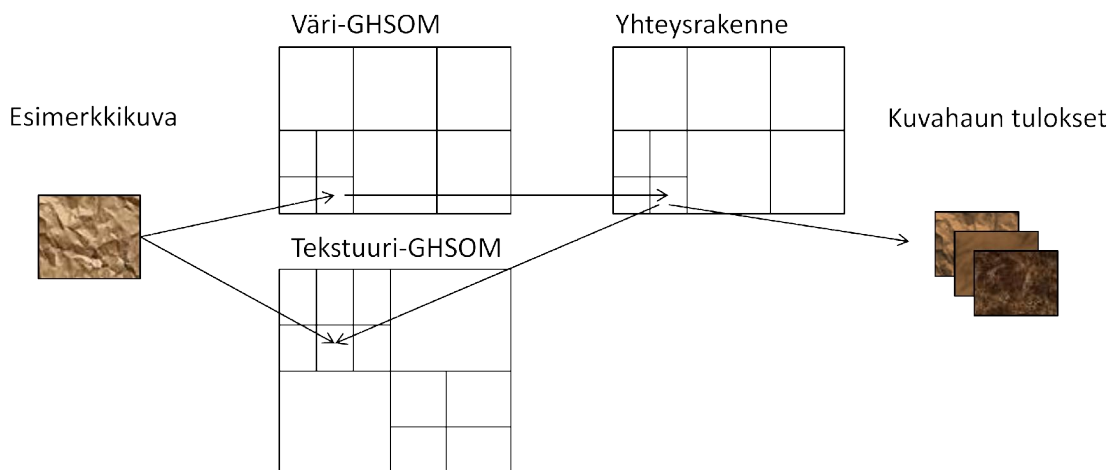
Pelkän värin tai tekstuurin perusteella voi hakea kuvia myös yhteyksien muodostukseen käytettyjen kuvien joukosta. Tällöin järjestelmä laskee esimerkkikuvasta esim. väripiirteet ja kartoittaa sen väri-GHSOM:lle. Jos löytyneellä voittajaneuronilla on yhteyksiä, järjestelmä antaa tuloksena yhteysrakenteeseen tallennetut, tämän yhteyden luoneet kuvat. Jos yhteydet on muodostettu samalla kuvajoukolla, kuin millä väri-GHSOM on koulutettu, tulokset ovat samat kuin haettaessa kuvia suoraan väri-GHSOM:lta (alikohta 5.5.1). Jos väri-GHSOM:n neuronilla, johon kuva kartoittuu, on useita yhteyksiä, tuloksina saaduista kuvista mainitaan, mistä yhteydestä ne ovat peräisin. Kaikkien kuvien pitäisi olla lähellä esimerkkikuvan väriä. Samasta yhteydestä löytyneillä kuvilla pitäisi myös olla keskenään samanlaiset tekstuurit, jotka kuitenkin saattavat poiketa muista yhteyksistä saatujen kuvien tekstuureista. Jos esimerkkikuvalle lasketulla voittajaneuronilla ei ole yhteyksiä, tuloksia ei saada. Haku tekstuurin perusteella tapahtuu vastaavasti kuin värin perusteella: järjestelmä kartoittaa esimerkkikuvan ensin tekstuuri-GHSOM:lle, ja jos saadulla voittajaneuronilla on yhteyksiä, antaa tuloksena niihin liittyvät kuvat yhteysrakenteesta. Tämän tyyppiset haut on kuvattu pääpiirteissään kuvassa 7.



**Kuva 7: Samanväristen tai -tekstuuristen kuvien haku esimerkkikuvalle yhteysrakenteesta.**

### 5.5.3 Haku sekä värin että tekstuurin mukaan

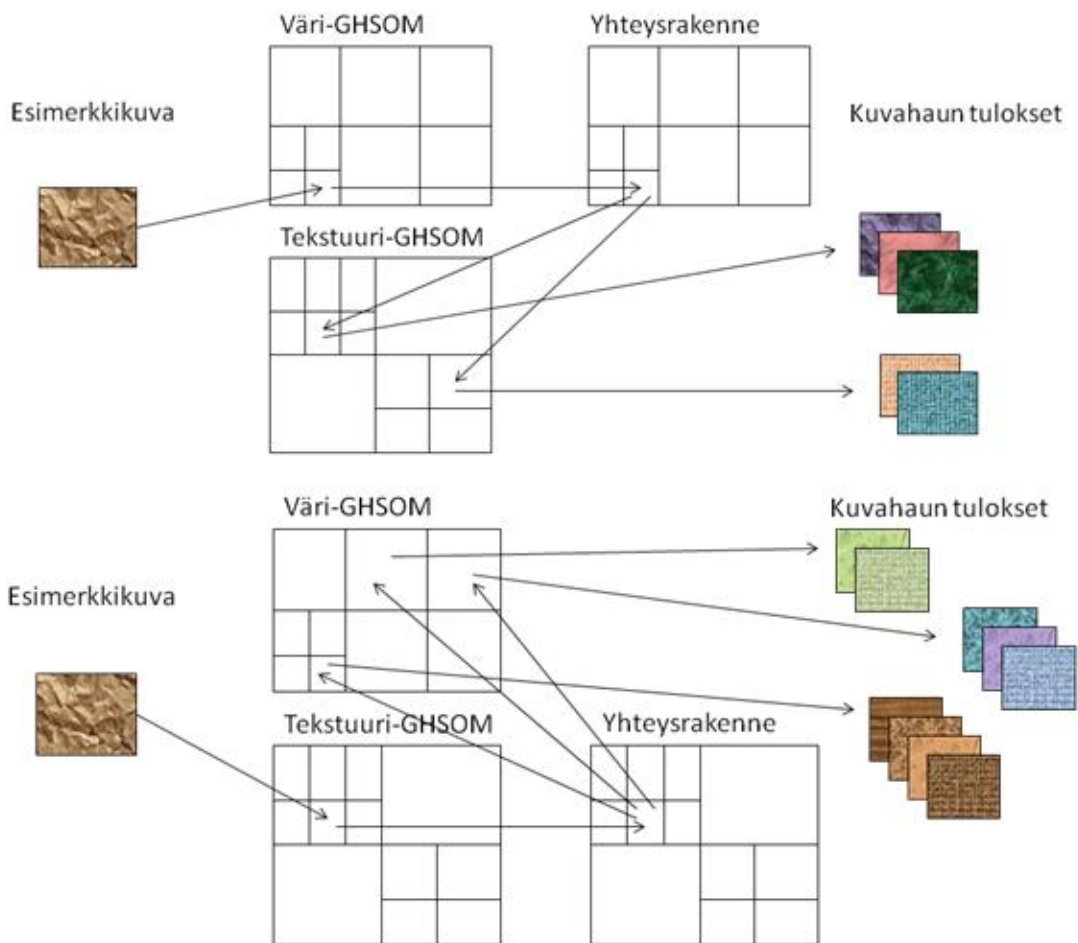
Jos halutaan löytää kuvia, jotka vastaavat esimerkkikuvaa sekä värin että tekstuurin puolesta, on haettava kuvia yhteyksien muodostukseen käytetystä kuvajoukosta. Järjestelmä kartoittaa esimerkkikuvan väri-GHSOM:lle väripiirteiden mukaan ja teksturi-GHSOM:lle tekstuuripiirteiden mukaan. Jos näiden voittajaneuronien välillä on yhteys, järjestelmä antaa tuloksena yhteysrakenteesta ko. yhteyden luoneet kuvat. Jos taas voittajaneuronien välillä ei ole yhteyttä, tuloksia ei saada, koska yhteyksien muodostukseen käytetyssä kuvajoukossa ei ole yhtään kuvaa, joka vastaisi esimerkkikuvaa kummankin piirrelajin mukaan. Haettaessa kahden piirrelajin mukaan saadaan teoriassa vähemmän tuloksia kuin yhden piirrelajin mukaan haettaessa, mutta tulokset ovat enemmän esimerkkikuvan kaltaisia. Haku kahden piirrelajin mukaan on kuvattu pääpiirteissään kuvassa 8.



**Kuva 8: Samankaltaisten kuvien hakeminen esimerkkikuvalla sekä värin että tekstuurin mukaan yhteysrakenteesta.**

#### **5.5.4 Kuvan väriin liittyviä tekstuureja tai tekstuuriin liittyviä värejä edustavien kuvien haku**

Haettaessa kuvia, jotka edustavat esimerkkikuvan värin kanssa esiintyviä tekstuureja, järjestelmä laskee esimerkkikuvasta väripiirteet, kartoittaa sen väri-GHSOM:lle, seuraa väri-GHSOM:lta löytyneen voittajaneuronin yhteyksiä tekstuuri-GHSOM:lle ja antaa hakutuloksena näihin neuroneihin liittyvät tekstuuri-GHSOM:n koulutuskuvajoukon kuvat. Haettaessa esimerkkikuvan tekstuuriin liittyviä värejä edustavia kuvia tehdään sama päinvastoin: Esimerkkikuvasta lasketaan tekstuuripiirteet, kartoitetaan se tekstuuri-GHSOM:lle, seurataan löytyneen voittajaneuronin yhteyksiä väri-GHSOM:lle ja annetaan hakutuloksena näihin neuroneihin liittyvät väri-GHSOM:n koulutuskuvajoukon kuvat. Jos ensimmäisen GHSOM:n voittajaneuronilla ei ole yhteyksiä tai toisen GHSOM:n neuroneihin ei liity yhtään kuvaa, tuloksia ei saada. Jos ensimmäisen GHSOM:n neuronilla on yhteyksiä useampaan kuin yhteen toisen GHSOM:n neuroniin, tuloskuvista mainitaan, mistä yhteydestä ne ovat peräisin. Tämänäntyyppiset haut on kuvattu pääpiirteissään kuvassa 9.



