

LAPPEENRANNAN TEKNILLINEN KORKEAKOULU

Tietotekniikan osasto

SPEKTRIKUVIEN TIIVISTÄMINEN
ITSEORGANISOITUVAN KARTAN AVULLA

Diplomityön aihe hyväksytty Tietotekniikan osaston osastoneuvostossa 15.9.1999.

Työn ohjaaja: Professori Pekka Toivanen

Tarkastajat: Professori Heikki Kälviäinen, Professori Pekka Toivanen

Lappeenrannassa 31.1.2000

Anssi Lehtinen

Punkkerikatu 1 B 24

53850 Lappeenranta

040- 765 5213

alehtine@necsom.com

TIIVISTELMÄ

Tekijä: Anssi Lehtinen
Nimi: Spektrikuvien tiivistäminen itseorganisoituvan kartan avulla
Osasto: Tietotekniikan osasto
Vuosi: 2000
Paikka: Lappeenranta
Diplomityö. Lappeenrannan teknillinen korkeakoulu
60 sivua, 25 kuvaa, 6 taulukkoa.
Tarkastajat: Professori Heikki Kälviäinen, Professori Pekka Toivanen
Avainsanat: Spektrikuva, tiivistäminen, itseorganisoituva kartta, kuvankäsittely, tietokonenäkö

Vuosi vuodelta kasvava tietokoneiden prosessointikyky on mahdollistanut harmaataso- ja RGB-värikuvia tarkempien spektrikuvien käsittelyn järjestetyn järjellisessä ajassa ilman suuria kustannuksia. Ongelmana on kuitenkin, ettei talletus- ja tiedonsiirtomedia ole kehittynyt prosessointikyvyn vauhdissa. Ratkaisu tähän ongelmaan on spektrikuvien tiivistäminen talletuksen ja tiedonsiirron ajaksi.

Tässä työssä esitellään menetelmä, jossa spektrikuva tiivistetään kahdessa vaiheessa: ensin ryhmittelemällä itseorganisoituvan kartan (SOM) avulla ja toisessa vaiheessa jatketaan tiivistämistä perinteisin menetelmin. Saadut tiivistyssuhteet ovat merkittäviä vääristymän pysyessä siedettävänä.

Työ on tehty Lappeenrannan teknillisen korkeakoulun Tietotekniikan osaston Tietojenkäsittelytekniikan tutkimuslaboratoriossa osana laajempaa kuvantiivistyksen tutkimushanketta.

ABSTRACT

Author: Anssi Lehtinen

Title: Compressing spectral images using the Self-Organizing Map

Department: Department of Information Technology

Year: 2000

Place: Lappeenranta

Master's thesis. Lappeenranta University of Technology

60 pages, 25 figures, 6 tables.

Supervisors: Professor Heikki Kälviäinen, Professor Pekka Toivanen

Keywords: spectral image, compression, self-organized map, image processing,
computer vision

Evergrowing computing power has made it possible to process more and more accurate spectral images in reasonable time without huge costs. The problem is that performance of storage and transmission media is not up to the task when dealing with large spectral images. One way to overcome this is to compress spectral images.

In this work, a method for compressing spectral images in two stages is presented. In the first stage the image is clustered using the Self-Organizing Map (SOM), and in the second stage the image is compressed using more traditional techniques. Compression rates achieved are substantial and the distortion in an image is tolerable.

This thesis was done in the Laboratory of Information Processing of the Department of Information Technology of the Lappeenranta University of Technology as a part of image compression research.

ALKUSANAT

Työ spektrikuvien tiivistäminen itseorganisoituvan kartan avulla perustuu Lappeenrannan teknillisen korkeakoulun Tietotekniikan osaston Tietojenkäsittelytekniikan tutkimuslaboratoriossa syksyllä 1998 kehitettyyn menetelmään, jolla voidaan tiivistää spektrikuvia. Itseorganisoituvan kartan opetuksessa käytettiin valmista ohjelmaa ja muu työ toteutettiin Matlab-ympäristössä. Apuna käytettiin muita yleisiä Unix-ympäristön työkaluja, kuten xv:tä.

Kiitän työn tarkastajia prof. Pekka Toivasta ja prof. Heikki Kälviäistä, joista Pekka Toivanen toimi myös työn ohjaajana.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. VALO, VÄRI JA KUVAT	2
2.1. Ultraviolettivalo.....	2
2.2. Infrapunavalo	3
2.3. Näkyvä valo.....	4
2.3.1. Kirkkaus, sävy ja puhtaus.....	5
2.3.2. Värit.....	6
2.3.3. Värien yhdistyminen	7
2.4. Värimallit	8
2.4.1. RGB-malli	8
2.4.2. CMYK-malli	9
2.4.3. HSV-malli	9
2.4.4. CIE-väristandardi	9
2.5. Kuvat	12
2.5.1. Vektorikuvat.....	13
2.5.2. Rasterikuvat.....	14
2.5.3. Spektrikuvat.....	15
3. TIIVISTÄMINEN	16
3.1. Tiivistyksen ominaisuudet	17
3.2. Erimittaiset koodisanat	19
3.3. HUFFMAN-koodaus.....	20
3.4. Kuvien tiivistäminen yleisesti.....	21
3.4.1. Sarjamuotoon muuntaminen.....	23
3.4.2. Sisällön vähentäminen.....	24
3.4.3. Deltakoodaus	25
3.4.4. Ennakoiva koodaus.....	25
3.4.5. Todennäköisyyksien laskeminen ja koodisanan sijoitus	27
3.5. Hukkaava tiivistäminen.....	27
3.5.1. Kvantisointi	28
3.5.2. Vektorikvantisointi	29
4. ITSEORGANISOITUVA KARTTA	31
4.1. Itseoppivat neuroverkot.....	31

4.2. Kohosen itseorganisoituva kartta.....	32
4.2.1. Itseorganisoituvan kartan rakenne	32
4.2.2. Kartan alustus ja opetus.....	33
5. KEHITETTY TIIVISTÄMISMENETELMÄ	36
5.1. Spektrikuvan tiivistäminen.....	37
5.2. Spektrikuvan palauttaminen.....	39
6. MENETELMÄN SOVELTAMINEN	40
6.1. Parannuksia menetelmään	41
6.1.1. Deltakoodaus	42
6.1.2. 2-ulotteinen ennakointi	42
6.1.3. Huffman-koodaus	42
6.2. Muita parannusyrityksiä	45
6.3. Vertailua kartan ja indeksin tiivistämiseen muilla menetelmillä.....	47
7. YHTEENVETO	49
LÄHTEET	51

SYMBOLI- ja LYHENNELUETTELO

λ	aallonpituus
H	entropia
h	naapuruusfunktio
$I(\lambda)$	intensiteetti jollakin aallonpituudella
\bar{L}	koodisanan keskimääräinen pituus
p	todennäköisyys
R	redundanssi
w	neuroni, painovektori
BMP	BitMap format
BMU	Best Matching Unit
CALIC	Context-based, Adaptive, Lossless Image Codec
CIE	Commission Internationale de l'Éclairage
CMYK	Cyan, Magenta, Yellow, Black
FELICS	Fast, Efficient and Lossless Image Compression System
GIF	Graphics Interchange Format
HSV	Hue, Saturation, Value
JPEG	Joint Photographic Experts Group
PCX	PC paintbrush format
RGB	Red, Green, Blue
SOM	Self-Organizing Map
TIFF	Tagged Image File Format
TMW	Tischer, Meyer, Worley
UV	Ultra Violet

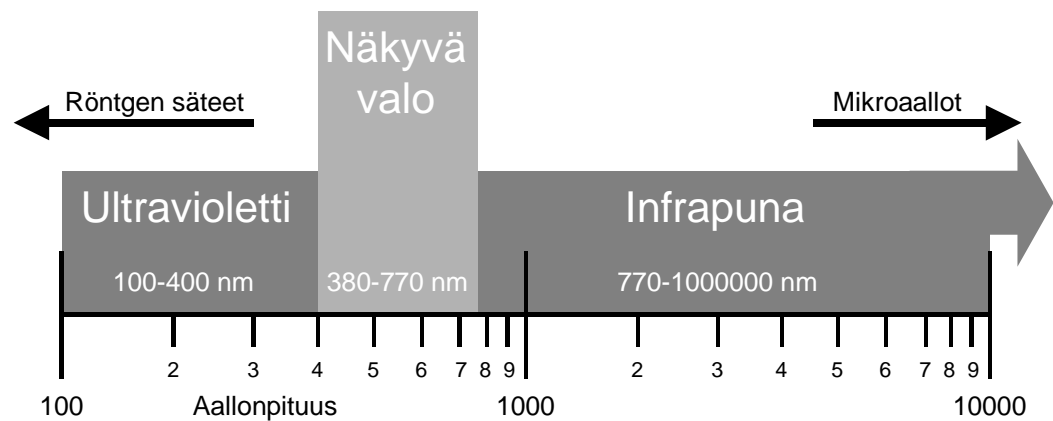
1. JOHDANTO

Vuosi vuodelta kasvava tietokoneiden prosessointikyky on mahdollistanut harmaataso- ja RGB-värikuvia tarkempien spektrikuvien käsittelyn järjestetyn ajassa ilman suuria kustannuksia. Ongelmana on kuitenkin, ettei talletus- ja tiedonsiirtomedia ole kehittynyt prosessointikyvyn vauhdissa. Ratkaisu tähän ongelmaan on spektrikuvien tiivistäminen talletuksen ja tiedonsiirron ajaksi. Yleiset tiivistysalgoritmit eivät kuitenkaan ole niin hyviä kuin spektrikuvien ominaisuudet huomioon ottavat menetelmät. Tällaisia menetelmiä onkin kehitetty, mutta mikään menetelmä ei ole vakiintunut käyttöön. Tässä työssä tarkastellaan LTKK:n Tietotekniikan osaston Tietojenkäsittelytekniikan tutkimuslaboratoriossa kehitettyä tietoa hukkaavaa menetelmää spektrikuvien tiivistämiseksi, jossa käytetään apuna neuraalilaskennan itseoppivaa Kohosen karttaa (Self-Organizing Map, SOM).

Luvussa 2 käsitellään mitä ovat valo, väri ja kuvat ja miten ne liittyvät toisiinsa. Luvussa 3 käsitellään yleistä tiedon tiivistämistä ja miten kuvia tiivistetään. Luvussa 4 käsitellään itseoppivaa Kohosen karttaa. Luvuissa 5 ja 6 käsitellään kehitettyä menetelmää ja sen soveltamista. Luvussa 7 on yhteenveto työstä.

2. VALO, VÄRIT JA KUVAT

Näkyvä valo kattaa vain osan maailmankaikkeuden elektromagneettisista aalloista. Sähkömagneettinen spektri kattaa alueen useamman metrin aallonpituuden radioaalloista aina röntgensäteisiin, joiden aallonpituus on metrin miljardisosa. Optinen säteily sijaitsee radioaaltojen ja röntgensäteilyn välillä (kuva 1) ja sisältää säteily-, aalto- ja kvanttiominaisuuksia.

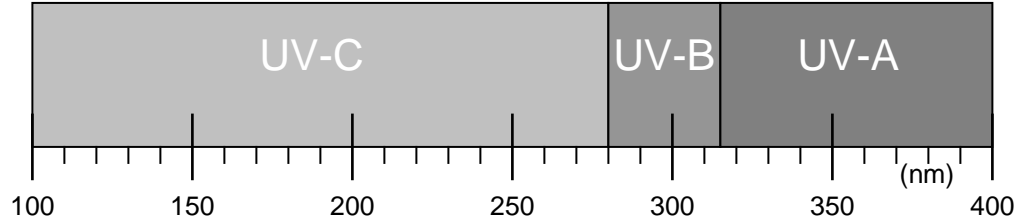


Kuva 1. Näkyvä valo spektrissä.

Röntgensäteilyssä ja sitä lyhyemmissä aallonpituuksissa elektromagneettinen säteily on hiukkasluonteista, kun taas pidemmillä aallonpituuksilla se on enimmäkseen aaltomuotoista. Näkyvä valo sijaitsee niiden välissä ja on ominaisuuksiltaan sekä aalto- että hiukkasluonteista.

2.1. Ultraviolettivalo

Ultraviolettivalo (UV) on ominaisuuksiltaan enemmän kvanttiluonteinen kuin näkyvä tai infrapunavalon. Ultraviolettivalo jakautuu kolmeen kaistaan ominaisuuksiensa mukaan (kuva 2).



Kuva 2. Ultraviolettikaista.

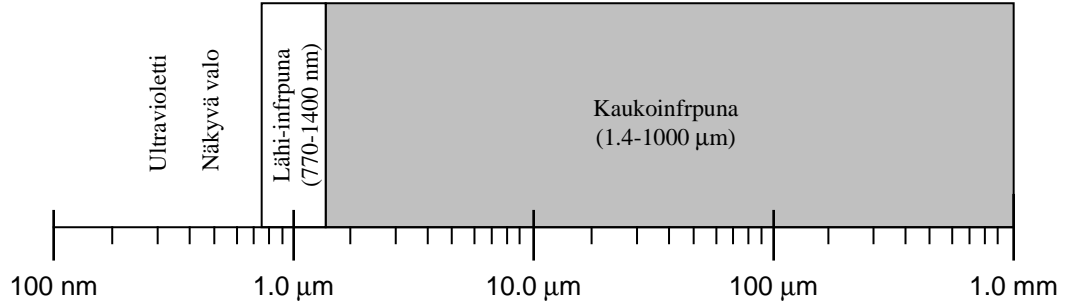
UV-A on yleisin ja harmittomin ultraviolettivalo, koska se sisältää vähiten energiaa. UV-A valoa kutsutaan myös mustaksi valoksi ja sitä käytetään valoterapiaan ja solariumeissa.