**Kuva 9: Esimerkkikuvan väriin liittyvien tekstuurien tai tekstuuriin liittyviä värejä edustavien kuvien haku.**

Haettaessa esimerkkikuvan väriin liittyviä tekstuurien edustavia kuvia, samasta yhteydestä saaduilla kuvilla pitäisi olla keskenään samanlainen tekstuuri, joka kuitenkin saattaa poiketa paljonkin muista yhteyksistä saatujen kuvien tekstuurista. Jos järjestelmä on oppinut yhteyksien muodostuksen aikana assosioimaan esimerkkikuvan värin ja tekstuurin toisiinsa, ainakin yhdestä yhteydestä saatujen kuvien pitäisi olla tekstuuriltaan samankaltaisia kuin esimerkkikuva. Haettaessa kuvan tekstuuriin liittyviä värejä edustavia kuvia samasta yhteydestä saatujen kuvien pitäisi olla keskenään samanvärisiä mutta mahdollisesti erivärisiä kuin muista yhteyksistä saadut kuvat. Jos järjestelmä on oppinut yhteyksien muodostuksen aikana assosioimaan esimerkkikuvan värin ja tekstuurin toisiinsa, ainakin yhdestä yhteydestä saatujen kuvien pitäisi olla jokseenkin samanvärisiä kuin esimerkkikuva.

## 6 Järjestelmän toiminnan testaus ja arviointi

Tässä luvussa testattiin ohjelman varsinaista toimintaa eli kuvien luokittelua GHSOM:ien avulla ja hakuja eri hakufunktiolla. Ensimmäisessä käyttötapauksessa (kohta 9.1) väri-GHSOM koulutettiin kuvilla, joissa kaikissa oli jokseenkin yhtenäisen väri, ja tekstuuri-GHSOM kuvilla, joiden oli tarkoitus esittää erilaisia tekstuureja. Toisessa käyttötapauksessa (kohta 9.2) molemmat GHSOM:t koulutettiin kuvilla, jotka esittivät erilaisia kohteita ja saattoivat sisältää erivärisiä alueita. Testien tulokset selviävät näissä kohdissa 9.1 ja 9.2.

### 6.1 Toiminta yhtenäisen värisillä ja eri tekstuureja edustavilla kuvilla

Väri-GHSOM koulutettiin kuvilla, joissa jokaisessa oli yksi tietty perusväri. Kuvissa hyväksyttiin pieniä valkoisia alueita, samoin värin tummuusasteen lievää vaihtelua. Kuvia pyrittiin valitsemaan melko tasainen määrä eri pääväreissä. Syntyneen GHSOM:n rakenne ja kuvien sijoittuminen solmuihin kävi ilmi `makeColorGhsom`-funktion automaattisesti tuottamasta tulosteesta ja piirrettiin kuvaksi manuaalisesti (kuva 10). Kuvasta nähdään, että hyvin tummat ja hyvin vaaleat kuvat sijoittuivat kartalla vastakkaisiin nurkkiin, samoin sini-violetit ja keltaiset. Väri muuttuu melko tasaisesti nurkasta nurkkaan. Punaiset kuvat sijaitsevat lähellä toisiaan, ja punaisten ja keltaisten kuvien välissä on oransseja kuvia. Vihreät kuvat jakautuivat hieman eri kohtiin GHSOM:ia: Tummanvihreät kuvat sijaitsevat lähellä muita tummia kuvia, hyvin vaaleanvihreät lähellä vaaleimpia kuvia, kellanvihreät lähellä keltaisia kuvia ja sinertävän- tai harmaanvihreät kuvat siniharmaiden kuvien vieressä. Värikomponenttia *a* vastaavalla vihreä-magenta-akselilla ei siis ole yhtä selvää jakoa kuin *L*-komponenttia vastaavalla tumma-vaalea- tai *b*-komponenttia vastaavalla sininen-keltainen-akselilla. Tämä on ymmärrettävää, koska SOM tekee kuvista kaksiulotteisen esityksen.





**Kuva 10: Yhtenäisen värisillä kuvilla koulutettu väri-GHSOM.**

Tekstuuri-GHSOM koulutettiin kuvilla, joiden oli tarkoitus edustaa erilaisia tekstuuria (kuva 11). Osassa kuvista oli suuria kontrasteja, osassa taas ei juuri lainkaan. Jotkut kuvista sisälsivät säännöllisiä kuvioita, jotkut epäsäännöllisiä. Jotkut kuvat taas koostuivat muutamasta yhtenäisestä alueesta, joiden välinen kontrasti oli suuri. Koska ohjelma muuttaa värikuvat harmaasävykuviksi ennen tekstuuripiirteiden laskeamista, värisävyn vaihteluja ei huomioida tekstuuripiirteissä, ainoastaan intensiteetin muutoksia.





**Kuva 11: Erilaisia tekstuureja edustavilla kuvilla koulutettu tekstuuri-GHSOM.**

Käytetyt tekstuureja kuvaavat suureet olivat entropia, kontrasti, energia ja homogeenisuus. Tällaisia ominaisuuksia on vaikeampi hahmottaa silmin kuin värejä. Kuvan 11 perusteella voisi kuitenkin sanoa, että kuvat, joissa oli suuria yhtenäisiä alueita ja kontrasteja niiden välillä, kartoituivat yhteen GHSOM:n nurkkaan. Suurikontrastiset mutta enemmän toistuvia kuvioita sisältävät kuvat sijoittuivat näiden viereen, samoin pienempikontrastiset suuria yhtenäisiä alueita sisältävät kuvat. Vastakkaiseen nurkkaan taas sijoittuivat kuvat, joissa oli jatkuvasti pieniä asteittain muuttuvia melko epäsäännöllisiä intensiteettieroja. Järjestelmä luokitteli yhden kuvista yksinään yhteen ylimmän tason kartan neuroniin heti koulutuksen alussa (kuva 11 oikea yläkul-

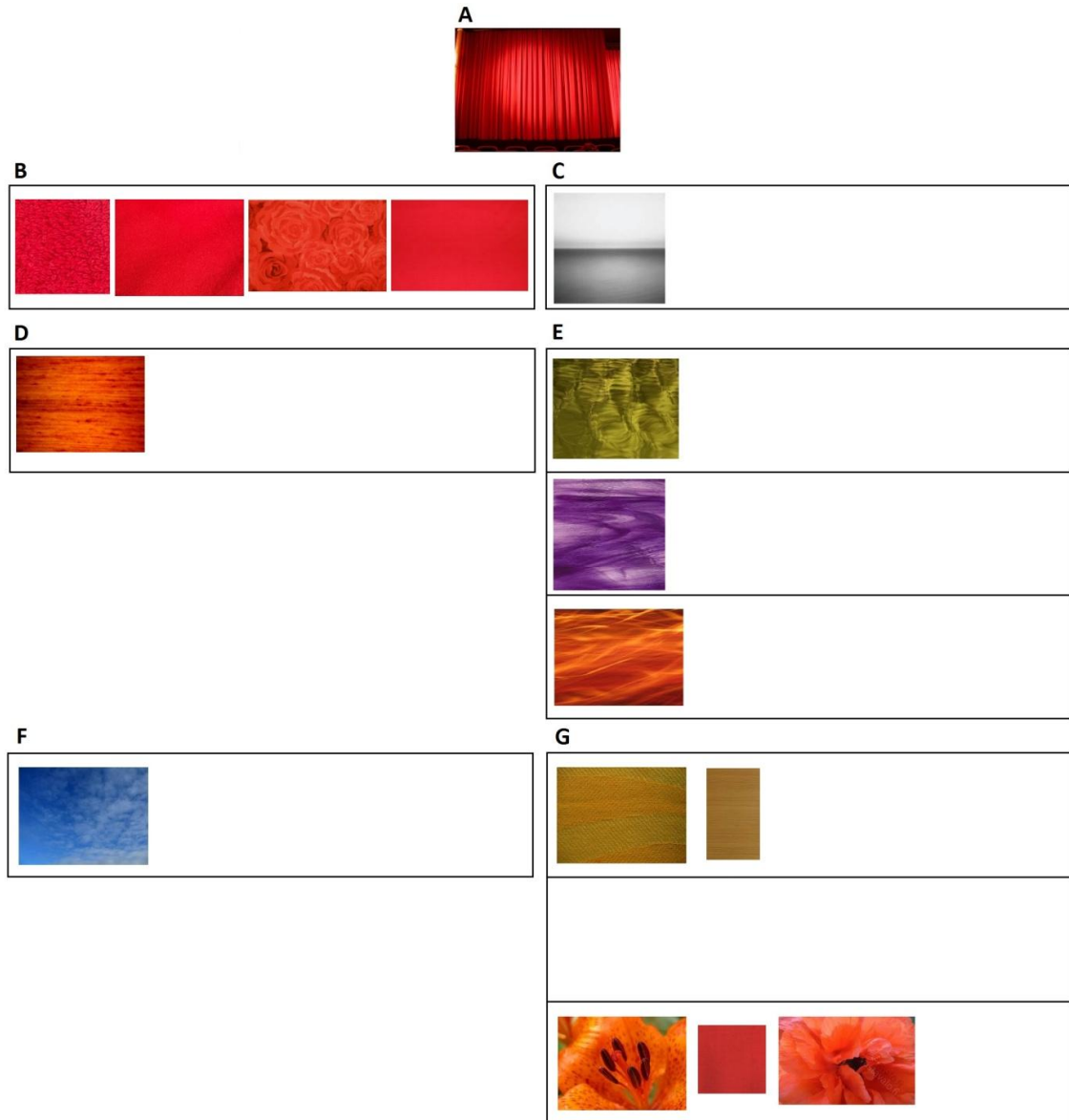
ma, heikosti erottuva harmaa neliö harmaan neuronin keskellä). Kyseisessä kuvassa oli säännöllisesti toistuvia ohuita tummemman ja vaaleamman harmaita viivoja, joiden intensiteetit olivat kohtalaisen lähellä toisiaan. Silmämääräisesti on vaikea sanoa, miten juuri tämä kuva poikkeaa kaikista muista kuvista. Se ei ole ainoa pienikontrastinen kuva, hyvin säännöllinen kyllä.

Edellä muodostettujen GHSOM:ien välille luotiin yhteydet kolmannella kuvajoukolla, jossa sekä värit että tekstuurit olivat kohtalaisen yhtenäisiä. Yhteyksien muodostamisen jälkeen testattiin kuvahakua esimerkkikuvalla, joka oli lähes kokonaan punainen ja jossa oli toistuvia pienikontrastisia pystyviivoja (kuva 12 A).

Haettaessa samanvärisiä kuvia väri-GHSOM:lta (funktio `searchColor`) saatiin tulokseksi neljä punaista kuvaa, joiden sävy oli melko lähellä esimerkkikuvaa (kuva 12 B). Haettaessa samanvärisiä kuvia värin yhteysrakenteesta (funktio `matchColor`) saatiin tulokseksi yksi oranssin- tai ruskehtavan punainen kuva (kuva 12 D). Tämäkin oli väriltään melko lähellä esimerkkikuvaa. `searchColor`- ja `matchColor` -funktioiden tulosten erot johtuvat siitä, että niillä haettiin tuloksia eri kuvajoukoista.

Haettaessa samantekstuurisiiä kuvia tekstuuri-GHSOM:lta (funktio `searchTexture`) saatiin tulokseksi yksi kuva, jossa on samanlainen asteittainen liukuma keskellä olevasta vaaleammasta sävystä tummempaan kuin esimerkkikuvassa ja toisaalta jako kahteen selvästi eri intensiteettitasoon (kuva 12 C). Pystyraidallisia tai verhon laskoksia muistuttavia kuvia ei kuitenkaan saatu hakutuloksena, vaikka niitäkin olisi koulutuskuvajoukossa ollut (ks. kuva 11).

Haettaessa samantekstuurisiiä kuvia tekstuurin yhteysrakenteesta (funktio `matchTexture`) saatiin tuloksia kolmesta eri yhteydestä, yksi kuva joka yhteydestä (kuva 12 E). Jokaisessa näistä näkyy jonkinlaisia matalan kontrastin epäsäännöllisiä raitoja, jotka muistuttavat hieman esimerkkikuvan verhon laskoksia. Kuten esimerkkikuvassa, intensiteetti muuttuu tuloskuvissa pehmeästi asteittain. Eri yhteyksistä tulleet kuvat ovat selvästi erivärisiä, kuten voi odottaa, koska jokainen yhteys on väri-GHSOM:n eri neuroniiin.



**Kuva 12: Hakutulokset eri hakufunktiolla yhdelle esimerkikuvulle (A). B) searchColor, C) searchTexture, D) matchColor, E) matchTexture, F) searchColor2Texture, G) searchTexture2Color. Hakufunktiolla matchBoth ei saatu tuloksia.**

Haettaessa sekä värin että tekstuurin puolesta samanlaisia kuvia yhteysrakenteesta (funktio matchBoth) ei saatu tuloksia. Tämä johtuu siitä, että yhteyksien muodostukseen käytetyssä kuvajoukossa ei ollut yhtään kuvaa, joka olisi kartoittunut sekä samaan väri-GHSOM:n neuronin että samaan tekstuuri-GHSOM:n neuronin kuin esimerkikuvaa.

Haettaessa kuvan väriin liittyviä tekstuureja edustavia kuvia tekstuuri-GHSOM:lta (funktio searchColor2Texture) saatiin tulokseksi yksi kuva (kuva 12 F). Esimerkkikuva siis kartoittui väri-GHSOM:lla neuronin, johon on tässä tapauksessa kartoittu-

nut yksi yhteyksien muodostukseen käytetyn kuvajoukon kuva eli kuva 12 D. Tämä kuva 12 D taas oli kartoittunut yhteen tekstuuri-GHSOM:n neuroniin ja luonut siihen yhteyden. Samaan tekstuuri-GHSOM:n neuroniin oli kartoittunut yksi tekstuuri-GHSOM:n koulutukseen käytetty kuva, siis kuva 12 F, joka saatiin funktion searchColor2Texture haun tuloksena. Kuvien 12 D ja 12 F tekstuurien pitäisi siis olla lähellä toisiaan, mikä näyttää myös pitävän paikkansa. Näiden kuvien tekstuuri on se tekstuuri, joka järjestelmän mukaan liittyy esimerkkikuvan väriin.

Toisaalta haettaessa kuvan väriin liittyviä tekstuureja edustavia kuvia tekstuuri-GHSOM:lta saatiin eri tulokset kuin hakemalla suoraan kuvan tekstuurin kanssa samanlaisia tekstuureja (kuvat 12 F ja C). Näiden tulokuvien tekstuurit kuitenkin muistuttavat hieman toisiaan, ja ne sijaitsevat tekstuuri-GHSOM:n vierekkäisissä neuroneissa (kuvan 11 alareunassa kolmannen tason alikartassa). Tämän perusteella järjestelmän voidaan ajatella oppineen yhdistämään esimerkkikuvassa esiintyvät värin ja tekstuurin toisiinsa, koska melko samanlainen yhdistelmä oli yhteyksien muodostukseen käytetyssä kuvassa, tässä tapauksessa kuvassa 12 D.

Jo edellä matchTexture-funktiolla tehdyssä haussa kävi ilmi, että sillä tekstuuri-GHSOM:n neuronilla, johon esimerkkikuva kartoittui, oli kolme yhteyttä. Haettaessa kuvan tekstuuriin liittyviä värejä edustavia kuvia väri-GHSOM:lta (funktio searchTexture2Color) saatiin ensimmäisestä yhteydestä kaksi vaaleanruskeaa kuvaa, toisesta yhteydestä ei yhtään kuvaa ja kolmannelta yhteydestä kolme kuvaa, joiden väri vaihteli punaisen ja oranssin välillä (kuva 12 G). Ensimmäisen yhteyden kuvien (ylimmät ruudut) väri on suunnilleen samanlainen kuvissa 12 E ja 12 G, samoin kolmannen yhteyden kuvien (alimmat ruudut). Toisen yhteyden väri-GHSOM:n neuronin ei ollut kartoittunut yhtään väri-GHSOM:n koulutukseen käytettyä kuvaa.

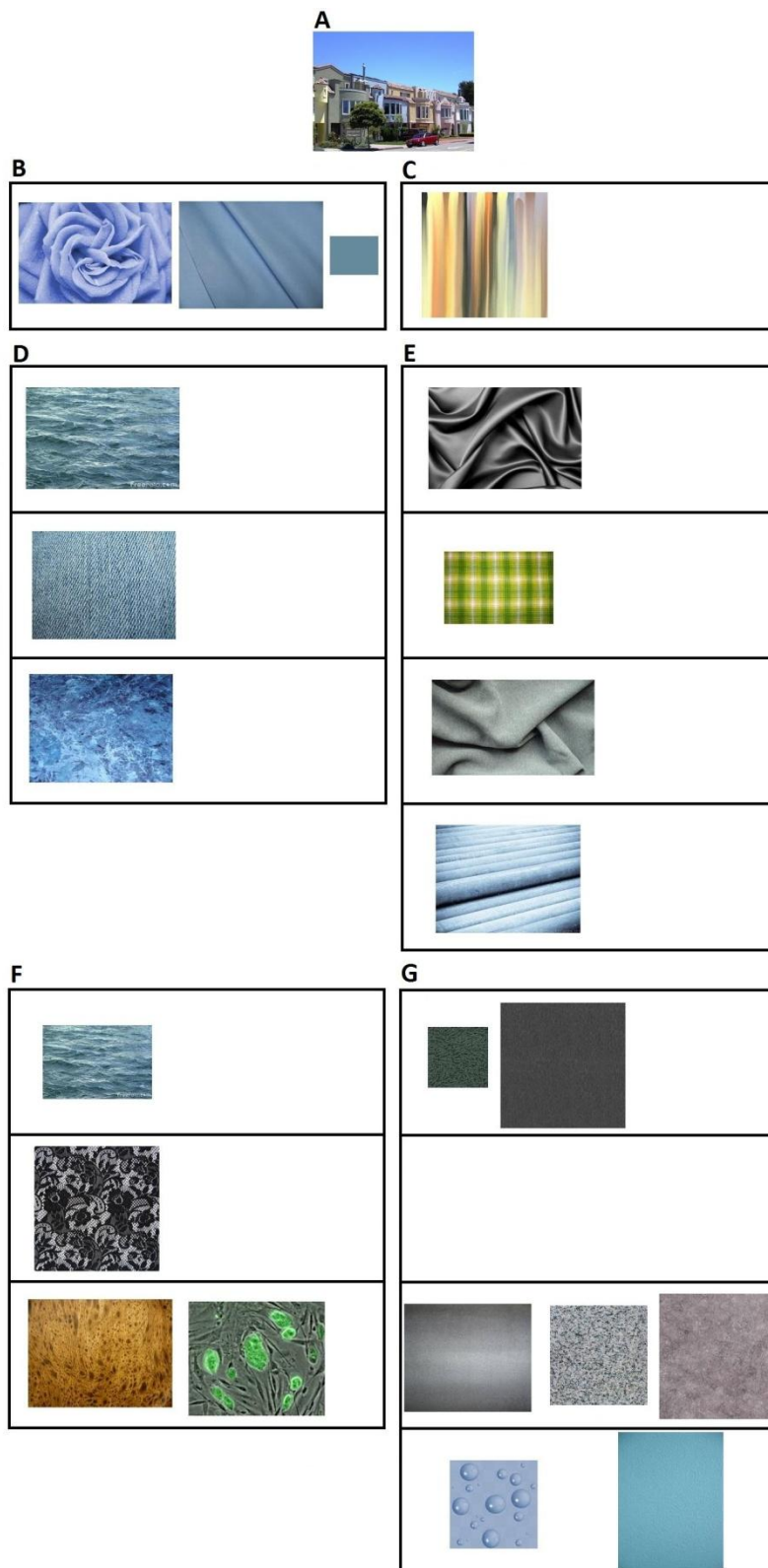
Myös haettaessa kuvan tekstuuriin liittyviä värejä edustavia kuvia väri-GHSOM:lta saatiin eri tulokset kuin haettaessa suoraan samanvärisiä kuvia esimerkkikuvalla (kuvat 12 G ja B). Tosin kolmannelta yhteydestä saatiin lähes samanvärisiä kuvia kuin suoraan väri-GHSOM:lta, ja nämä kuvat sijaitsevat väri-GHSOM:n vierekkäisissä neuroneissa (kuva 10 oikea reuna, kolmas alikartta ylhäältä). Tämän perusteella järjestelmä on oppinut yhdistämään esimerkkikuvassa esiintyvät värin ja tekstuurin toisiinsa, koska melko samanlainen yhdistelmä oli yhteyksien muodostukseen käytety-