UV-B on tuhoisin ultraviolettivalon muoto, koska se sisältää tarpeeksi energiaa vahingoittaakseen biologisia kudoksia, muttei tarpeeksi energiaa, jotta ilmakehä kokonaan absorboisi sen. UV-B aiheuttaa ihosyöpää ja koska ilmakehä pysäyttää suurimman osan maan ulkopuolisesta UV-B säteilystä, muutokset otsonikerroksessa voivat huomattavasti kasvattaa riskiä saada ihosyöpä.

Lyhyimmän aallonpituuden UV-C absorboituu lähes täydellisesti ilmassa jo muutaman sadan metrin matkalla. Kun UV-C fotonit törmäävät happimolekyyleihin, energiavaihto muodostaa otsonia. UV-C lampuja käytetään puhdistamaan ilmaa ja vettä niiden bakteereja tappavan ominaisuuden takia /18/.

2.2. Infrapunavalo

Infrapunavalo (kuva 3) sisältää valon kaistoista vähiten energiaa fotonia kohden. Infrapunafotoni ei tästä johtuen usein sisällä tarpeeksi energiaa, jotta se havaittaisiin kvantti-ilmaisimella. Infrapunavaloa mitataankin usein käyttämällä lämpöilmaisimia, jotka mittaavat absorboituneen energian synnyttämää lämpötilan muutosta.

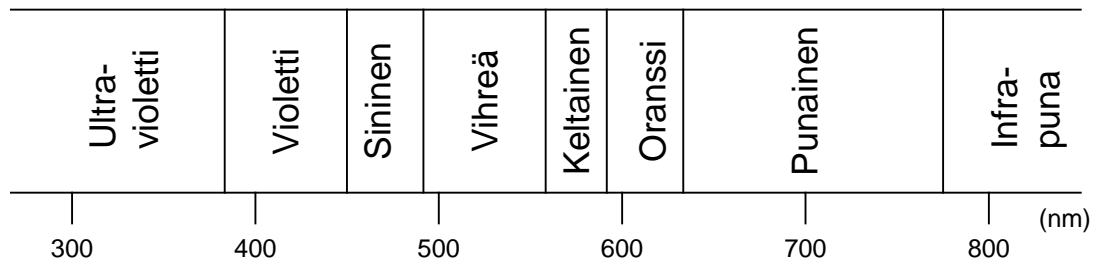


Kuva 3. Infrapunakaista.

Koska lämpö on eräs infrapunavalon muoto, kauko-infrapunailmaisimet ovat herkkiä ympäristön lämmönvaihteluille, kuten esimerkiksi ihmisen liikkumiselle havaintoalueella. Pimeänäkölaitteet hyödyntävätkin tätä ilmiötä ja vahvistavat infrapunavaloa jotta ihmiset ja koneet erottuvat pimeydestä /18/.

2.3. Näkyvä valo

Ihmisen silmät ja aivot aistivat valona sellaisen sähkömagneettisen säteilyn, jonka aallonpituus (λ) on 400 - 700 nm. Hyvissä olosuhteissa valona aistitaan jopa aallonpituudet 380 - 770 nm. Kuvassa 4 on näkyvän valon päävärit niiden aallonpituuden mukaisissa alueissa. Yleensä valossa esiintyy kaikkia näkyvän valon aallonpituuksia. Esimerkiksi prismalla valkoinen valo voidaan jakaa spektriksi. Spektrin värit ovat puhtaita, eli monokromaattisia värejä, joita kutakin vastaa jokin aallonpituus λ . Sekavalossa kukin spektrin väri esiintyy omalla intensiteetillään $I(\lambda)$ /16/.



Kuva 4. Näkyvän valon värien alueet aallonpituuksien mukaan. /7/

Kappaleen tai pinnan väri määräytyy sen ominaisuudesta heijastaa tai lähettää tiettyjä aallonpituuksia eli suoraan sen lähettämän tai heijastaman valon intensiteetti- eli energijakaumasta $I(\lambda)$. Esimerkiksi keltaiset pinnat näkyvät keltaisina, koska ne heijastavat suurimman osan niihin osuvista keltaisen valon aallonpituuksista 570-580 nm ja absorboivat muita aallonpituuksia.

Intensiteettijakaumassa $I(\lambda)$ on kuitenkin värien aistimisen kannalta liikaa informaatiota, koska moni erilainen jakauma "näyttää" samalta "väriltä" eli vastaa fysiologisesti samaa väriaistimusta. Tällaisia samaa väriä vastaavia intensiteettijakaumia sanotaan toistensa metameereiksi.

2.3.1. Kirkkaus, sävy ja puhtaus

Värien luokitteluksi tarvitaan vain kolme fysikaalista, tai oikeammin fysiologista parametria: kirkkaus, sävy ja puhtaus. Tämä perustuu silmän fysiologisiin ominaisuuksiin (tristimulusteoria /15/).

Kirkkaus tai vaaleus kuvaa värin vaaleus- tai tummuusastetta. Se vastaa silmän havaitsemaa maksimi- tai kokonaisintensiteettiä. Mutta fysiologisille suureille ominaisesti kirkkaus käyttäytyy logaritmisesti, koska silmä havaitsee vain suhteellisia eroja. Murretut värit, kuten ruskea, saadaan vähentämällä kirkkautta eli sekoittamalla

väriin mustaa, tai vähentävässä järjestelmässä lisäämällä vastavärin pigmenttiä, eli väriainetta.

Sävy vastaa valon hallitsevaa aallonpituutta, joka ei välttämättä ole aallonpituus, jossa intensiteetti $I(\lambda)$ on suurimmillaan. Joissakin väreissä esiintyy esimerkiksi useampia intensiteettihiippuja, jolloin värillä ei ole hallitsevaa aallonpituutta. Väriaistimus voi tällaisessa tapauksessa tuntua aivan muuta aallonpituutta vastaavalta väriltä, tai sitten värillä ei yksinkertaisesti ole spektrissä vastinetta, metameeria. Tällaisia spektristä löytymättömiä, mutta kuitenkin puhtaiksi katsottuja värejä, joilla ei ole dominoivaa aallonpituutta, sanotaan purppurasävyiksi. Purppuravärejä ovat esimerkiksi purppura ja magenta. Ne aistitaan, kun silmään osuu samanaikaisesti sekä pitkäaaltoista punaista että lyhytaaltoista violettiä säteilyä.

Puhtaus eli kylläisyys kertoo, kuinka paljon muita aallonpituuksia keskimäärin on värin intensiteettijakaumassa mukana. Värin puhtaus vähenee, kun väriin sekoittuu valkoista tai additiivisessa järjestelmässä vastaväriä. Monokromaattisilla spektrin väreillä puhtaus on suurimmillaan ja valkoiselle valolle puhtaus on pienimmillään /16/.

2.3.2. Värät

Ihminen erottaa täysin kylläisiä puhtaita värejä noin 1-4 nm:n aallonpituusvälein, kaikkiaan ehkä noin 130 eri sävyä. Puhtausasteita erotetaan noin 20, joten kun vielä otetaan huomioon useita satoja kirkkausvaihtoehtoja, ihminen erottaa kaikkiaan satojatuhansia eri värejä.

Pääsävyt ovat väriympyrän mukaisesti oranssi, punainen, magenta, violetti, sininen, syaaninsininen, vihreä ja keltainen. Sävyttömiä värejä ovat musta, jossa ei ole väriä eikä valoa ollenkaan, ja valkoinen, jolla on tasainen spektri, eli jossa on tasaisesti

kaikkia värejä suurimmalla intensiteetillä. Harmaan sävyt ovat mustan ja valkoisen välissä ja myös niillä on tasainen spektri, mutta intensiteetti ei ole suurimmillaan. Ruskea on taitettu eli murrettu punainen, oranssi tai keltainen.

Värejä luokitellaan usein vertaamalla niitä julkistettuihin standardivärikarttoihin. Värikarttoja ovat esimerkiksi Munsell /14/ tai maalikaupan värikartta.

2.3.3. Värien yhdistyminen

Erilaisia värien yhdistymis- ja sekoittumistapoja on neljä: vähentävä, summaava ja summaavan alalajit optinen ja diffuusi.

Vähentävässä värinmuodostuksessa värisävyt syntyvät pigmenttien yhdistymisestä, jolloin valon määrä vähenee, koska väriaineita sekoitettaessa heijastuva valo vähenee valon absorption kasvaessa. Yhdistämällä esimerkiksi punaista ja keltaista väriainetta saadaan aikaan oranssi sävy. Syntyvä väri on tummempi kuin lähtövärit keskimääräisesti. Syntyvä väri on myös lähtövärejä likaisempi, murrettu. Vähentävällä menetelmällä on vaikeaa tuottaa vaaleaa keltaista sävyä, jos sitä ei ole lähtövärinä käytettävissä. Vähentävää menetelmää käytetään tulostimissa ja maalausteitteissa.

Summaavassa värinmuodostuksessa värisävyt syntyvät värillisten valonlähteiden yhdistymisestä, jolloin valon määrä lisääntyy. Yhdistämällä esimerkiksi punaista ja vihreää valoa saadaan aikaan keltainen valo. Syntyvä väri on kirkkaampi kuin lähtövärit keskimääräisesti. Syntyvä väri on myös puhtaampi kuin lähtövärit. Summaamalla menetelmällä on vaikea tuottaa esimerkiksi tummaa violettiä sävyä, jos sitä ei ole lähtövärinä käytettävissä. Summaavaa menetelmää käyttävät esimerkiksi kuvaputkeen perustuvat näyttölaitteet.

Optisessa yhdistymisessä värisävyt syntyvät värien sekoittumisesta liikkeen vuoksi. Tuttuna esimerkkinä on sektoreihin jaettujen levyjen pyörittäminen. Menetelmällä ei ole suurempaa merkitystä tietokonegrafiikassa.

Diffuusissa sekoittumisessa samanaikaiset väriaistimukset sekoittuvat. Kaukaa katsottuna kohteesta erotetaan vain intensiteettien keskiarvot silmän toimiessa alipäästösuodattimena.

Sekoitusmenetelmässä käytetään hyväksi diffuusista sekoittumista. Käytettävissä olevista väreistä muodostetaan matriisin avulla lisävärejä. Tätä värien muodostustapaa käytetään hyväksi esimerkiksi tulostettaessa harmaasävykuvia mustavalkotulostimella /16/.

2.4. Värimallit

Värimalleja käytetään värien luokitteluun ja standardisointiin. Tärkeimpiä tietotekniikassa käytettyjä värimalleja ovat RGB, CMYK, HSV ja CIE /7, 15, 16, 17/.

2.4.1. RGB-malli

Tietokonemonitoreissa ja televisiovastaanottimissa käytetty RGB-malli koostuu kolmesta komponentista, punaisen (Red), vihreän (Green) ja sinisen (Blue) valon määrästä. Muut värit saadaan yhdistämällä näitä kolmea komponenttia. RGB-malli on summaava, eli komponenttien intensiteetit summataan yhteen, jolloin saatu väri on kirkkaampi kuin komponentit. Jos kaikkien komponenttien arvo, eli intensiteetti, on yhtäsuuri, saadaan jokin harmaan sävy, tai musta tai valkoinen.

2.4.2. CMYK-malli

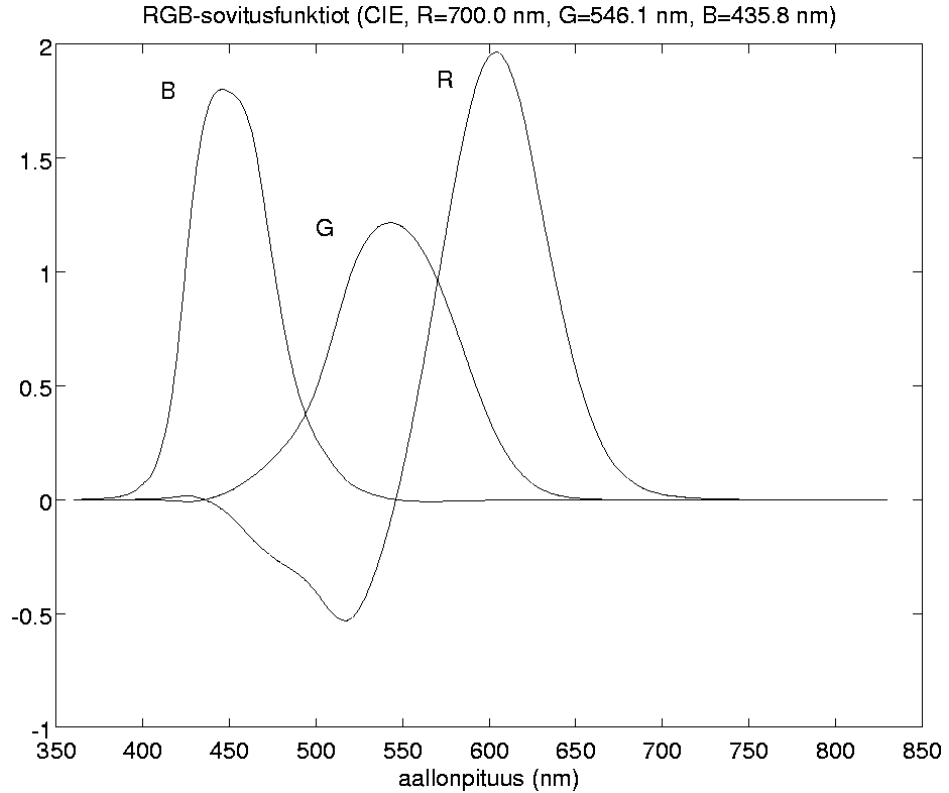
Maalaustaiteessa, tulostimissa ja neliväripainamisessa käytetty CMYK-malli koostuu neljästä komponentista: syaaninsinisestä, magentasta, keltaisesta ja mustasta. Muut värit luodaan yhdistämällä näitä neljää väriä. Teoriassa mustaa väriä ei tarvita, mutta sitä käytetään, jotta tulos saadaan terävämmäksi. CMYK on vähentävä värimalli, eli mitä suurempia määriä komponentteja yhdistetään, sitä vähemmän ne heijastavat valoa.

2.4.3. HSV-malli

HSV-malli perustuu sävyyn (Hue), puhtauteen (Saturation) ja kirkkauteen (Value). Kappaleessa 2.3.1. on kuvattu kuinka niiden perusteella voidaan esittää värejä.

2.4.4. CIE-väristandardi

Tärkein tällä hetkellä käytössä oleva väristandardi on kansainvälisen valaistuskomission (Commission Internationale de l'Éclairage, CIE) vuonna 1931 vahvistama XYZ-värikoordinaatistojärjestelmä. Tämän standardin avulla voidaan jokainen väri määrittellä yksiselitteisesti. Vaikka jokainen väri voidaan esittää summaavalla yhdistämisellä kolmen perusvärin avulla, perusvärien määrät vaihtelevat hieman havainnoijasta riippuen. CIE määrittä standardihavainnoitsijan laskemalla perusvärien määrille keskiarvon suuresta joukosta eri havainnoitsijoiden tuloksia. Kuvassa 5 on esitetty saadut sovituskäyrät, jotka määrittävät kullekin spektrin värille näiden kolmen puhtaan perusvärin valoisuudet, joiden perusteella havainnoitsija saa saman väriaistimuksen kuin tarkasteltavasta puhtaasta spektriväristä. Sovituskäyrät kertovat siis, kuinka paljon kutakin perusväriä tarvitaan kutakin aallonpituutta vastaavan värin saamiseksi /16/.



Kuva 5. Kokeelliset fysiologiset sovituskäyriä perusvääreille. /16/

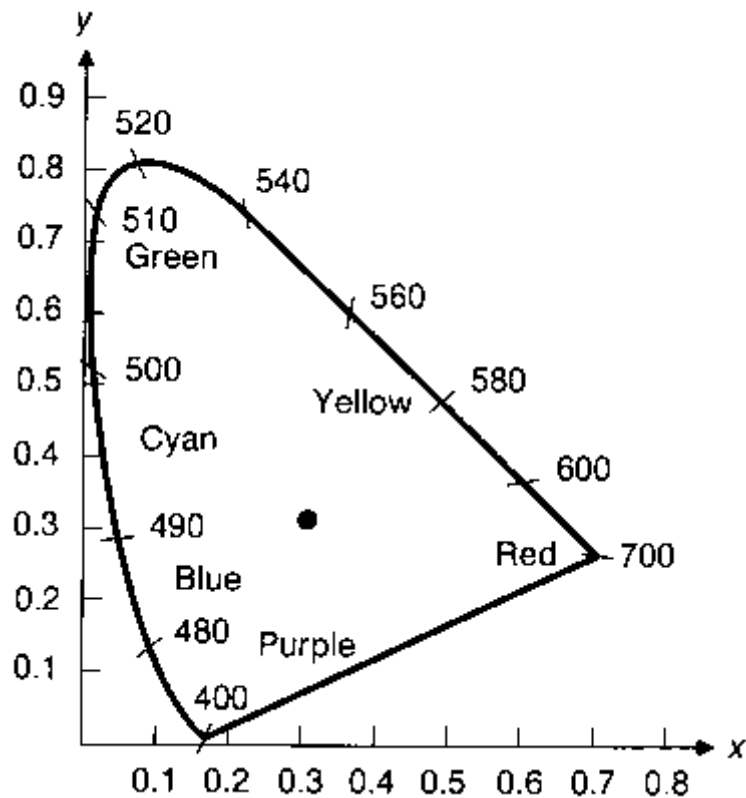
Jotta välttyäisiin negatiivisilta sovituskäyriä arvoilta, CIE määrittä uudet tristimulusarvot X, Y ja Z, jotka saadaan tristimulusarvoista R, G, ja B seuraavien yhtälöiden avulla:

$$\begin{aligned}
 X &= 0.49R & + 0.31G & + 0.20B \\
 Y &= 0.17697R + 0.81240G & + 0.01063B & \\
 Z &= 0.00R & + 0.01G & + 0.99B
 \end{aligned}
 \tag{1}$$

CIE-koordinaattien X, Y ja Z sijasta yleensä annetaan kaksi kromaattisuuskoordinaattia x ja y ja luminanssi Y. Tristimulusarvojen X, Y ja Z kromaattisuuskoordinaatit ovat

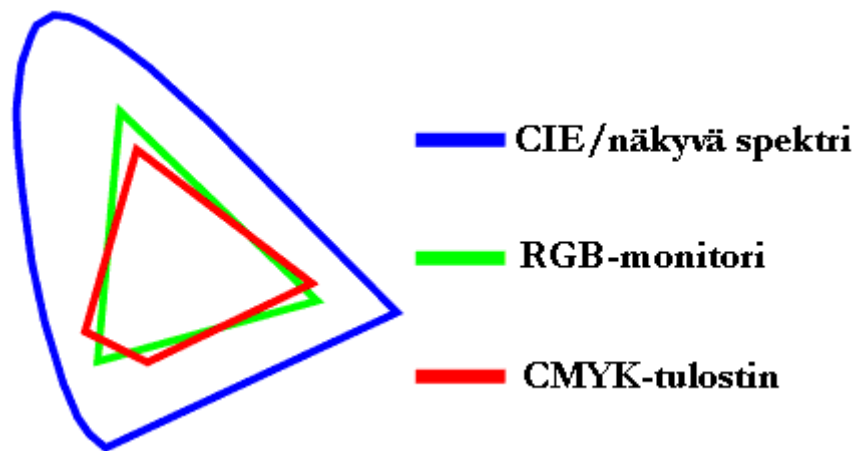
$$\begin{aligned}
 x &= \frac{X}{X+Y+Z} \\
 y &= \frac{Y}{X+Y+Z} \\
 z &= \frac{Z}{X+Y+Z}
 \end{aligned}
 \tag{2}$$

Koska $x+y+z=1$, jokainen ajateltavissa oleva väri voidaan esittää kahden kromaattisuuskoordinaatin x ja y ja valon heijastusprosentin tai luminanssin Y avulla. Kukin väri vastaa pistettä (x,y) kuvan 6 CIE-kromaattisuuskartassa. Mahdolliset värit muodostavat alueen, jota rajoittaa hevosenkengän muotoinen spektrikäyrä, jolla ovat kirjon puhtaat värit ja suora "purppuraviiva", joka sulkee hevosenkengän. Puhtaat värit sijoittuvat siis reunalle. Alueen keskellä on valkopiste. Usein reunakäyrä "venytetään" ympyräksi, jolloin saadaan väriympyrä.



Kuva 6. CIE- kromaattisuuskartta. /16/

Mitkä tahansa kolme väriä voivat yhdistettäessä tuottaa kaikki ne värit, jotka ovat näitä värejä vastaavien pisteiden virittämän kolmion eli värialueen sisällä. Kolme väriä ei riitä tuottamaan aivan kaikkia värejä, valittiinpa nämä perusvärit miten tahansa, koska mikään kolmio ei CIE-kartan reunan muodosta johtuen tietenkään peitä aivan koko aluetta. Esimerkiksi tyypillisen näytön värialue peittää suurin piirtein vain oikeanpuoleisen alanurkkaan rajoittuvan puolen kartasta, ja tyypillisen väritulostimen värikolmio peittää tästäkin alueesta vain osan (kuva 7).



Kuva 7. RGB- ja CMYK-värit CIE kartassa.

2.5. Kuvat

Näkö on ihmisen tärkein aisti. Aistimme yli 99% ympäristöstämme silmien avulla. Näkökyvyn tärkeyttä vahvistaa myös se, kuinka suuri osa aivoista liittyy näkökykyyn ja hahmottamiseen verrattuna muiden aistien osuuksiin aivojen käytöstä. Näkökyvyn tärkeyttä päivittäisessä elämässä kuvaa hyvin, kuinka monet ihmiset käyttävät silmälaseja parantaakseen näkökykyään, kun taas kuulon heikentymistä siedetään hyvinkin pitkälle ennen kuuloa parantavan laitteen käyttöä. Ihmisen muita aisteja parantavia laitteita ei sinänsä ole olemassakaan, mausteita ja hajusteita käytetään paremminkin korostamaan tai peittämään tiettyjä makuja tai hajuja kuin auttamaan havainnoimisessa /17/.

Näkökyvyn hallitsevuus vaikuttaa myös lähestymistapaamme ratkoa teknisiä ongelmia. Laitteet useammin näyttävät tietoja käyttäjälle kuvien tai valon avulla kuin äänen tai hajun avulla. Tekniset laitteet myös vahvistavat näkökykyämme, mikroskoopin ja teleskoopin avulla pystymme näkemään huomattavasti pienempiä ja suurempia asioita kuin luontaisesti kykenisimme. Modernimpien laitteiden avulla pystymme laajentamaan näkökykyämme näkyvän valon lisäksi myös muillekin sähkömagneettisen säteilyn kaistoille. Infrapuna- ja ultravioletivalon lisäksi pystymme näkemään röntgensäteilyä ja radioaaltoja. Röntgen- ja ultraäänilaitteiden avulla muodostettujen kuvien avulla pystymme näkemään myös kiinteiden kappaleiden sisälle. Videokamera mahdollistaa tapahtumien tarkastelemisen jälkikäteen hidastetusti tai pysäytettynä johonkin tiettyyn hetkeen.

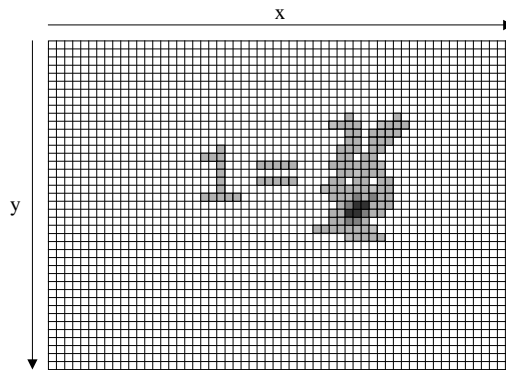
Sanotaan, että kuva kertoo enemmän kuin tuhat sanaa ja kuvien suuren informaation vuoksi ihmiset ovat aina käyttäneet niitä välittämään tietoja ja tunteita. Kuvan tarkoitus on siirtää tietoa paikasta ja ajasta toiseen. Tietokoneella käsiteltävät kuvat voivat olla monessa eri muodossa, joista yleisimmät ovat vektorikuva ja rasterikuva. Kuvat esittävät jotakin rajattua, äärellistä aluetta ja sisältävät äärellisen määrän informaatiota /3/.

2.5.1. Vektorikuvat

Vektorikuvat koostuvat pisteistä kaksi- tai kolmiulotteisessa koordinaatistossa ja lauseista, jotka yhdistävät nämä pisteet viivoilla tai käyrillä alueiksi. Kuviin voi liittyä myös valonlähteitä ja tekstuureita. Vektorikuvien etuna on pieni koko ja mahdollisuus tarkkailla kuvaa monesta eri kulmasta. Kaksiulotteisen kuvan kokoa voi suurentaa tai pienentää tai kiertää halutun pisteen ympäri. Kolmiulotteista kuvaa voidaan tarkastella mistä tahansa kulmasta ja avaruuden pisteestä. Vektorikuvien huono puoli on, että niiden luominen automaattisesti on hyvin hankalaa, ja suurin osa vektorikuvista onkin ihmisten manuaalisesti luomia.

2.5.2. Rasterikuvat

Rasterikuva eli bittikarttakuva on kaksiulotteinen, suorakaiteen muotoinen näkymä jostakin rajatusta kohteesta ja se koostuu äärellisestä joukosta kuvaelementtejä, pikseleitä (kuva 8).