sä kuvassa (kuva 12 E alin ruutu). Esimerkkikuvan tekstuuriin liittyy myös muita värejä, ja näitä edustavat muiden yhteyksien kautta saadut kuvat (kuvien 12 E ja G kaksi ylintä ruutua).

Haut tehtiin myös kuvalla, jossa oli useita erivärisiä taloja vierekkäin, melko paljon sinistä taivasta ja punainen auto etualalla (kuva 13 A). Tämä poikkeaa edellisestä esimerkkikuvasta ainakin siten, että siinä on useita erivärisiä alueita. Haettaessa samanvärisiä kuvia väri-GHSOM:lta (funktio `searchColor`) saatiin tulokseksi kolme harmahtavan sinistä kuvaa (kuva 13 B). Myös haettaessa samanvärisiä kuvia värin yhteysrakenteesta (funktio `matchColor`) saatiin tulokseksi kolme harmaansinistä kuvaa, kukin eri yhteydestä (kuva 13 D). Esimerkkikuvan hallitsevin sävy on myös sininen. Toisaalta koska väripiirteinä käytettiin keskimääräisiä värikomponenttien arvoja, kuva, joka sisältää useita eri värejä, voidaan tulkita keskimäärin harmaaksi.

Haettaessa samantekstuurisii kuvia tekstuuri-GHSOM:lta (funktio `searchTexture`) saatiin tulokseksi yksi kuva, jossa oli intensiteetiltään eritasoisia viivoja (kuva 13 C). Haettaessa samantekstuurisii kuvia tekstuurin yhteysrakenteesta (funktio `matchTexture`) saatiin tulokseksi yhteensä neljä kuvaa, kukin eri yhteydestä (kuva 13 E). Näissäkin on intensiteetiltään hieman vaihtelevia viivoja, toiset säännöllisempiä, toiset epäsäännöllisempiä. Haettaessa sekä värin että tekstuurin puolesta samanlaisii kuvia yhteysrakenteesta (funktio `matchBoth`) ei saatu tuloksia.

Haettaessa kuvan väriin liittyviä tekstuureja edustavii kuvia tekstuuri-GHSOM:lta (funktio `searchColor2Texture`) saatiin tulokseksi yhteensä neljä kuvaa kolmesta eri yhteydestä (kuva 13 F). Tässäkin on olennaista verrata kuvien 13 D ja F ylimpiä kuvia keskenään, keskimmäisii kuvia keskenään ja alimpia kuvia keskenään. Näiden pitäisi olla tekstuuriltaan samanlaisii. Tässä tapauksessa ylimmät kuvat näyttävät olevan samat, koska sama kuva on ollut sekä väri-GHSOM:n koulutukseen että yhteyksien muodostukseen käytetyissä kuvajoukoissa. Myös alimman yhteyden kuvat näyttävät melko samanlaisilta. Kuvien 13 D ja F tekstuurit ovat niitä, jotka järjestelmän mukaan liittyvät esimerkkikuvan värisiin (keskimäärin siniharmaisiin) kuviin. Tulokset olivat tässäkin erilaiset haettaessa tekstuuri-GHSOM:lta suoraan tai yhteyden kautta (kuvat 13 C ja F).



**Kuva 13: Hakutulokset eri hakufunktiolla toiselle esimerkkikuvalla (A). B) searchColor, C) searchTexture, D) matchColor, E) matchTexture, F) searchColor2Texture, G) searchTexture2Color. Hakufunktiolla matchBoth ei saatu tuloksia.**

Haettaessa kuvan väriin liittyviä tekstuureja (searchColor2Texture, kuva 13 F) ei tässäkään saatu täysin samoja tuloksia kuin suorassa tekstuurihaussa (searchTexture, kuva 13 C). Mikään yhteyksistä ei myöskään osunut aivan viereiseen neuroniin, mutta ensimmäinen ja kolmas yhteys osuivat samaan alikarttaan (ks. kuvan 11 vasemman alikartan keskirivin neuronit). Järjestelmä oli siis jossain määrin oppinut yhdistämään esimerkkikuvan värin ja tekstuurin toisiinsa.

Haettaessa kuvan teksturiin liittyviä värejä edustavia kuvia väri-GHSOM:lta (funktio searchTexture2Color) ensimmäisestä yhteydestä saatiin kaksi tummanharmaata kuvaa, toisesta ei yhtään kuvaa, kolmannesta kolme vaaleamman harmaata kuvaa ja neljännestä kaksi vaaleansinistä kuvaa (kuva 13 G). Tulokset vastaavat väriltään kuvan 13 E eri yhteyksien kuvia. Esimerkkikuvan teksturiin liittyi järjestelmän mukaan siis joko tummanharmaita, vaaleanvihreitä, vaaleanharmaita tai sinisiä kuvia. Jos verrataan väri-GHSOM:lta suoraan ja yhteyden kautta löytyneitä kuvia, nämä ovat eri kuvia mutta muistuttavat värinsä puolesta jonkin verran toisiaan. Neljännen yhteyden kautta löytyneet kuvat (kuva 13 G alin ruutu) sijoittuivat viereiseen neuroniin ja kolmannen yhteyden kautta löytyneet kuvat (kuva 13 G toiseksi alin ruutu) muuten samaan alikarttaan (kuva 10 alikartta vasemmalla toinen alhaalta) kuin suorassa värihaussa löytyneet kuvat (kuva 13 B). Järjestelmä oli tämänkin perusteella oppinut jossain määrin yhdistämään esimerkkikuvan tekstuurin ja värin toisiinsa.

## **6.2 Toiminta erivärisiä alueita sisältävillä esittäville kuvilla**

Toisessa tapauksessa väri-GHSOM koulutettiin kuvilla, jotka selvästi esittivät erilaisia asioita eivätkä yleensä olleet yhtenäisen värisiä (kuva 14). Kuvien järjestyminen kartalle värin mukaan ei ole yhtä selvä kuin yhtenäisten kuvien tapauksessa. Saman alikartan kuvat ovat kuitenkin enimmäkseen melko samanvärisiä. Vasemmassa alakulmassa on selvästi vaaleita kuvia ja vastakkaisessa kulmassa tummia kuvia. Oikean alakulman kuvat ovat pääasiassa punaisia, keltaisia ja oransseja, kun taas vasemman yläkulman kuvien värit ovat enimmäkseen tumman harmaita tai sinertäviä.









**Kuva 15: Tekstuuri-GHSOM, joka on koulutettu erivärisiä alueita sisältävillä esittäville kuvilla.**

Yhteydet muodostettiin kolmannella kuvajoukolla, joka myös sisälsi esittäviä, ei välttämättä väriltään tai tekstuurltaan yhtenäisiä kuvia. Aluksi tehtiin kaikki haut esimerkikuvalla, jossa oli merenranta, metsää, harmaata muurirakennelmaa ja pieni kaistale taivasta (kuva 16 A).