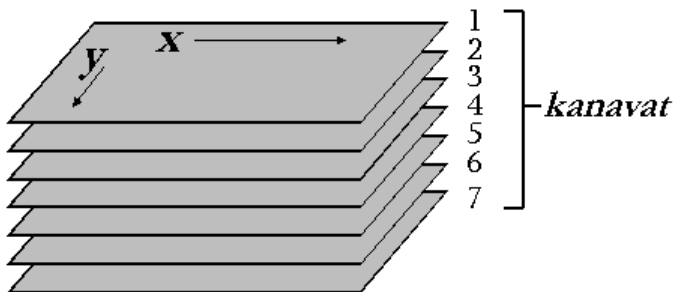
Kuva 8. Rasterikuva.

Pikselien määrälle asettavat rajan sekä mittauslaitteiden että tietokonelaitteistojen rajoitukset ja tyypillisessä rasterikuvassa pikseleitä on vaaka- ja pystysuunnassa noin tuhat eli yhteensä noin miljoona pikseliä. Pikseli sisältää kuvatyypistä riippuen erilaista tietoa: harmaatasokuvassa valon intensiteetin määrän, RGB-värikuvassa punaisen, vihreän ja sinisen valon intensiteettejä ja spektrikuvassa lukuisan määrän intensiteettiarvoja eri aallonpituuksilla.

Jos kuva sisältää vähäisen määrän (yleensä alle 256) värejä, käytetään usein ns. palettikuvia, jotka koostuvat joukosta värejä ja itse kuvasta, jonka pikselit sisältävät indeksin palettiin. Tällaiset kuvat vievät huomattavasti vähemmän tilaa kuin kuvat, joissa jokainen pikseli sisältää oman värin.

2.5.3. Spektrikuvat

Spektrikuva, jota kutsutaan myös monikanavakuvaksi tai monikomponenttikuvaksi, on useista aallonpituusalueista koostuva esitys, missä kukin alue on esitetty harmaatasokuvana. RGB-värikuva on yleisin käytetyistä monikanavakuvista. Se koostuu punaisesta, vihreästä ja sinisestä kuvasta, jotka on otettu eri aallonpituuksia mittaavalla sensorilla. Sensori mittaa vastaavien värien arvot. Joskus spektrikuvaksi määritellään vain kuvat, joissa on enemmän kuin kolme kanavaa. Useammasta kanavasta koostuvia spektrikuvia käytetään usein kaukokartoituksessa. Satelliittien ottamat kuvat sisältävät näkyvän valon taajuuksien lisäksi usein myös näkyvän valon ulkopuolella olevia kanavia, kuten infrapunataajuuksia. Esimerkiksi Landsat 5-järjestelmän ottamat spektrikuvat sisältävät 7 kanavaa (kuva 9) aallonpituuksien 450 nm ja 1250 nm välillä [2]. Ensimmäinen kanava vastaa siis aallonpituutta 450 nm ja seitsemäs 1250 nm.



Kuva 9. 7-kanavainen spektrikuva.

3. TIIVISTÄMINEN

Tiedon talletus tulee sitä halvemmaksi, mitä vähemmän talletettavaa dataa on. Samoin tiedonsiirto on sitä nopeampaa, mitä vähemmän siirrettävää dataa on. Tiedon pakkauksen, eli tiivistämisen, tarkoituksena on esittää data niin, että uusi pakattu tieto vie mahdollisimman vähän tilaa. Tästä pakatusta tiedosta täytyy tietenkin vielä pystyä palauttamaan alkuperäinen tieto.

Informaatioteorian /19/ yksi perustotuus on, että yleinen tiedon pakkaaminen on mahdotonta. Tämä tarkoittaa sitä, että mikään pakkausalgoritmi ei voi tiivistää kaikkea mahdollista dataa, vaan jokaista pakkautuvaa tiedostoa kohti on olemassa ainakin yksi tiedosto, jonka koko pakatessa vain kasvaa. Pakkausalgoritmien ideana onkin, että ne pakkaisivat tietyn tyyppistä dataa.

On olemassa algoritmeja, jotka tiivistävät hyvin melkein kaikkea dataa, näistä tunnetuin on Ziv-Lempel-algoritmi /22/. Yleisten algoritmien puute on, etteivät ne ota huomioon tietyn tyyppiselle datalle ominaisia piirteitä, kuten esimerkiksi kuvien kaksiulotteista rakennetta.

Tiivistämismenetelmät voidaan jakaa kahteen luokkaan: tietoa hukkaamattomiin ja tietoa hukkaaviin. Hukkaamattomalla algoritmilla pakattu tieto saadaan palautettua alkuperäiseksi, kun taas hukkaavalla algoritmilla pakatessa tieto vääristyy. Jos datan vääristyminen tiettyyn pisteeseen asti kuitenkin sallitaan, saavutetaan hukkaavilla algoritmeilla moninkertainen tiivistyssuhde hukkaamattomiin verrattuna. Varsinkin, jos data on tarkoitettu ihmisen tulkittavaksi, esimerkiksi kuva- tai äänimuodossa, pienestä vääristymästä ei yleensä ole haittaa.

3.1. Tiivistyksen ominaisuudet

Tiivistämisen onnistumisen toteamiseksi ja eri menetelmien vertailemiseksi tiivistyksen ominaisuuksia täytyy voida mitata. Eri sovelluksissa tiivistämiseltä halutaan erilaisia ominaisuuksia, esimerkiksi videoneuvotteluissa pakkaamisen ja purkamisen tulee olla ennen kaikkea nopeaa, kun taas tiedon varastoinnissa tärkeämpää on saada data tiivistettyä mahdollisimman pieneksi. Joissakin sovelluksissa (esim. kuvat, musiikki, elokuvat) data pakataan kerran, mutta puretaan useamman kerran, jolloin purkamisen viemä aika on tärkeämpää kuin tiivistykseen kulunut aika. Tiivistämisen määrään ja sen viemään aikaan vaikuttaa myös käytetyn algoritmin kompleksisuus ja sen laitteistovaatimukset. Yleisenä nyrkkisääntönä voidaan pitää, että nopeat algoritmit ovat myös yksinkertaisempia mutta tiivistyssuhteeltaan heikompia, kun taas tiivistyssuhteeltaan tehokkaammat algoritmit ovat hitaita ja vaativat laitteistolta enemmän muistia ja prosessointitehoa.

Tietoa tiivistetään hyödyntämällä sen sisältämää ylimääräistä tietoa, redundanssia. Redundanssi voidaan jakaa kahteen luokkaan: tilastolliseen ja subjektiiviseen redundanssiin. Tilastollista redundanssia voidaan hyödyntää käyttämällä hyväksi datan sisältämää samankaltaisuutta, korrelaatiota ja ennustettavuutta. Tilastollinen redundanssi on luonteeltaan hukkaamatonta ja data, josta tilastollinen redundanssi on poistettu, voidaan palauttaa takaisin alkuperäiseen muotoon. Subjektiivinen redundanssi on informaatiota, joka voidaan poistaa ihmisen huomaamatta, mutta dataa ei voida palauttaa alkuperäiseen muotoonsa /15/.

Shannonin informaatioteorian /19/ mukaan lähdedata koostuu symboleista, joiden informaation sisällön mitta biteissä on $-\log_2 p(x)$, missä $p(x)$ on symbolin x esiintymistodennäköisyys lähdedatassa. Eli jos $p(x) = 1$, symbolin x informaation sisältö on olematon, koska sen täytyi tapahtua. Toisaalta mitä pienempi todennäköisyys symbolilla x on, sitä suurempi on sen informaation sisältö. Lähdedatan symbolien keskimääräisen informaation sisällön mitta on entropia. Entropia H

lasketaan painottamalla lähdedatan kunkin symbolin informaation sisältö sen esiintymisen todennäköisyydellä

$$H = \sum_{i=1}^n [-p(a(i)) \log_2 p(a(i))]. \quad (3)$$

Koska symbolia vastaavan koodisanan $a(i)$ pituus täytyy olla tarpeeksi pitkä informaatiollista sisältöä varten, entropia rajaa pienimmän bittien määrän, jolla symboli voidaan koodata. Koodatun datan bittien kokonaismäärän tulee siis olla vähintään lähdedatan symbolien määrän ja entropian tulon pituinen. Redundanssi R voidaan mitata laskemalla keskimääräisen koodisanan pituuden ja entropian erotus

$$R = \sum_{i=1}^n [p(a(i))l(i)] - \sum_{i=1}^n [-p(a(i)) \log_2 p(a(i))], \quad (4)$$

jossa $l(i)$ on viestiä $a(i)$ vastaavan koodisanan pituus biteissä.

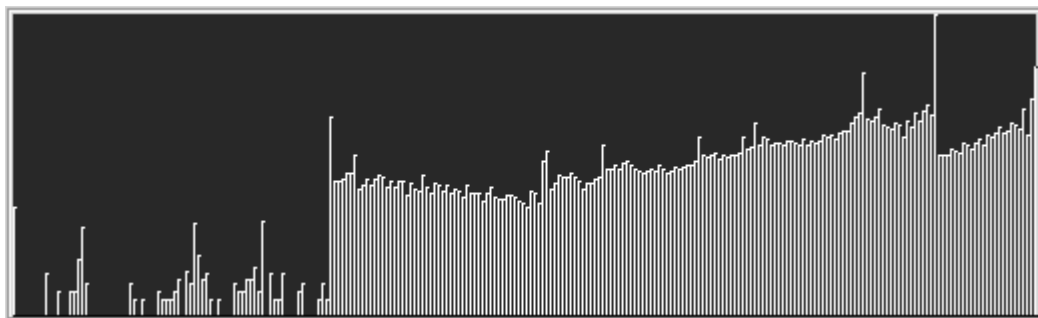
Käytetyn koodaustavan tuottamaa tiivistystä voidaan mitata myös tiivistyssuhteen avulla. Tiivistyssuhteella tarkoitetaan alkuperäisen lähdedatan ja tiivistetyn datan viemän tilan suhdetta. Se voidaan laskea joko edellä mainitulla tavalla, jolloin se ilmaisee kerrointa kuinka paljon dataa on tiivistetty, tai käänteisenä, jolloin saadaan luku, joka ilmaisee, kuinka pieneksi data on tiivistetty verrattuna alkuperäiseen. Erilaisia tiivistyksen mittoja on esitetty taulukossa 1, jossa esiintyvä symboli t tarkoittaa tiivistetyn datan ja l tiivistämättömän datan viemää tilaa.

Taulukko 1. Tiivistyksen mittoja.

Mitta	Kaava	Esimerkki
Tiivistyssuhde	t/l	2.74:1
Suhteellinen tiivistetty koko	l/t	0.36 tai 36%
Suhteellinen tiivistys	$(t-l)/t$	0.64 tai 64%

3.2. Erimittaiset koodisanat

Jotta tilastollista redundanssia voidaan hyödyntää, tulee datan tilastolliset piirteet olla tiedossa. Kuvassa 10 on eräästä harmaatasokuvasta muodostettu histogrammi, jossa x-akselilla on harmaatasot ja y-akselilla kunkin harmaatason esiintymisen määrä kuvassa.



Kuva 10. Harmaatasokuvan histogrammi.

Tiivistettäessä dataa, jossa on joitakin useasti esiintyviä koodisanoja ja useita harvemmin esiintyviä koodisanoja, yksi yleinen varsin yksinkertainen tiivistäminen menetelmä on erimittaisten koodisanojen käyttäminen. Siinä useasti esiintyvät koodisanat esitetään lyhyemmillä ja harvemmin esiintyvät koodisanat pidemmällä bittijonoilla.

Lennätinmellä sähkötetty Morse-koodi on ehkä yleisesti tunnetuin erimittaiten koodisanojen menetelmä. Siinä aakkoset, numerot ja välimerkit esitetään pisteiden (.) ja viivojen (-) yhdistelminä. Morse-koodin lyhyin merkki (.) vastaa englannin kielen useimmin esiintyvää kirjainta E ja pisin merkki (---) vastaa harvoin esiintyvää kirjainta Q. Morse-koodi ei sellaisenaan sovellu käytettäväksi tietotekniikassa, sillä Morse-koodissa kirjaimet erotetaan toisistaan sekaannusten välttämiseksi taukojen avulla.

Binäärikoodattujen erimittaiten koodisanojen erottamiseen toisistaan on useita menetelmiä, joista käytetyin on *prefix*-koodaus, jossa koodisanat ovat yksilöllisiä siten, ettei mikään jotakin tiettyä koodisanaa vastaava bittijono voi alkaa sellaisella jonolla, joka olisi jokin koodisana.

3.3. HUFFMAN-koodaus

Tietotekniikassa käytetyin erimittaiten koodisanojen menetelmä on Huffman-koodaus. Huffman-koodauksessa jokainen lähdesymboli pyritään koodaamaan sellaisella koodisanalla, jonka pituus bitteinä karkeasti vastaa kyseisen symbolin informaatiollista sisältöä. Tuloksena on koodijoukko, jonka keskimääräinen koodisanan pituus lähestyy lähteen entropiaa $1/n$.

Huffmanin algoritmi saa syötteekseen listan, joka sisältää kunkin symbolin ja sen todennäköisyyden tiivistettävässä datassa. Algoritmin mukaan listan todennäköisyyksistä rakennetaan binääripuu siten, että jatkuvasti yhdistetään kaksi pienintä todennäköisyyttä uudeksi puuksi kunnes jäljellä on yksi puu, jossa symbolit ja niiden todennäköisyydet ovat lehtinä. Kutakin symbolia vastaava koodisana saadaan, kun kuljetaan kyseistä lehdestä juurta kohti ja aina solmun vasemman lapsen kautta kuljettaessa koodiin lisätään 0 ja oikean lapsen kautta kuljettaessa 1. Vastaavasti dekkoodaus suoritetaan käänteisesti kulkemalla juuresta lehteä kohti.

Koodisanoista tulee siis erimittaisia, useasti esiintyvistä lyhyempiä ja harvoin esiintyvistä pidempiä kuin alkuperäiset koodisanat. Algoritmi tuottaa optimaalisen koodin kyseiselle datalle, toisin sanoen algoritmin tuottamien uusien koodisanojen avulla saavutetaan paras mahdollinen tiivistyssuhde, mikä erimittaisten koodisanojen avulla on mahdollista. Koska koodisanat koostuvat kokonaisista biteistä, Huffman-koodauksella ei saavuteta optimaalista entropiaa paitsi, jos jokaisen koodisanan esiintymisen todennäköisyys on 2^{-n} , missä n on positiivinen kokonaisluku. Keskimääräinen koodisanan pituus \bar{L} on Huffman-koodatulla datajoukolla B välillä $H(B) \leq \bar{L} \leq H(B) + 1$ bittiä/koodisana, missä H on entropia.

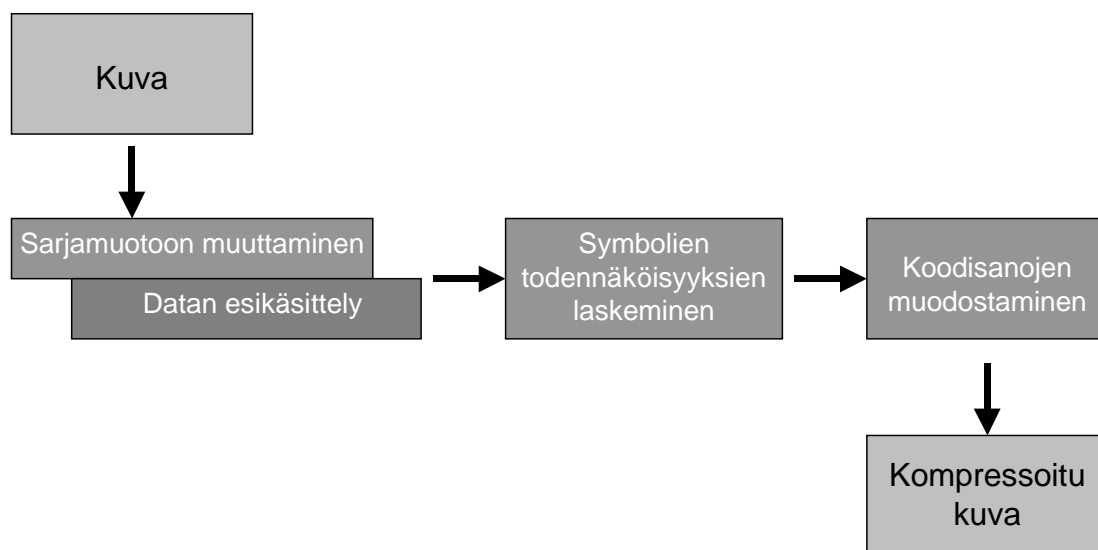
Huffman-koodaus ei sovellu kaikkiin tarkoituksiin kuten esimerkiksi tiedonsiirtoon, koska Huffman-koodausta varten koko koodattava aineisto on oltava tiedossa ennen tiivistämisen alkamista. Pelkkä Huffman-koodaus ei ole käytetty tiivistysmenetelmä, koska se ei ota huomioon datan rakennetta ja sisäistä riippuvuutta, mutta sitä käytetään osana monia tiivistysmenetelmiä kuten esim. JPEG ja ZIP.

3.4. Kuvien tiivistäminen yleisesti

Tiivistysmenetelmän valintaan vaikuttaa tiivistettävän datan luonne. Rasterikuvilla on joitakin tiivistämisessä huomion arvoisia erityispiirteitä kuten kaksiulotteisuus ja kiinteä koko. Kuvan voisi ajatella olevan kaksiulotteinen taulukko, joka koostuu väreistä. Väri taas on harmaatasokuvan tapauksessa yksi luku, joka ilmaisee valoisuuden määrää, RGB-kuvassa kolme lukua, jotka ilmaisevat punaisen, vihreän ja sinisen määrää ja spektrikuvassa usean valon eri aallonpituuden voimakkuutta ilmaisevan luvun joukko. Kuvan kiinteä koko vaikuttaa tiivistykseen siten että sen statistiikka, varsinkin symbolien todennäköisyydet, on tiedossa ennen tiivistystä.

Kuvat eivät sisällä satunnaista kohinaa vaan ovat yleensä mittaustietoa ympäröivästä maailmasta kuten valokuvat, tai ihmisen luomia kuvioita. Ne sisältävät yhtenäisiä värialueita ja muotoja, joita voidaan käyttää hyväksi tiivistyssuhteen parantamiseksi.

Useimmat hukkaamattomat kuvien tiivistysmenetelmät koostuvat neljästä eri vaiheesta (kuva 11). Nämä vaiheet eivät ole peräkkäisesti suoritettavia, vaan osa niistä suoritetaan samanaikaisesti, limittäin. Ensin kuva muutetaan kaksiulotteisesta sarjamuotoon, eli yksiulotteiseksi. Toiseksi kuva esikäsitellään ja yritetään vähentää datan kompleksisuutta. Kolmanneksi jokaiselle esikäsitellylle datan symbolille lasketaan esiintymisen todennäköisyys. Neljänneksi esikäsitellyn datan ja sen todennäköisyyden perusteella muodostetaan koodisana, joiden avulla saadaan tiivistetty kuva /1, 5, 13, 21/.

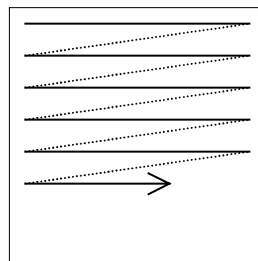


Kuva 11. Nelivaiheinen yleinen kuvien tiivistysmalli.

3.4.1. Sarjamuotoon muuntaminen

Kuvan muuntaminen kaksiulotteisesta datasta yksiulotteiseksi ja datan kompleksisuuden vähentäminen eivät ole riippumattomia toisistaan. Kuvassa lähekkäisten pisteiden korrelaation hyödyntämiseksi tulee tietää myös tapa, jolla kuva muutetaan yksiulotteiseksi. Sarjamuotoon muuntamisen tulee siis parhaalla mahdollisella tavalla hyödyntää lähekkäisten pisteiden korrelaatiota. Sarjamuotoon muuntaminen voidaan tehdä usealla tavalla ja tavoitteena on, että seuraavan sarjamuotoon muutettavan pisteen eli pikselin alkuperäisistä naapureista mahdollisimman moni olisi jo käsitelty.

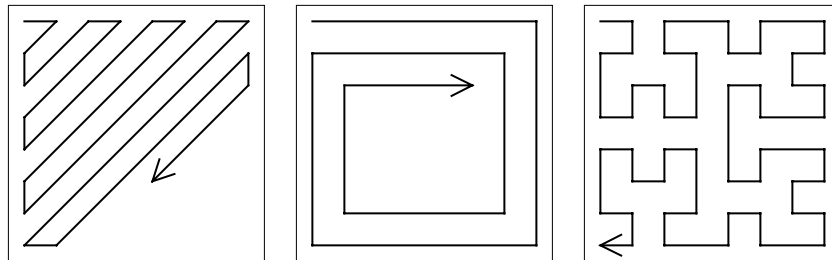
Yleisin ja yksinkertaisin tapa käydä läpi kaksiulotteinen kuva on rivi kerrallaan vasemmalta oikealle (kuva 12). Tällaisessa tavassa käydä läpi kaksiulotteinen kuva paitsi ylimmässä rivissä ja vasemmalla ja oikealla laidassa olevien pisteiden naapureista on suurin osa on tiedossa. Muutoin pisteen naapurustosta on tiedossa paitsi ylempänä ja vasemmalla oleva piste niin myös oikealla ja vasemmalla yläviistossa olevat pisteet, eli puolet naapuripisteistä.



Kuva 12. Rasteri -skannaus.

Melkein vastaava tapa käydä läpi kuva on aloittaa vasemmasta yläkulmasta ja siirtyä siksakmaisesti alaviistoon kohti oikeaa alakulmaa (kuva 13). Kuten edellisessäkin tavassa, lukuun ottamatta kuvan reunoja, naapurustosta on tiedossa kaksi lähinnä olevaa pistettä ja kaksi viistossa olevaa pistettä. Kuva voidaan käydä läpi myös

reunoja pitkin, spiraalimaisesti (kuva 13), jolloin keskipisteen suhteen symmetriaa omaavien kuvien tapauksessa tapa olisi parempi kuin edelliset.



Kuva 13. Siksak-, ympyrä- ja peano-skannaus.

Peano-käyrän (kuva 13) avulla tapahtuvan skannauksen hyvä puoli on, että edelliset läpikäytyt pikselit ovat lähempänä kuin muissa, lineaarisemmissä, tavoissa. Myöskään kaukaisempia jo läpikäytyjä pikseleitä ei tarvitse säilyttää muistissa. Menetelmän haittapuolena on, että jo läpikäytyjen naapuripikseleiden määrä vaihtelee ja on yleensäkin pienempi kuin muissa menetelmissä. Menetelmä ei myöskään sovellu kuin kuville, joiden pikselien määrä sivu- ja pystysuunnassa on käyrän leveyden monikerta.

3.4.2. Sisällön vähentäminen

Samalla kun kuvaa muutetaan sarjamuotoon, pyritään datan kompleksisuutta vähentämään hyödyntämällä lähekkäisten kuvapisteiden välistä korrelaatiota. Sarjamuotoon muuttaminen ja kompleksisuuden vähentäminen ovat rinnakkaisesti suoritettavat operaatiot ja myös toisistaan riippuvaiset. Jotta korrelaatio voidaan hyödyntää parhaalla mahdollisella tavalla, täytyy sarjamuotoon muuttaminen ja sisällön vähentäminen toteuttaa yhdessä.

Korrelaatio laskee pisteiden etäisyyden kasvaessa toisistaan ja yleensä muita kuin viereisiä pisteitä ei kannatakaan käyttää apuna. Yksinkertaisin tapa hyödyntää kuvien korrelaatiota on deltakoodaus /15/.

3.4.3. Deltakoodaus

Deltakoodauksessa /15/ peräkkäisiä arvoja ei koodata sellaisenaan vaan koodataan arvon erotus edelliseen arvoon. Vierekkäisten pikselien korrelaation johdosta peräkkäisten pikselien arvojen erotus on yleensä pieni. Mittauksiin perustuvissa kuvissa peräkkäiset mittaukset ovat riippuvaisia toisistaan eikä suuria erotuksia esiinny. Deltakoodatuilla arvoilla on kaksi ominaisuutta verrattuna alkuperäisiin arvoihin: ensinnäkin niiden arvoalue on suppeampi ja toiseksi pienet arvot ovat paljon todennäköisempiä kuin suuret arvot.

Suppeamman arvoalueen johdosta arvot voidaan koodata vähemmällä määrällä bittejä, eli sen entropia pikseliä kohti on pienempi, jolloin kuva saadaan tiivistettyä pienempään tilaan menettämättä informaatiota. Toisaalta koska pienet arvot ovat huomattavasti todennäköisempiä kuin suuret arvot, erimittaisten koodisanojen menetelmällä, esim. Huffman-koodauksella, voidaan kuvaa pakata vieläkin pienemmäksi.

3.4.4. Ennakoiva koodaus

Redundanssia voidaan koettaa vähentää myös hyödyntämällä pisteiden välistä korrelaatiota siten, että yritetään ennakoida jo tiedossa olevien pikselien avulla seuraava pikseli. Arvoksi koodataan sitten alkuperäisen pikselin ja ennakoidun pikselin erotus. Käytettäessä ennakoinnissa vain edellistä pikseliä antaa menetelmä samanlaisen lopputuloksen kuin deltakoodauskin, joskin hieman eri lailla toteutettuna. Käyttämällä useampia edellisiä pikseleitä saadaan pikselin arvo

ennakoitua vielä paremmin. Ongelmana on kuitenkin, että mitä kauempana pikselit ovat toisistaan, sitä vähemmän korrelaatiota niiden välillä esiintyy. Siksi edelliset pikselit täytyykin painottaa jollakin sopivalla, etäisyyden huomioon ottavalla menetelmällä.

Luotaessa ennakointifunktioita, ennustajia, kuvia tiivistettäessä on hyvä huomioida niiden kaksiulotteinen luonne ja pikseleiden välinen korrelaatio molemmissa ulottuvuuksissa. Tällöin käytössä on yleensä kaksi yhden pikselin etäisyydellä olevaa naapuripikseliä ja kaksi $\sqrt{2}$ pikselin etäisyydellä olevaa naapuripikseliä. Ennakointifunktion ei tietenkään tarvitse olla kiinteä, vaan niitä voi olla useita joista jollakin perusteella valitaan sopivin. Esimerkiksi hukkaamattomassa JPEG-standardissa on 8 ennustajaa, joista valita (taulukko 2) /1/.