Haettaessa samanvärisiä kuvia väri-GHSOM:lta (funktio `searchColor`) saatiin tulokseksi yksi kuva (kuva 16 B). Sekin esitti vettä, metsää ja hieman taivasta. Haettaessa samanvärisiä kuvia värin yhteysrakenteesta (funktio `matchColor`) saatiin tulokseksi kolme kuvaa, kaikki eri yhteyksistä (kuva 16 D). Kaikki nämä kuvat esittivät maise-

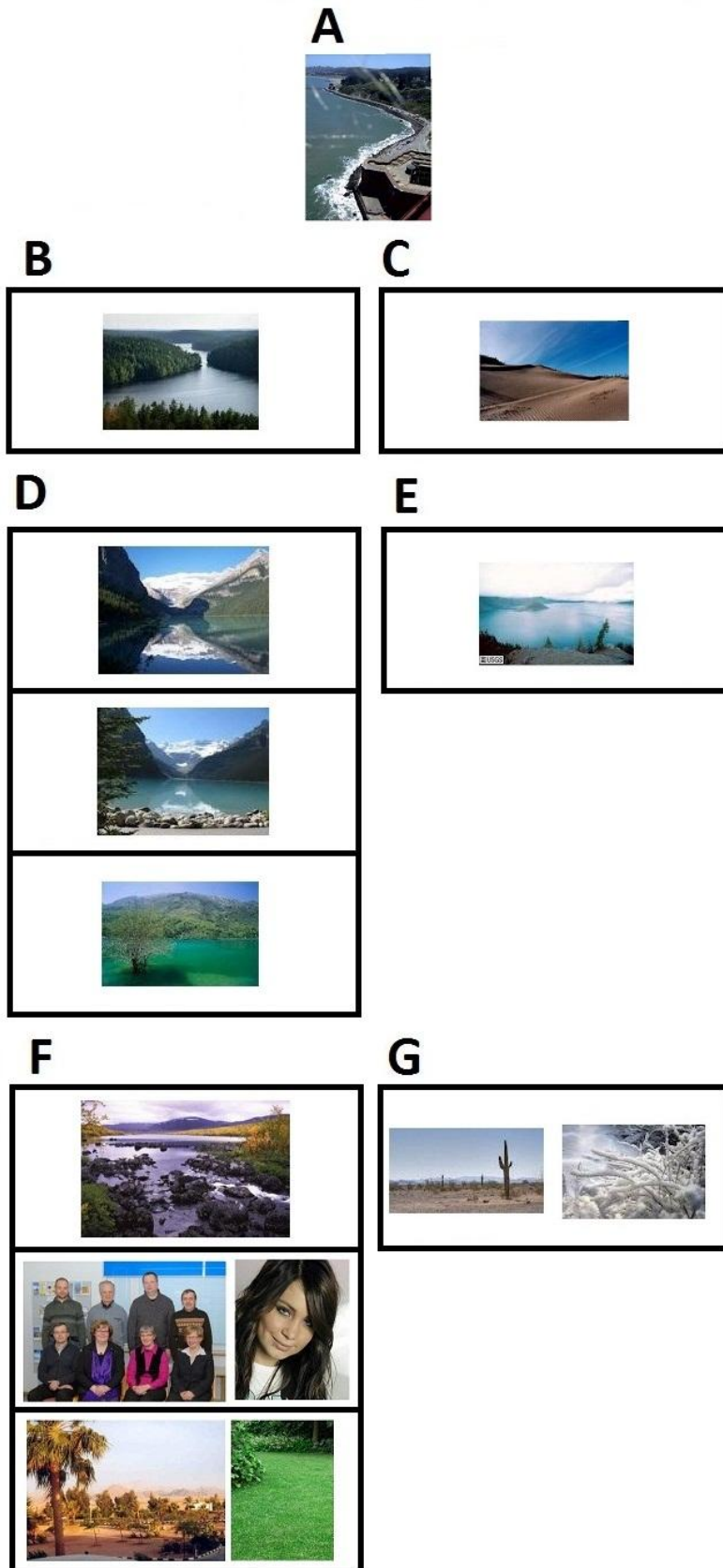
maa, jossa oli vettä, metsää ja hieman taivasta. Yhdessä kuvista oli myös harmaata maata. Tässä tapauksessa siis tuloksena saatujen kuvan värit ja sattumalta myös merkityssisältö vastasivat hyvin esimerkkikuvaa.

Haettaessa samantekstuurisia kuvia teksturi-GHSOM:lta (funktio `searchTexture`) saatiin tulokseksi yksi kuva (kuva 16 C). Sekin oli maisema, jossa tosin ei ollut vettä eikä metsää vaan hiekkaa. Jos värejä ei huomioida, kuva muistuttaa jonkin verran esimerkkikuvaa. Haettaessa samantekstuurisia kuvia tekstuurin yhteysrakenteesta (funktio `matchTexture`) saatiin tulokseksi yksi maisemakuva (kuva 16 E), jossa myös oli vettä, metsää, taivasta ja maata.

Haettaessa esimerkkikuvan väriin liittyviä tekstureja, saatiin tulokseksi yhteensä viisi kuvaa kolmesta eri yhteydestä. Ensimmäisestä yhteydestä saatu kuva oli hieman esimerkkikuvaa muistuttava maisemakuva. Toisesta yhteydestä saadut kaksi kuvaa esittivät ihmisiä ja kolmannesta yhteydestä saadut kaksi kuvaa lähinnä kasveista muodostuvia maisemia. Ensimmäisen ja kolmannen yhteyden kuvien voi ajatella olevan samanlaisia kuvissa 16 D ja F, mutta toisen yhteyden kuvat poikkeavat ainakin merkityssisällöltään selvästi toisistaan. Näilläkin saattaa silti olla jokseenkin samanlainen teksturi. Esimerkkikuvan väriin liittyi selvästi maisemakuvan teksturi mutta myös muunlaisia asioita esittäviä tekstureja.

Jos verrataan väriin liittyvien tekstuurien haussa löytyneitä kuvia suorassa teksturihaussa löytyneisiin kuviin (kuvat 16 F ja C), nämä eivät juurikaan muistuta toisiaan ja ne sijoittuivat teksturi-GHSOM:lla melko kauas toisistaan (kuva 15). Järjestelmä ei ollut oppinut yhdistämään esimerkkikuvan väriä ja tekstuuria toisiinsa, koska samanlaista yhdistelmää ei ollut ollut yhteyksien muodostukseen käytetyssä kuvajoukossa.

Haettaessa esimerkkikuvan teksturiin liittyviä värejä saatiin tulokseksi kaksi kuvaa yhdestä yhteydestä. Molemmat ovat vaaleita ja hieman sinertäviä ja melko samanvärisiä kuin kuva 16 E. Esimerkkikuvan teksturiin siis liittyi siis vaalea väri yhteyksien muodostukseen käytetyssä kuvajoukossa. Näin löytyneet kuvat eivät juurikaan muistuta suoraan värihaussa löytyneitä kuvia (kuva 16 B) ja ne sijoittuivat väri-GHSOM:lla melko kauas näistä (kuva 14).

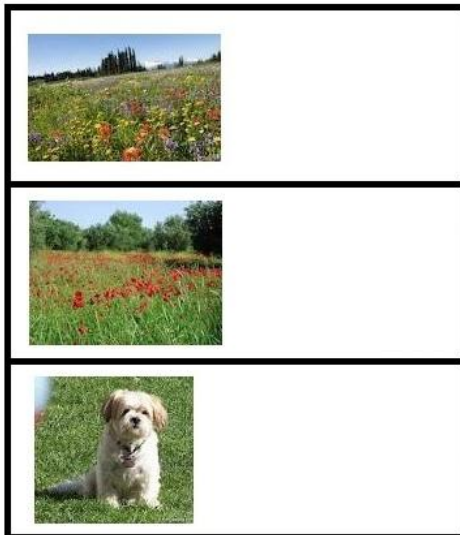
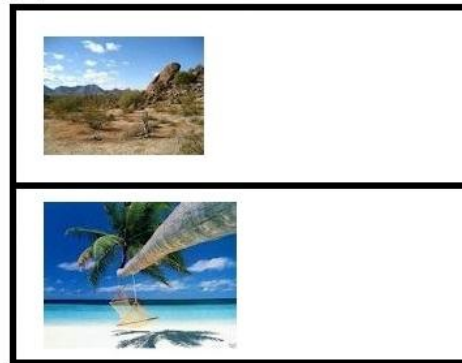
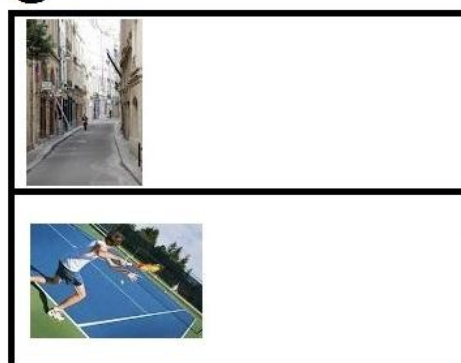


**Kuva 16: Hakutulokset eri hakufunktiolla yhdelle esimerkikuvulle (A). B) searchColor, C) searchTexture, D) matchColor, E) matchTexture, F) searchColor2Texture, G) searchTexture2Color. Hakufunktiolla matchBoth ei saatu tuloksia.**

Haut tehtiin myös toisella esittäväällä kuvalla, jossa oli vihreitä lehtipuita ja harmaata maata (kuva 17 A). Haettaessa samanvärisiä kuvia väri-GHSOM:lta (funktio searchColor) saatiin tulokseksi kaksi kuvaa (kuva 17 B). Niissä on taustalla vihreää heinää tai nurmikko, joka muistuttaa esimerkkikuvan puiden lehtiä. Lisäksi tuloskuvissa on harmaa eläin. Haettaessa samanvärisiä kuvia värin yhteysrakenteesta (funktio matchColor) saatiin tulokseksi kolme kuvaa, kukin eri yhteydestä (kuva 17 D). Niissäkin on paljon vihreää heinää tai nurmikko. Kahdessa kuvista on erivärisiä kukkia ja yhdessä harmaa eläin.