Taulukko 2. hukkaamattoman JPEG-standardin lineaariset ennustajat.

Moodi	Ennakoitu arvo, pikseli $x(i,j)$
0	0 (Ei ennakointia)
1	$x(i-1,j)$
2	$x(i,j-1)$
3	$x(i-1,j-1)$
4	$x(i,j-1) + x(i-1,j) - x(i-1,j-1)$
5	$x(i,j-1) + (x(i-1,j) - x(i-1,j-1)) / 2$
6	$x(i-1,j) + (x(i,j-1) - x(i-1,j-1)) / 2$
7	$(x(i,j-1) + x(i-1,j)) / 2$

Kiinteät lineaariset ennustajat, kuten hukkaamattomassa JPEG-standardissa käytetyt, ovat yksinkertaisimpia käytettyjä ennustajia. Niiden ongelmana on huono toimivuus alueilla, jotka sisältävät reunoja. Adaptiiviset, epälineaariset ennustajat osaavat muuttaa parametrejaan sen mukaan toimitaanko reunoja sisältävällä alueella /21/.

3.4.5. Todennäköisyyksien laskeminen ja koodisanan sijoitus

Edellisestä vaiheesta saaduille arvoille lasketaan niiden esiintymistodennäköisyydet, joiden perusteella kullekin arvolle muodostetaan uusi koodisana. Suurimmalle osalle kuvia ennakointivirheiden todennäköisyyksien jakauma on hyvin lähellä Laplacen jakaumaa, jossa jakauman keskelle muodostuu piikkimäinen kärki. Tällaisille jakaumille koodisanat muodostetaan yleensä käyttämällä jotakin erimittaiset koodisanat generoivaa algoritmia, joista käytetyimmät ovat Huffman-koodaus ja aritmeettinen koodaus /1/.

3.5. Hukkaava tiivistäminen

Hukkaamattomilla kuvien tiivistämismenetelmillä saavutetaan yleensä tiivistyssuhde väliltä 2:1 - 4:1 riippuen tiivistetystä kuvasta. Huomattavasti suurempi tiivistyssuhde, jopa 200:1, saavutetaan, jos sallitaan pikselien arvojen muutokset tiivistämisen aikana /4/. Hyväksyttävän vääristymisen määrittämiseen käytetään kahta menetelmää, laskennallista eroa alkuperäiseen kuvaan ja tiivistetyn kuvan silmämääräistä arviointia. Edellistä käytetään, kun kuvaa on tarkoitus analysoida laskennallisesti tietokoneella. Jälkimmäistä käytetään, koska mitkään laskennalliset suureet eivät kerro koko totuutta.

Silmämääräisesti arvioituna hukkaavasti tiivistetty kuva voi näyttää paremmalta kuin sama kuva tiivistettynä jollakin toisella hukkavalla menetelmällä, vaikka laskennallinen virhe olisi jälkimmäisessä pienempi. Yleensä myös tiivistyssuhde on jälkimmäisellä menetelmällä huonompi. Tiivistysalgoritmit voivat hyödyntää ihmisen silmän ja näkökyvyn ominaisuuksia monella tavalla. Ihmiset havaitsevat paremmin valon intensiteetin kuin värin sävyn eroja, joten usein kuvan intensiteetti- ja väri-informaatio erotetaan toisistaan ja intensiteetti talletetaan suuremmalla tarkkuudella kuin pikselin värin sävy. Ihmisen näkökyvyn logaritmisesta havainnoinnista johtuen erotamme helpommin pienet erot kuvan tummilla alueilla kuin vastaavan kokoiset

erot intensiteetissä kuva kirkkailla alueilla. Emme myöskään havaitse kuvan sisältämien alueiden tai kappaleiden reunoissa tapahtuvia suuriakin virheitä pikselien intensiteetissä tai värin vuotamista reunan yli /17/. Käytetyin menetelmä, joka hyödyntää ihmisen näkökyvyn ominaisuuksia, on JPEG-standardin hukkaava menetelmä.

Spektrikuvien tiivistämisessä ei ole hyvä käyttää menetelmiä, joissa saavutettu tiivistyssuhde perustuu ihmisen silmän ominaisuuksiin, vaan tulisi käyttää mieluummin menetelmää, joka pienentää laskennallista virhettä. Tämä johtuu siitä että spektrikuvia yleensä analysoidaan laskennallisesti.

3.5.1. Kvantisointi

Kvantisoinnissa jatkuva, yleensä kuitenkin diskreetti, arvoalue kuvataan arvoalueen karkeammalle osajoukolle, jolloin tietoa hukkuu. Esimerkiksi kuvaamalla arvot 0..255 osajoukolle {16, 48, 80, 112, 144, 176, 208, 240} saavutetaan tiivistyssuhde 3/8 virheen ollessa pahimmassa tapauksessa 16 yksikköä alkuperäisestä.

Koska lähdeaineisto ei yleensä ole tasaisesti jakautunut, ei lineaarinen kvantisointi myöskään useimmiten ole paras mahdollinen. Varsinkin, jos virheen prosentuaalinen osuus alkuarvoon verrattuna halutaan pitää tasaisesti jakautuneena, on tärkeämpää saada pienet arvot lähemmäs alkuperäisiä kuin suuremmat. Edellisen esimerkin arvoalueella epälineaaraisesti kvantisoitu joukko voisi olla {1, 2, 4, 8, 16, 32, 64, 128}. Epälineaarista kvantisointia käytetään esimerkiksi kun koodataan peräkkäisten kuvapikseleiden erotusta (esim. differentiaalinen pulssikoodimodulaatio /15/). Ihmisen näkökyky on herkkä havaitsemaan pieniä virheitä kahden samankaltaisen kuvapisteen välillä mutta huono havaitsemaan virheitä kahden erilaisen kuvapisteen välillä. Tästä johtuen reuna-alueilla olevat kuvapisteen välillä olevat virheet voivat olla virheeltään suurempia kuin tasaisemmillä kuvan alueilla /15/.

Edellä kuvattua kvantisointia kutsutaan myös skalaarikvantisoinniksi, sillä siinä yksi syöte kvantisoidaan yhdeksi arvoksi.

3.5.2. Vektorikvantisointi

Vektorikvantisoinnissa joukko alkuarvoja, vektori, kuvataan yhdellä kerralla yhdeksi luvuksi, indeksiksi, joka vastaa kyseistä osajoukkoa. Indeksien määrä on yleensä huomattavasti pienempi kuin mahdollisten osajoukkojen määrä, jolloin samaan indeksiin kuvautuu useita, samankaltaisia osajoukkoja. Vektorikvantisoinnin avulla voidaan tiivistää tietoa vielä tehokkaammin kuin skalaarikvantisoinnilla, koska datajoukon alkiota, kuten esimerkiksi 25-komponenttista pikseliä, voidaan kuvata yhdellä luvulla, indeksillä /15/.

Tiivistettäessä kuvia vektorikvantisoinnin avulla vektorit muodostetaan usein esim. 8x8-pikselin lohkoista. Jos pikselin sisältämä väri muodostuu useammasta komponentista, lohko on yleensä pienempi. Jotta kuvan lohkot voidaan kuvata indekseiksi, tulee käytössä olla kuvalle sopiva 'koodikirja', joka sisältää joukon sellaisia lohkoja, joille kuvautuvien kuvan lohkojen virhe on mahdollisimman pieni. Koodikirja voidaan luoda erikseen kutakin kuvaa varten tai luoda yleinen koodikirja, jota käytetään joukolla samankaltaisia kuvia. Koodikirjaa luotaessa voidaan käyttää hyväksi tiivistettävän kuvan kaikkia lohkoja, osaa lohkoista tai satunnaisia lohkoja useista kuvista. Tiivistäminen tapahtuu etsimällä tiivistettävän kuvan kullekin lohkolle koodikirjasta virheeltään vähäisin, eli sitä lähinnä oleva lohko. Tiivistetty kuva muodostuu siten koodikirjasta ja indeksikuvasta siihen. Kuvan palauttaminen suoritetaan sijoittamalla indeksikuvaan indeksiä vastaava lohko koodikirjasta /4/.

Neurolaskennalla tehtävässä vektorikvantisoinnissa itseorganisoidaan neuroverkko siten, että sovelluksessa esiintyviä datavektoreita edustetaan mahdollisimman hyvin pienemmällä joukolla neuroneita. Kullekin datavektorille etsitään sitä parhaiten

vastaava neuroni, jolloin siirrettävänä tietona voidaan käyttää yhtä lukua, neuronin indeksii, monen luvun mittaisen vektorin sijaan. Vastaanottaja voi indeksin saatuaan palauttaa sen takaisin vektoriksi, jota vastaanotettu neuroni edustaa.

Tiivistäminen vektorikvantisoinnin avulla on monimutkaista ja aikaa vievää, kun taas kuvan palauttaminen on yksinkertaista ja nopeaa. Lisäksi muistin tarve on vähäinen ja vektorikvantisaatio, sekä tiivistäminen ja palauttaminen, hyvin skaalautuvaa.

4. ITSEORGANISOITUVA KARTTA

Itseorganisoituva kartta on yksi suosituimmista neuraalilaskentamenetelmistä /10/. Se kuuluu kilpailuoppiviin verkkoihin, eli opetukseen ei tarvitse puuttua eikä syötejoukon ominaisuuksia tarvitse tietää etukäteen. Sen avulla voidaan kuvata n-ulotteisen datajoukon näytteet alempiulotteiseen avaruuteen niin, että alunperin lähellä toisiaan olevat näytteet ovat lähellä toisiaan myös kuvauksessa, eli kartalla.

4.1. *Itseoppivat neuroverkot*

Oppivilla neuroverkoilla pystytään tehokkaasti ryhmittelemään suuria tietomääriä ja löytämään mittauksissa ja isoissa tietomassoissa olevia yhteyksiä tai ryhmiä, joita on hankala löytää tilastolaskennalla, matemaattisilla malleilla tai loogisilla säännöillä. Neurolaskennan algoritmit voidaan jakaa kolmeen luokkaan, jotka ovat periaatteiltaan erilaisia: kiinteäpainoiset verkot, ohjatun oppimisen verkot ja itseoppivat verkot /11/.

Itseoppimisessa tunnetaan syöteinformaatio, muttei vastetta. Tällöin oppimisen tehtävänä on löytää annetuista syötteistä lainalaisuuksia, joita voidaan hyödyntää informaation tulkinnassa tai muiden neurolaskennan systeemien syötteinä.

Tilastotieteestä tuttuja menetelmiä ovat ryhmittely ja pääkomponenttianalyysi, joille on olemassa vastineet neurolaskennassa. Ohjaamattoman opetuksen periaatetta voisi havainnollistaa siten, että syöteinformaation käyttäytymisestä muodostetaan malli neurolaskennalla. Opetusalgoritmi, eli mallin tekeminen, on usein toteutettu siten, että siinä neuronit kilpailevat keskenään siitä, mikä edustaa kutakin syötettä parhaimmin. Tämän vuoksi menetelmistä käytetään usein termiä kilpailuoppiminen.

Opetusta ohjaavana kriteerinä on ajatus, että neurolaskennan esitys ei hävitä tietoa. Toisin sanoen syöte, joka esitetään neuroverkon avulla, voidaan palauttaa

mahdollisimman lähelle alkuperäistä. Samalla tämä malli on huomattavasti yksinkertaisempi ja helpommin tulkittava kuin opetusaineisto /11/.

4.2. Kohosen itseorganisoituva kartta

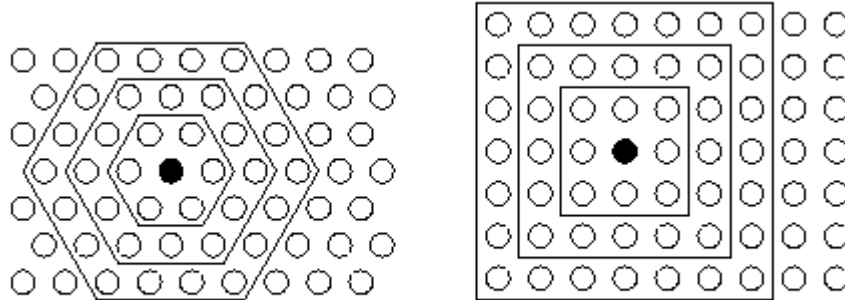
Tunnetuin ohjaamattoman opetuksen menetelmistä on professori Teuvo Kohosen kehittämä itseorganisoituva kartta /10/. Siinä havaintoaineisto esitetään joukolla neuroneita, joille on määrätty topologinen riippuvuus toisistaan. Riippuvuudeksi valitaan yleensä kaksiulotteinen pinta, jolloin neuroverkkoa voidaan käyttää monen yhtäaikaisesti vaikuttavan muuttujan esittämiseksi ihmiselle helppolukuisena kuvana.

Itseorganisoituvalla kartalla on kaksi tärkeää ominaisuutta: vektorikvantisointi ja jatkuva topologinen kuvaus kahden avaruuden välillä. Tämä tarkoittaa, että neuronit pyrkivät muuttamaan painokertoimiaan kohti tilaa, jossa jokainen neuroni on keskimääräisesti yhtä usein aktiivinen ja aktiivisuudet ovat jakautuneet mahdollisimman tasaisesti erilaisten syötteiden joukkoon. Lisäksi neuronit, jotka ovat verkossa lähellä toisiaan, reagoivat samantyyppisiin syötteisiin, kun taas verkossa kauempana olevat neuronit reagoivat erityyppisiin syötteisiin. Topologista itseorganisoituvaa karttaa voidaan soveltaa moniin tehtäviin, joissa esimerkiksi datan ryhmittely ja korrelaatioiden etsiminen ovat tärkeitä.

4.2.1. Itseorganisoituvan kartan rakenne

Itseorganisoituvan kartan muodostavat säännöllisellä, yleensä 1- tai 2-ulotteisella ruudukolla sijaitsevat neuronit. Jokainen neuroni koostuu n -ulotteisesta painovektorista, missä n on yhtäsuuri kuin syötevektorin ulottuvuus. Kartan neuronit on yhdistetty vierekkäisiin neuroneihin naapuruussuhteella, joka määrää kartan rakenteen. Kuvassa 14 on heksagonaalinen ja suorakaiteen muotoinen kartta-ruudukko. Kuvaan on myös rajattu mustalla merkityn neuronin naapurit etäisyyksien

mukaan siten, että sisimmät naapurit ovat etäisyydellä 1, seuraavat naapurit etäisyydellä 2 ja uloimmat etäisyydellä 3 /8/.

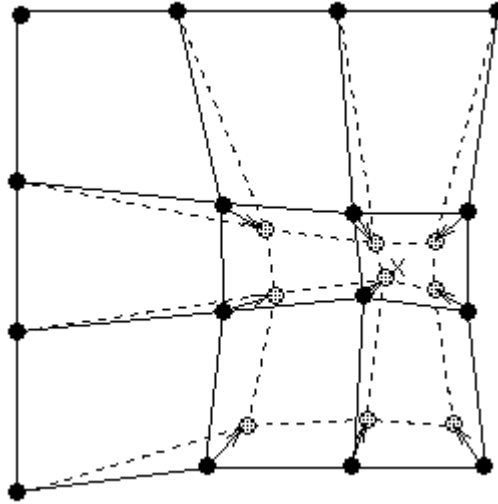


Kuva 14. Heksagonaalien ja suorakaiteen muotoiset karttaruudukot. /8/

4.2.2. Kartan alustus ja opetus

Kartan neuronien painovektorit, neuronien määrä ja topologiset suhteet on alustettava ennen varsinaista opetusta. Painovektorien alustus vaikuttaa siihen, kuinka nopeasti kartta konvergoituu hyväksyttävään ratkaisuun. Yleensä painovektorit alustetaan joko pienillä satunnaisluvuilla, satunnaisilla syötedatan arvoilla, tai lineaarisesti kattamaan koko syötejoukon arvoalueen.

Jokaisessa varsinaisen opetuksen askeleessa syötteestä valitaan satunnainen syötevektori, jolle lasketaan kartan neuroneista euklidiselta etäisyydeltään lähin neuroni BMU (Best Matching Unit). Tämän jälkeen lähimmän neuronin ja sen naapuruston painovektoreita siirretään lähemmäksi syötevektoria (kuva 15) /8/.



Kuva 15. Neuronien siirtyminen lähemmäs syötevektoria X . /8/

Algoritmissa 1 on esitetty varsinainen opetusalgoritmi. Algoritmissa mainittu etäisyys $d(-,-)$ kahden vektorin välillä on euklidinen etäisyys $\|x - w_i\|^2$, jota opetusvaiheessa pienennetään säännöllä

$$w_i(t+1) = w_i(t) + h_{BMU_i}(t)[x(t) - w_i(t)],$$

missä t on aika ja $h_{BMU_i}(t)$ on naapurisuusfunktio. Yksinkertaisimmillaan naapurisuusfunktio $h_{BMU_i}(t) = \alpha(t)$, jos $i \in N_{BMU}(t)$ ja $h_{BMU_i}(t) = 0$, jos $i \notin N_{BMU}(t)$, joissa $\alpha(t)$ on opetuksen muutoskerroin ($0 < \alpha(t) < 1$) ja $N_{BMU}(t)$ on BMU:n naapurineuronit. Sekä $\alpha(t)$ että $N_{BMU}(t)$ pienenevät yleensä monotonisesti ajan mukaan opetuksen edistyessä.

Algoritmi 1. Itseorganisoituvan kartan opetusalgoritmi. /10/

1. Ota vastaan opetusvektori x .
2. Valitse verkosta opetusta parhaiten vastaava neuronin BMU siten, että
$$\|x - w_{BMU}\| = \min \|x - w_i\|.$$
3. Muuta BMU:n ja sen naapurien painoja $w_i(t+1) = w_i(t) + h_{BMU_i}(t)[x(t) - w_i(t)]$, jolloin BMU:n asema paranee $d(w_i(t+1), x) < d(w_i(t), x)$, missä i on BMU tai sen naapuri ja $w(t)$ on vanha arvo sekä $w(t+1)$ on uusi arvo.

Naapurineuronien painovektoreita siirrettäessä voidaan huomioida myös niiden etäisyys syötevektoria lähimmästä neuronista siten, että kaukaisempia naapureita siirretään vähemmän kuin lähempiä naapureita. Opetus lopetetaan, kun joko ennalta asetettu määrä askelia on suoritettu, tai muuten havaitaan kartan konvergoituneen hyväksyttävään ratkaisuun.

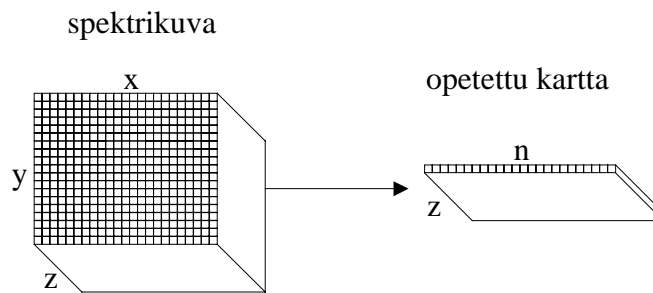
5. KEHITETTY TIIVISTÄMISMENETELMÄ

Tietokonelaitteistojen, varsinkin prosessorien, viimeaikainen kehitys on mahdollistanut yhä suurempien ja enemmän informaatiota sisältävien spektrikuvien käsittelyn. Kuitenkin spektrikuvien varastoiminen ja siirtäminen on ongelmallista niiden viemän tilan vuoksi. Esimerkiksi 7-kanavainen Landsat V -spektrikuva saattaa olla kooltaan yli 200 megatavua. Spektrikuva voidaan tietysti pakata jollakin yleisellä tiivistysalgoritmilla, mutta parempi ratkaisu olisi spektrikuvan ominaisuudet huomioon ottava algoritmi, jolloin päästäisiin parempaan tiivistyssuhteeseen. Spektrikuvien tiivistämiseen on olemassa joitakin menetelmiä, mutta ei yleisesti käytettyä ratkaisua.

Lappeenrannan teknillisen korkeakoulun Tietojenkäsittelytekniikan tutkimuslaboratoriossa kehitetyssä menetelmässä käytetään Kohosen itseorganisoituvaa karttaa spektrikuvien tiivistämiseksi /20/. Spektrikuvat esittävät useimmiten todellisia, luonnossa esiintyviä kohteita kuten esimerkiksi ilmakuvia maanpinnasta. Tällaiset kuvat koostuvat yleensä samankaltaisista väreistä ja alueista, joissa kuvan pikselit eroavat toisistaan vain vähän tai ei lainkaan. Spektrikuvan pikseli koostuu joukosta intensiteettiarvoja eri aallonpituuksilta. Toistuvaa dataa saadaan vähennettyä tiivistämällä spektrikuva. Tiivistäminen suoritetaan vektorikvantisoimalla spektrikuva itseorganisoituvan kartan avulla. Vektorikvantisoinnin avulla kuvataan alkuperäisen spektrikuvan pikselit siten, että jokaista alkuperäisen spektrikuvan pikseliä vastaa jokin itseorganisoituvan kartan avulla saatu joukko intensiteettiarvoja, eli värin spektri.

5.1. Spektrikuvan tiivistäminen

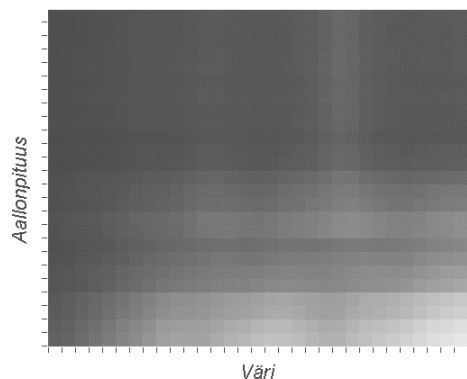
Itseorganisoitua kartta saa syötteekseen spektrikuvan, josta se muodostaa halutun määrän ryhmiä, eli spektrejä (kuva 16). Kuvassa x ja y määräävät kunkin pikselin paikan kuvassa, z on pikselin sisältämien intensiteettien määrä ja n opetetun kartan spektrien määrä.



Kuva 16. Pikseleiden ryhmittely itseorganisoituvan kartan avulla.

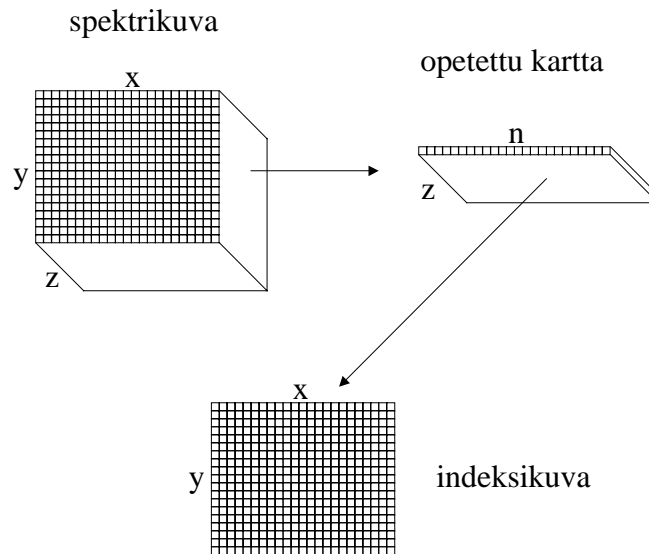
Itseorganisoitua kartta ei ryhmittele pikseleitä yhtäsuuriin ryhmiin. Ryhmässä, johon kuvautuu eniten pikseleitä on pikselin keskimääräinen virhe pienin. Vastaavasti ryhmässä, jossa on vähän pikseleitä, keskimääräinen virhe on suurempi.

Alkuperäisestä spektrikuvasta voidaan itseorganisoituvan kartan avulla saatujen spektrien, eli värikartan (kuva 17), avulla muodostaa indeksikuva.



Kuva 17. Spektrikuvalla opetettu 32 spektrinen kartta.

Indeksikuvassa alkuperäisen spektrikuvan pikseliä vastaa indeksi euklidisen etäisyyden mielessä lähimpään, eli virheeltään pienimpään, spektriin (kuva 18). Algoritmissa 2 on esitetty tiivistysalgoritmi yksinkertaistettuna.



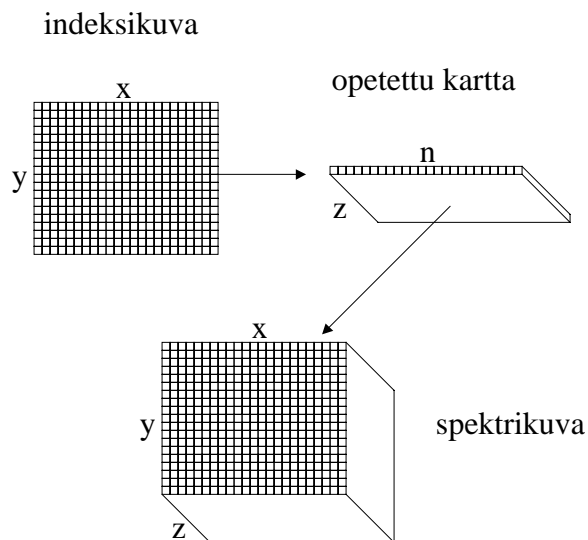
Kuva 18. Indeksikuvan muodostaminen spektrikuvan ja opetetun kartan avulla.

Algoritmi 2. Tiivistäminen itseorganisoituvan kartan avulla.

1. Itseorganisoituva kartta ryhmittelee spektrikuvasta halutun kokoisen kartan, joka sisältää spektrejä.
2. Muodostetaan spektrikuvan kokoinen indeksikuva.
3. Käydään spektrikuvan pikselit läpi yksitellen ja sijoitetaan indeksikuvaan vastaavaan kohtaan indeksi siihen kartan spektriin, joka on euklidisesti lähinnä spektrikuvan pikseliä.

5.2. Spektrikuvan palauttaminen

Tiivistetty spektrikuva koostuu siis opetetusta kartasta ja indeksikuvasta. Näiden avulla saadaan palautettua alkuperäisestä hieman vääristynyt spektrikuva (kuva 19). Virheen määrään voidaan vaikuttaa käytetyn kartan koolla. Isolla kartalla virhe ja tiivistyssuhdekin pienenee ja vastaavasti pienellä kartalla saavutetaan suurempi tiivistyssuhde, mutta virhekin kasvaa. Tiivistetyn spektrikuvan palauttaminen on esitetty algoritmissa 3.