Haettaessa samantekstuurisia kuvia teksturi-GHSOM:lta (funktio searchTexture) saatiin tulokseksi kaksi kuvaa (kuva 17 C). Näistä toisessa on samoin taustalla vihreää nurmikko ja harmaanruskea eläin. Toisessa ei ole nurmikko vaan luminen vuori. Haettaessa samantekstuurisia kuvia tekstuurin yhteysrakenteesta saatiin tulokseksi kaksi kuvaa eri yhteyksistä (kuva 17 E). Molemmat ovat maisemakuvia ja niissä on hieman kasvillisuutta.

Haettaessa kuvan väriin liittyviä tekstuureja saatiin tulokseksi yhteensä viisi kuvaa kolmesta eri yhteydestä (kuva 17 F). Kaikissa on lähinnä puita tai nurmikko, mikä sopii hyvin siihen, että esimerkkikuva on pääasiassa vihreä. Haettaessa kuvan teksturiin liittyviä värejä saatiin tulokseksi kaksi kuvaa eri yhteyksistä (kuva 17 G). Nämä kuvat esittävät muuta kuin kasvillisuutta, tosin toisessa kuvassa on hieman nurmea ja puita. Kuvan teksturiin yleisimmin liittyvä väri ei siis ollut vihreä vaan joko harmaa tai sininen.

**A****B****C****D****E****F****G**

Kuva 17. Hakutulokset eri hakufunktioilla toiselle esimerkikuvulle (A). B) searchColor, C) searchTexture, D) matchColor, E) matchTexture, F) searchColor2Texture, G) searchTexture2Color. Hakufunktiolla matchBoth ei saatu tuloksia.

## 7 Yhteenveto

Työssä toteutettiin järjestelmä, joka luokittelee kuvia toisaalta väri- ja toisaalta tekstuuripiirteiden avulla sekä luo assosiaatioita yhdessä esiintyvien värien ja tekstuurien välille. Värien esittäminen värikomponenttien keskimääräisinä arvoina ennalta määrättyissä kuva-alueissa toimi hyvin käytettäessä yhtenäisen värisiä kuvia. Tällaisten keskiarvoihin perustuvien väripiirteiden ei voi edes odottaa toimivan yhtä hyvin kuville, joissa on useita erivärisiä alueita. Siitä huolimatta luokittelu värien mukaan sekä haut värien mukaan toimivat kohtalaisen hyvin myös objekteja esittäville kuville. Useita erivärisiä alueita sisältävien kuvien esittämiseen sopisi mahdollisesti vielä paremmin esim. värihistogrammi, jota on myös käytetty paljon sisältöperusteisessa kuvahaussa.

Käytetyt tekstuuripiirteet eivät vastaa yhtä selvästi ihmisen havaintoa kuin esimerkiksi väripiirteet. On esimerkiksi vaikea sanoa silmämääräisesti, missä kuvissa energia, homogeenisuus tai entropia on suuri ja missä pieni. Tekstuuri-GHSOM:ssa selvimmin havaittava trendi oli suuria yhtenäisiä alueita sisältävien suurikontrastisten kuvien sijoittuminen yhteen GHSOM:n kulmaan. Siinä missä väri-GHSOM:n voi sanoa toimivan parhaiten yhtenäisen värisille kuville, tekstuuuri-GHSOM:lle on vaikeampi nimetä tiettyä erityisen hyvin toimivaa kuvatyyppeä. Tekstuuriin perustuvissa kuten myös väriin perustuvissa hauissa hakutulokset vastasivat sisällöltään erimerkkikuvaa joskus mutta eivät aina. Hakujen onnistuminen riippuu aina tietenkin myös siitä, onko koulutuskuvajoukossa ylipäänsä esimerkkikuvaa vastaavia kuvia.

Järjestelmän oli tarkoitus oppia yhteyksien muodostukseen käytetyn kuvajoukon perusteella assosioimaan yhdessä esiintyviä värejä ja tekstuureja. Luvussa 3 kuvatuissa järjestelmissä (Yang & al., 2009; Ahmad & al., 2003; Johnsson & Balkenius, 2008) assosiaatioiden onnistumista arvioitiin testaamalla, kartoittuvatko kuvat samaan kohtaan SOM:ia suoraan SOM:n oman piirrelajin syötteellä kuin assosiaatioiden kautta toisen piirrelajin syötteellä. Näissä kolmessa kirjallisuudesta löytyneessä järjestelmässä assosiaatiot aktivoivat aina vahvimmin vain yhden tietyn kohdan toisen piirrelajin SOM:illa, jota voitiin verrata SOM:n oman piirrelajin syötteen aktivoimaan kohtaan. Nämä kohdat eivät esitetyissä teorioissa välttämättä ole samat edes

käytettäessä koulutuskuvajoukon kuvaa. Näissä julkaisuissa assosiaatioiden onnistumista testattiinkin pääasiassa juuri koulutuskuvajoukon kuvilla, joilla tulosten voi olettaa olevan parempia kuin koulutuskuvajoukon ulkopuolisilla kuvilla.

Tässä työssä toteutettu järjestelmä kuitenkin poikkeaa hieman aiemmista: Siinä yksi kuva voi aktivoida assosiaatioiden kautta yhtä vahvasti useita kohtia GHSOM:illa yhden sijasta. Lisäksi kuva, jota on käytetty yhteyksien muodostamisessa, osuu jo teorian mukaan aina täsmälleen samaan kohtaan GHSOM:ia yhden assosiaatioista kautta kuin suoraan ko. GHSOM:n omalla piirrelajilla. Tämän vuoksi assosiaatioiden onnistumista ei voi arvioida yhteyksien muodostuksessa käytettyjen kuvien avulla kuten aiemmissa järjestelmissä (tällä tavoin testattiin ainoastaan, oliko ohjelmointi toteutettu teorian mukaisesti oikein). On mielekkäämpää testata assosiaatioiden toimintaa kuvilla, jotka eivät kuuluneet yhteyksien muodostukseen käytettyyn kuvajoukkoon.

Jos kuva on yhteyksien muodostukseen käytetyn kuvajoukon ulkopuolelta, siinä voi olla värin ja tekstuurin yhdistelmä, jota yhteyksien muodostukseen käytetyssä kuvajoukossa ei ollut ja jota järjestelmä siksi ei ole voinut oppia. Jos taas yhteyksien muodostukseen käytetyssä kuvajoukossa on ollut myös esimerkkikuvan mukaisia värin ja tekstuurin yhdistelmiä, niiden neuronien välillä, joihin esimerkkikuva kartoittuu, pitäisi olla yhteys tai ainakin toisesta neuronista pitäisi olla yhteys toisen lähiympäristöön. Tuloksissa tämä näyttäisi siltä, että hakufunktiolla `searchTexture2Color` saataisiin yhden yhteyden kautta joko täysin samat tulokset kuin funktiolla `searchColor` tai muuten lähes esimerkkikuvan värisiä kuvia. Yhdessäkään testitapauksessa `searchColor` ei antanut täysin samoja tuloksia kuin mikään `searchTexture2Color` yhteyksistä, joten täsmälleen juuri niiden väri- ja teksturi-GHSOM:n neuronien välillä, joihin esimerkkikuva kartoittui, ei ollut yhteyttä. Kuitenkin niissä tapauksissa, joissa käytettiin yhtenäisen värisiä kuvia, esimerkkikuvan voittajaneuronista teksturi-GHSOM:lla oli yhteys lähelle esimerkkikuvan voittajaneuronista väri-GHSOM:lla eli funktio `searchTexture2Color` antoi yhdestä yhteydestä lähes esimerkkikuvan värisiä tuloksia. Järjestelmä oli siis oppinut suunnilleen esimerkkikuvassa olevan väri-teksturiyhdistelmän. Tällaisesta tilanteesta on se hyöty, että haut suoraan ja yhteyden kautta eivät tarjoa keskenään samoja vaan toisiaan täydentäviä tuloksia. Siis jos esimerkkikuvalla halutaan löytää samanvärisiä kuvia, parhaita



tuloksia saa todennäköisesti suoralla haulla värin mukaan (`searchColor`) mutta myös assosiaatioiden kautta (`searchTexture2Color`) saattaa löytää lisää halutunlaisia tuloksia.

Merkki yhteyksien onnistumisesta saadaan myös vertaamalla esim. hakujen `matchTexture` ja `searchTexture2Color` tuloksia yhteyksittäin. Kummankin funktion ensimmäisestä yhteydestä saatujen kuvien pitäisi olla keskenään suunnilleen samanvärisiä, samoin kummankin toisesta funktiosta saatujen kuvien keskenään jne., koska jokainen tekstuurin yhteysrakenteesta löytyvä kuva on luonut yhteyden väri-GHSOM:n johonkin neuroniiin, jossa pitäisi olla sen kanssa samanvärisiä kuvia. Testien mukaan näin myös oli. Tämä kertoo assosiaatioiden onnistumisesta yleensä riippumatta siitä, onko järjestelmä oppinut juuri esimerkikuvassa olevan yhdistelmän.

Testihauissa saatiin yleisesti melko pieniä määriä tuloksia. Erityisen ongelmallinen oli vaatimuksiltaan tiukin haku eli haku samanaikaisesti sekä värin että tekstuurin mukaan, jolla ei saatu lainkaan tuloksia yhdessäkään testihauissa. Tulosten pieni määrä johtuu siitä, että hakufunktiot antavat aina tuloksia yhdestä neuronista (tai yhdestä neuronista kutakin yhteyttä kohti) ja kuhunkin neuroniiin kartoittui testitapauksissa suhteellisen pieni määrä kuvia. Koulutuskuvajoukkojen kasvattaminen saattaisi lisätä tuloksia, mutta vain siinä tapauksessa, että GHSOM:n neuroneissa sallittaisiin enemmän sisäistä vaihtelua. Tätä säätelee lähinnä parametri  $\tau_2$ . Tällä hetkellä sen arvo on suhteellisen pieni, jolloin neuroneille tehdään lapsikarttoja suhteellisen helposti. Tulosten määrää voitaisiin siis haluttaessa lisätä säätämällä GHSOM:n parametria  $\tau_2$  suuremmaksi.

Toinen parannusehdotus tilanteisiin, jossa saadaan liian vähän tai ei lainkaan tuloksia, olisi, että järjestelmä hakisi aina tietyn määrän kuvia paremmuusjärjestyksessä (k `nearest neighbors` -haku). Tavalliselle SOM:lle on jo olemassa tällaisia algoritmeja (Oh & al., 2002; Im & al., 2005.), joissa haku etenee voittajaneuronista sen ympäristöön, kunnes tarvittava määrä tuloksia on saatu. GHSOM:n tapauksessa ei ole kuitenkaan yksiselitteistä, mistä lähimmät naapurit löytyvät. On mahdollista suosia voittajaneuronin oman alikartan neuroneja tai viereisen sisarusalikartan lähintä reunaa tai vanhempikartan naapurineuroneja. Haettaessa samanlaisia kuvia sekä värin että tekstuurin puolesta tai haettaessa kuvan väriin liittyviä tekstuureja tai tekstuureihin liitty-



viä värejä kyse ei ole ainoastaan yhden GHSOM:n läpikäyntijärjestyksestä vaan kahden GHSOM:n ja yhteyksien käyttö monimutkaistaa tilannetta entisestään. Olisi ainakin määriteltävä, kumman GHSOM:n voittajaneuronin ympäristöön siirrytään ensin ja miten yhteyksien painot otetaan huomioon.

Järjestelmää olisi mahdollista ottaa mukaan myös muuntyyppisiä piirteitä, kuten esimerkiksi muoto tai avainsana-annotaatiot, joko korvaamalla toisen käytössä olevista GHSOM:eista tai laajentamalla järjestelmää lisäämällä siihen uusia GHSOM:eja. Muoto olisi mahdollisesti tekstuuria parempi piirrelaji samoja objekteja esittävien kuvien löytämiseen, mutta muotopiirteiden laskeminen on haastavampaa varsinkin, jos siinä pyritään automaattiseen segmentointiin tai jopa objektien löytämiseen. Yhteyksistä olisi erityisen paljon hyötyä käytettäessä avainsana-annotaatioita, koska näin pystyttäisiin yhdistämään visuaalisia ja merkityssisältöjä toisiinsa. Annotaatioiden käyttö voisi mahdollistaa uusien kuvien automaattisen annotoimisen Yang & al. (2009) tapaan tai haut avainsanoilla.

Jos järjestelmään otetaan mukaan useampia piirrelajeja, jokaiselle uudelle piirrelajille on tehtävä funktiot piirreirrotukseen, GHSOM:n muodostukseen ja uusiin mahdollisiin hakuihin, ja yhteyksien muodostusfunktioita on muokattava. Yhteysrakenteet olisi mahdollista toteuttaa erikseen kaikille mahdollisille piirrelajipareille molempiin suuntiin. Tällöin jos järjestelmässä olisi  $n$  GHSOM:ia, yhteysrakenteita tarvittaisiin  $n(n-1)$ . Tämä ei vielä olisi liikaa, jos käytettyjä piirrelajeja oletetaan olevan vain muutama. Toinen mahdollisuus olisi tehdä jokaiselle GHSOM:lle yksi yhteysrakente, jonka jokaisesta yhteydestä kävisi myös ilmi, mihin GHSOM:iin se suuntautuu. Useamman GHSOM:n käyttö mahdollistaa yhteyksien kautta monia uudenlaisia hakuja. Ensinnäkin nykyiset haut olisivat mahdollisia uusille piirrelajeille ja kahden piirrelajin yhdistelmille. Lisäksi voisi vielä hakea esim. kuvan väriin liittyviin tekstuureihin liittyviä annotaatioita jne. Haku yhtä aikaa useamman kuin kahden piirrelajin mukaan ei kuitenkaan olisi mahdollinen niin kauan kuin jokaisessa yhteydessä on vain kaksi osapuolta.

Järjestelmää voidaan siis käyttää samanväristen tai tekstuuriltaan samanlaisten kuvien hakemiseen jostakin kohtuullisen kokoisesta kuvajoukosta, ja hakutulokset vastaavat esimerkkikuvaa melko hyvin näiden piirteiden osalta. Assosiaatioiden käyttö

tukee hakutoimintoja. Lisäksi assosiaatioiden avulla voi selvittää, mitkä värit ja tekstit liittyvät toisiinsa. Yhteyksien avulla voisi mahdollisesti myös esim. luoda kuville automaattisesti avainsana-annotaatioita, jos nämä otettaisiin järjestelmän yhdeksi piirrelajiksi.

## Viitteet

Ahmad, K., Casey, M., Vrusias, B., Saragiots, P. (2003) Combining Multiple Modes of Information using Unsupervised Neural Classifiers. *Multiple Classifier Systems: Proceedings of the 4<sup>th</sup> International Workshop* (toim. Windeatt, T., Roli, F.), Springer, Heidelberg, Saksa, 236-245.

Alahakoon, D., Halgamuge, S. K., Srinivasan, B. (2000) Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery. *IEEE Transactions on Neural Networks* **11**(3), 601-614.

Alhoniemi, E., Himberg, J., Parviainen, J., Vesanto, J. (2005) *SOM Toolbox 2.0 – kirjasto (versio 17.3.2005)*. (Saatavana: <http://www.cis.hut.fi/somtoolbox/download/>, haettu 16.9.2010)

Aslandogan, Y. A., Yu, C. T. (2000) Experiments in Using Visual and Textual Clues for Image Hunting on the Web. *Advances in Visual Information Systems: Proceedings of the 4<sup>th</sup> International Conference on Visual Information Systems* (toim. Laurini, R.), Springer, Heidelberg, Saksa, 108-119.

Blackmore, J., Miikkulainen, R. (1993) Incremental Grid Growing: Encoding High-Dimensional Structure into a Two-Dimensional Feature Map. *Proceedings of 1993 IEEE International Conference on Neural Networks* (toim. IEEE), IEEE Service Center, Piscataway, USA, 450-455.

Brandt, S., Laaksonen, J., Oja, E. (2002) Statistical Shape Features for Content-Based Image Retrieval. *Journal of Mathematical Imaging and Vision* **17**, 187-198.

Chuang, G. C.-H., Kuo, C.-C. J. (1996) Wavelet Descriptor of Planar Curves: Theory and Applications. *IEEE Transactions on Image Processing* **5**(1), 56-70.

Comon, P. (1994) Independent component analysis, a new concept? *Signal Processing* **36**(3), 287-314.

Daubechies, I. (1990) The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory* **36**(5), 961-1005.

Dittenbach, M., Rauber, A., Merkl D. (2001) Recent Advances with the Growing Hierarchical Self-Organizing Map. *Advances in Self-Organizing Maps: Proceedings of the 3rd Workshop on Self-Organizing Maps* (toim. Allinson, N., Yin, H., Allinson, L., Slack, J.), Springer, Heidelberg, Saksä, 140-145.

Frankel, C., Swain, M. J., Athitsos, V. (1996) *Webseer: An Image Search Engine for the World Wide Web*. Technical Report 96-14, The University of Chicago, Chicago, USA.

Freeman, H. (1974) Computer Processing of Line-Drawing Images, *ACM Computing Surveys* **6**(1), 57-97.

Fritzke, B. (1995) Growing Grid - A self-organizing network with constant neighborhood range and adaption strength. *Neural Processing Letters*, **2**(5), 9-13.

Golub, G. H., Kahan, W. (1965) Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis* **2**(2), 205-224.

Haralick, R., Shanmugam, K., Dinstein, I. (1973) Textual Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics* **3**(6), 610-621.

Hebb, D. O. (1949) *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, New York, USA.

Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**(6), 417-441.

Hsu, W., Chua, T. S., Pung, H. K. (1995) An Integrated Color-Spatial Approach to Content-Based Image Retrieval. *Proceedings of 3rd International ACM Multimedia Conference* (toim. Zellweger, P.), ACM Press, New York, USA, 305-313.

Hu, M.-K. (1962) Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory* **8**(2), 179-187.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., Zabih, R. (1997) Image Indexing Using Color Correlograms. *Proceedings of 1997 IEEE Conference on Computer Vi-*

sion and Pattern Recognition (toim. Plummer, D., Torwick, I.), IEEE Service Center, Piscataway, USA, 762-768.

Im, D.-J., Lee, M., Lee, Y. K., Kim, T.-E., Lee, S. W., Lee J., Lee, K. K., Cho, Y. D. (2005) Similarity Retrieval Based on Self-organizing Maps. *International Conference on Computational Science and its Applications 2005* (toim. Gervasi, O., Gavrilova, M. L., Kumar, V., Laganà, A., Lee, H. P., Mun, Y., Taniar, D., Tan, C. J. K.), Springer, Heidelberg, Saksa 474-482.

Johnsson, M., Balkenius, C. (2008) Associating SOM Representations of Haptic Submodalities. *Towards Autonomous Robotic Systems 2008* (toim. Ramamoorthy, S.), The University of Edinburgh, Edinburgh, UK, 124-129.

Johnsson, M., Balkenius, C. (2009) Associative Self-Organizing Map. *Proceedings of the International Joint Conference on Computational Intelligence 2009*, (toim. Correia, A. D., Rosa, A. C., Madani, K.), INSTICC Press, Setúbal, Portugali, 363-370.

Kaski, S. (1998) Dimensionality Reduction by Random Mapping: Fast Similarity Method for Clustering. *Proceedings of 1998 IEEE International Joint Conference on Neural Networks: IEEE World Congress on Computational Intelligence* (toim. IEEE), IEEE Service Center, Piscataway, USA, 413-418.

Kato, T. (1992) Database architecture for content-based image retrieval. *Proceedings of SPIE: Image Storage and Retrieval Systems* (toim. Sethi, I. K., Jain, R. J.), SPIE Press, Bellingham, USA, 112-123.

Khotanzad, A., Hong, Y. H. (1990) Invariant Image Recognition by Zernike Moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **12**(5), 489-497.

Kohonen, T. (1982) Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, **43**(1), 59-69.

- Koikkalainen, P., Oja, E. (1990) Self-Organizing Hierarchical Feature maps. *Proceedings of 1990 International Joint Conference on Neural Networks* (toim. IEEE), IEEE Service Center, Piscataway, USA, 279-284.
- Koskela, M. (2003) *Interactive Image retrieval using self-organizing maps*. Helsinki University of Technology, Dissertations in Computer and Information Science, Otamedia Oy, Espoo.
- Koskela, M., Laaksonen, J. (2003) Using Long-Term Learning to Improve Efficiency of Content-Based Image Retrieval. *Proceedings of the 3<sup>rd</sup> International Workshop on Pattern Recognition in Information Systems* (toim. Ogier, J.-M., Trupin, É.), ICEIS Press, Setúbal, Portugal, 72-79.
- Kruskal, J. B. (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **29**(1), 1-27.
- Laakso, S., Laaksonen, J., Koskela, M., Oja, E. (2001) Self-Organizing Maps of Web Link Information. *Advances in Self-Organising Maps* (toim. Allinson, N., Yin, H., Allinson, L., Slack, J.) , Springer, Lontoo, Iso-Britannia, 146-151.
- Laaksonen, J., Oja, E., Koskela, M., Brandt, S. (2000) Analyzing Low-Level Visual Features Using Content-Based Image Retrieval. *Proceedings of the 7<sup>th</sup> International Conference on Neural Information Processing*, (toim. Lee, S.-Y.), Korea Advanced Institute of Science and Technology, Daejeon, Korea, 1333-1338.
- Liu, F., Picard, R. W. (1996) Periodicity, directionality and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(7), 722-733.
- Mao, J., Jain, A. K. (1992) Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition* **25**(2), 173-188.
- Minka, T. P., Picard, R. W. (1997) Interactive learning with a “society of models”. *Pattern Recognition* **30**(4), 565-581.

Mitchell, B. (2013) *URL*.  
<http://compnetworking.about.com/od/internetaccessbestuses/g/bldef-url.htm> WWW-sivu (30.9.2013).

Natsev, A., Smith, J. R. (2002) A study of image retrieval by anchoring. *Proceedings of 2002 IEEE International Conference on Multimedia and Expo* (toim. SuviSoft Oy), IEEE Service Center, Piscataway, USA, 421-424.

Newsam, S., Sumengen, B., Manjunath, B. S. (2001) Category-based image retrieval. *Proceedings of 2001 IEEE International Conference on Image Processing* (toim. Mercer, B., Conference Management Services), IEEE Service Center, Piscataway, USA, 596-599.

Niblack, W., Barber, R., Equitz, W., Flickner, M., Petkovic, E. H. G. D., Yanker, P., Faloutsos, C., Taubin, G. (1993) The QBIC Project: Querying Images By Content Using Color, Texture and Shape. *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases* (toim. Niblack, W.), SPIE Press, Bellingham, USA, 173-187.

Ogle, V. E., Stonebraker, M. (1995) Chabot: Retrieval from a Relational Database of Images. *IEEE Computer* **28**(9), 40-48.

Oh, K. S., Zaher, A., Kim, P. K. (2002) Fast k-NN Image Search with Self-Organizing Maps. *Proceedings of International Conference on Image and Video Retrieval 2002* (toim. Lew, M. S., Sebe, N., Eakins, J. P.), Springer, Heidelberg, Saksa 299-308.

Pampalk, E., Chan, A. (2002) *GHSOM Toolbox for Matlab 5 -kirjasto*. (Saatavana: <http://www.ifs.tuwien.ac.at/~andi/ghsom/download.html>, haettu 16.9.2010)

Pass, G., Zabih, R., Miller, J. (1996) Comparing Images Using Color Coherence Vectors. *Proceedings of the 4<sup>th</sup> ACM International Conference on Multimedia 1996* (toim. Aigrain, P., Hall, W., Little, T. D. C., Bove, V. M. Jr.), ACM Press, New York, USA, 65-73.

- Patil, P. B., Kokare, M. B. (2011) Relevance Feedback in Content Based Image Retrieval: A Review. *Journal of Applied Computer Science & Mathematics* **10**(5), 41-47.
- Pentland, A., Picard, R. W., Sclaroff, S. (1996) Photobook: Content-Based Manipulation of Image Databases. *International Journal of Computer Vision* **18**(3), 233-254.
- Rafiee, G., Dlay, S. S., Woo, W. L. (2010) A Review of Content-Based Image Retrieval. *2010 7th International Symposium on Communication Systems, Networks and Digital Signal Processing* (toim. Ghassemlooy, Z., Ng, W. P.), IEEE Service Center, Piscataway, USA, 775-779.
- Rauber, A., Merkl, D., Dittenbach, M. (2002) The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data. *IEEE Transactions on Neural Networks* **13**(6), 1331-1341.
- Sclaroff, S., La Cascia, M., Sethi, S., Taycher, L. (1999) Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web. *Computer Vision and Image Understanding* **75**(1/2), 86-98.
- Shete D. S., Chavan M. S. (2012) Content Based Image Retrieval: Review. *International Journal of Emerging Technology and Advanced Engineering* **2**(9), 85-90.
- Smith, J. R., Chang, S.-F. (1995) Single Color Extraction and Image Query. *Proceedings of 1995 IEEE International Conference on Image Processing* (toim. IEEE), IEEE Service Center, Piscataway, USA, 528-531.
- Smith, J. R., Chang, S.-F. (1997) Visually Searching the Web for Content. *IEEE Multimedia* **4**(3), 12-20.
- Stricker, M., Orengo, M. (1995) Similarity of Color Images. *Storage and Retrieval for Image and Video Databases III* (toim. Niblack, W., Jain, R.), SPIE Press, Bellingham, USA, 381-392.
- Swain, M. J., Ballard, D. H. (1991) Color Indexing. *International Journal of Computer Vision* **7**(1), 11-32.



Sychay, G., Chang, E., Goh, K. (2002) Effective image annotation via active learning. *Proceedings of 2002 IEEE International Conference on Multimedia and Expo* (toim. SuviSoft Oy), IEEE Service Center, Piscataway, USA, 209-212.

Szumner, M., Picard, R. W. (1998) Indoor-Outdoor Image Classification. *Proceedings of 1998 IEEE International Workshop on Content-Based Access of Image and Video Database* (toim. Spencer Sipple, R.), IEEE Service Center, Piscataway, USA, 42-51.

Tamura, H., Mori, S., Yamawaki, T. (1978) Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics* **8**(6), 460-473.

Turner, M. R. (1986) Texture Discrimination by Gabor Functions. *Biological Cybernetics* **55**(2/3), 71-82.

Vailaya, A., Jain, A., Zhang, H. J. (1998) On image classification: City images vs. landscapes. *Pattern Recognition* **31**(12), 1921-1935.

Vailaya, A., Figueiredo, M. A. T., Jain, A. K., Zhang, H.-J. (2001) Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing* **10**(1), 117-130.

Vleugels, J., Veltkamp, R. C. (2002) Efficient Image Retrieval through Vantage Objects. *Pattern Recognition* **35**(1), 69-80.

Wang, J. Z., Liu, J., Wiederhold, G. (2001) SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(9), 947-963.

Wikipedia (2013a) *Lab color space*. [http://en.wikipedia.org/wiki/Lab\\_color\\_space](http://en.wikipedia.org/wiki/Lab_color_space) WWW-sivu (30.9.2013).

Wikipedia (2013b) *sRGB*. <http://en.wikipedia.org/wiki/sRGB> WWW-sivu (18.8.2013).

World Wide Web Consortium (2013) *HTML& CSS*.  
<http://www.w3.org/standards/webdesign/htmlcss> WWW-sivusto  
(30.9.2013).

Zahn, C. T., Roskies, R. Z. (1972) Fourier Descriptors for Plane Closed Curves.  
*IEEE Transactions on Computers* **C-21**(3), 269-281.

Zhang, C., Chen, T. (2002) An Active Learning Framework for Content-Based Information Retrieval. *IEEE Transactions on Multimedia* **4**(2), 260-268.