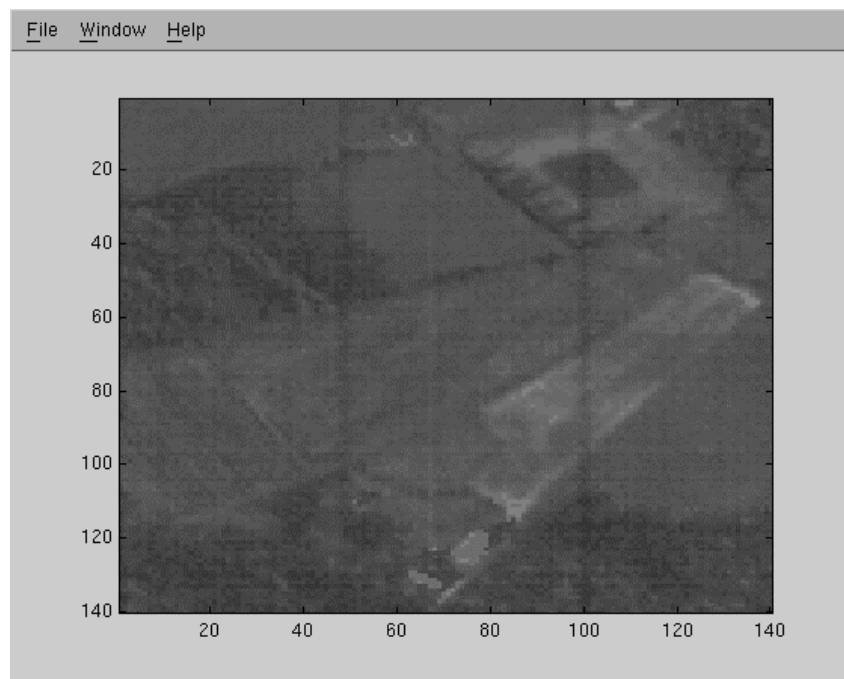
Kuva 19. Spektrikuvan palauttaminen.

Algoritmi 3. Tiivistetyn spektrikuvan palauttaminen.

1. Muodostetaan indeksikuvan kokoinen spektrikuva.
2. Käydään indeksikuvan indeksit läpi yksitellen ja sijoitetaan spektrikuvaan vastaavaan kohtaan indeksin osoittama spektri kartassa.

6. MENETELMÄN SOVELTAMINEN

Menetelmää kokeiltiin 25-kanavaisilla ilmakuvilla. Tässä mainitut tulokset saatiin kuvalla 20, jonka koko oli 140x140 pikseliä. Kuvan kanavat olivat tasavälein alkaen 649.9 nanometrillä ja loppuen 747.4 nanometriin ja jokaisen kanavan sisältämät arvot olivat välillä 0-255. Spektrikuvan alkuperäiseksi kooksi tuli siis $140 \times 140 \times 25 = 490000$ tavua. Kartta opetettiin maplearn-ohjelmalla, joka käyttää Luttrellin algoritmia [12]. Karttoja opetettiin useita eri kokoisia siten, että opetetun kartan spektrien määrä oli välillä 16-2048 (2^4 - 2^{11}).



Kuva 20. 25-kanavainen ilmakuva.

Spektrikuva tiivistettiin eri kokoisilla kartoilla ja laskettiin kussakin tapauksessa saavutettu tiivistyssuhde (taulukko 3). Tiivistämisen jälkeen spektrikuvat palautettiin ja laskettiin prosessin aikana syntynyt virhe. Virhe laskettiin mittaamalla euklidinen etäisyys alkuperäisen spektrikuvan pikseleiden ja palautetun spektrikuvan pikseleiden

välillä ja ottamalla niiden keskiarvo kanavaa kohti (taulukko 3). Palautetusta spektrikuvasta etsittiin myös suurin pikselin yksittäisellä kanavalla tapahtunut virhe.

Taulukko 3. Algoritmin 2 testitulokset kuvalla 20.

SOM koko	2048	1024	512	256	128	64	32	16
Suurin virhe	23	30	33	57	58	59	63	63
Keskiarvovirhe	2.19	2.37	2.53	2.66	2.78	2.92	3.13	3.49
SOM:n koko tavuina	51200	25600	12800	6400	3200	1600	800	400
Indeksikartan koko tavuina	26950	24500	22050	19600	17150	14700	12250	9800
Tiivistyssuhde	6.27	9.78	14.06	18.85	24.08	30.06	37.55	48.04

Tulosten perusteella pienikokoisten karttojen avulla saavutetaan suuret tiivistyssuhteet virheen pysyessä kohtuullisena, kun taas suurempien karttojen avulla virhe ei vähene yhtä selkeästi kuin tiivistyssuhde. Keskimääräiset virheet eivät ole suuria, mutta yksittäisissä pikseleissä virhe voi olla huomattava. Spektrikuvia käsiteltäessä yksittäiset pikselit eivät ole niinkään tärkeitä, mutta menetelmällä tiivistettyjä kuvia käsittelemällä saatuja tuloksia voi pitää vain suuntaa antavina.

6.1. Parannuksia menetelmään

Vaikka itseorganisoituvan kartan avulla suoritettua vektorikvantisoinnin avulla spektrikuvat saatiin tiivistettyä huomattavasti pienemmiksi, tiivistämisen tuloksena on opetettu kartta ja indeksikuva, jotka myös voidaan tiivistää. Yksinkertaisin tapa tiivistää karttaa ja indeksikuvaa on tallentaa ne bittikarttana ja sen jälkeen muuttaa johonkin toiseen kuvaformaattiin.

Yleiset hukkaamattomat kuvien tiivistysmenetelmät (GIF, TIFF, PCX, BMP) paransivat tiivistyssuhdetta vain hyvin vähän, jotkut eivät ollenkaan. Hukkaavista menetelmistä JPEG paransi tiivistyssuhdetta, mutta väärästi kuvaa liikaa. Koska

valmiit menetelmät eivät olleet hedelmällisiä, kokeiltiin oman menetelmän kehittämistä.

6.1.1. Deltakoodaus

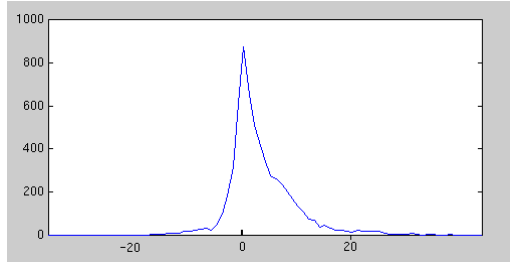
Menetelmällä tiivistetty spektrikuva koostuu opetetusta kartasta sekä indeksikuvasta, jotka myös voidaan tiivistää. Kohosen itseorganisoituvan kartan eräs ominaisuus on, että se säilyttää topologisen riippuvuuden, eli samankaltaiset pikselit ovat myös opetetussa kartassa lähekkäisissä spektreissä. Tämän rakenteellisen ominaisuuden perusteella opetettu kartta voidaan tiivistää siten, että kutakin spektriä ei talleteta absoluuttisina arvoina, eli 25:nä lukuna väliltä 0-255. Järkevämpää onkin huomioida spektrin kunkin aallonpituuden muutos verrattuna vastaavaan aallonpituuteen edellisessä spektrissä, koska muutoksen arvo on pienempi kuin tavu.

6.1.2. 2-ulotteinen ennakointi

Koska myös kunkin spektrin peräkkäisissä aallonpituuksissa on merkittävää riippuvuutta, voitaisiin muutos laskea myös siten. Parempaan tulokseen päästäisiin, jos käytettäisiin molempia riippuvuuksia hyväksi. Tämä onkin mahdollista ennakoimalla molempien riippuvuuksien avulla aallonpituuden arvo ja ilmoittamalla virhe oikeaan aallonpituuteen.

6.1.3. Huffman-koodaus

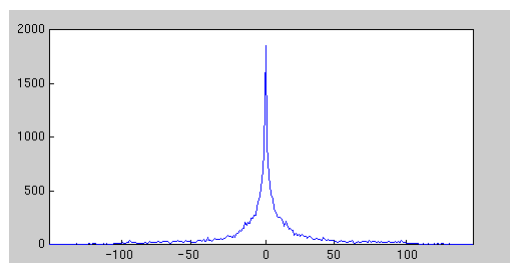
Luokittelemalla 256 värisessä opetetussa kartassa olevien virheiden arvot niiden esiintymismäärän mukaan saadaan kuvan 21 esittämä käyrä. Käyrästä voidaan todeta pienten virheiden olevan huomattavasti todennäköisempiä kuin satunnaisten, suurempien virheiden. Ennakoimalla saatujen virheiden todennäköisyysjakauma on hyvin lähellä Laplace-jakaumaa.



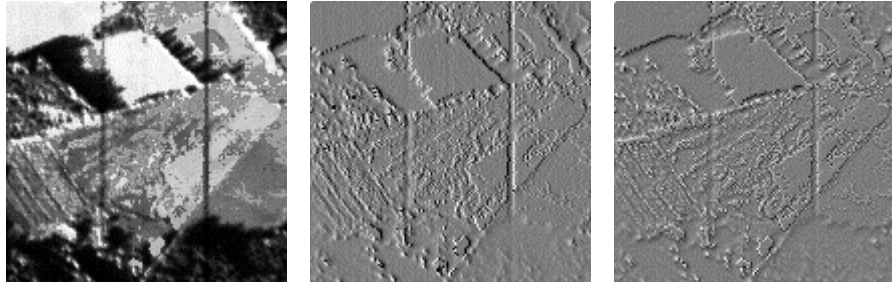
Kuva 21. Virheiden jakautuminen esiintymismäärän mukaan opetetussa kartassa.

Tällaista aineistoa, jossa on joitakin useasti esiintyviä arvoja ja useita harvoin esiintyviä arvoja, voidaan tiivistää koodaamalla arvot eri pituisilla yksilöllisillä bittijonoilla. Optimaaliset bittijonot saadaan selville arvojen todennäköisyyksien avulla käyttämällä Huffman-koodausta. Tiivistettyihin tiedostoihin sisällytetään myös bittijonoja vastaavat arvot sisältävä puu, jotta Huffman-koodattu data saadaan palautettua.

Koska spektrikuvat ovat usein kaukokartoitukseen otettuja kuvia maanpinnasta, niissäkin esiintyy 2-ulotteista riippuvuutta viereisissä pikseleissä. Edellä käytyä menetelmää itseorganisoidujen karttojen tiivistämiseen voidaan käyttää myös indeksikarttaan, jossa olevat peräkkäiset indeksit kuvautuvat usein lähellä toisiaan oleviin ryhmiin. Kuvassa 22 on indeksikartan indekseille tehty vastaava käyrä kuin opetetulle kartalle.



Kuva 22. Virheiden jakautuminen esiintymismäärän mukaan indeksikuvassa.



Kuva 23. 64-värinen indeksikuva, kuva deltakoodattuna ja kuva 2D-ennakoivalla koodauksella.

Kuvassa 23 on 64-värinen indeksikuva, sama kuva deltakoodattuna ja ennakoivalla koodauksella käyttäen ennustajaa $(x(i,j-1) + x(i-1,j))/2$. Kahdessa jälkimmäisessä on käytetty apuna myös Huffman-koodausta. Normaalin kuvan entropia on 6 bittiä/pikseli, deltakoodatun 4.52 bittiä/pikseli ja kaksiuolteista tietoa hyväksi käyttäen ennakoivasti koodatun 4.37 bittiä/pikseli. Parannukset entropiassa johtuvat siitä, että pikselin informaatio esitetään lyhyemmällä bittijonolla. Kaksiuolteisella ennakoinnilla saadaan hieman tarkempia tuloksia kuin pelkästään edellisen pikselin avulla ennakoitaessa. Algoritmissa 4 on esitetty, kuinka algoritmilla 2 saatu indeksikuva ja kartta tiivistetään edelleen käyttämällä hyväksi kaksiuolteista ennakoivaa koodausta ja Huffman-koodausta. Menetelmän alkuosa on sama kuin algoritmi 2.

Algoritmi 4. Tiivistys 2D-ennakoivalla Huffman-koodauksella.

1. Indeksikuvalle ja kartalle suoritetaan 2-ulotteinen ennakoiva koodaus käyttäen ennustajaa $(x(i,j-1) + x(i-1,j))/2$.
2. Saaduille arvoille lasketaan esiintymistodennäköisyydet.
3. Esiintymistodennäköisyyksien avulla lasketaan arvoille Huffman-koodauksen avulla koodisanat.
4. Sijoitetaan koodisanat indeksikuvaan ja karttaan.

Taulukossa 4 on algoritmilla 4 tiivistettyjen karttojen ja indeksikuvien koko eri kokoisilla kartoilla ja niiden tiivistyssuhteet. Koska tiivistysalgoritmi on hukkaamaton, vääristymä pysyy samana kuin aiemminkin (taulukko 3).

Taulukko 4. Tiivistyssuhteet parannetulla menetelmällä (algoritmi 4).

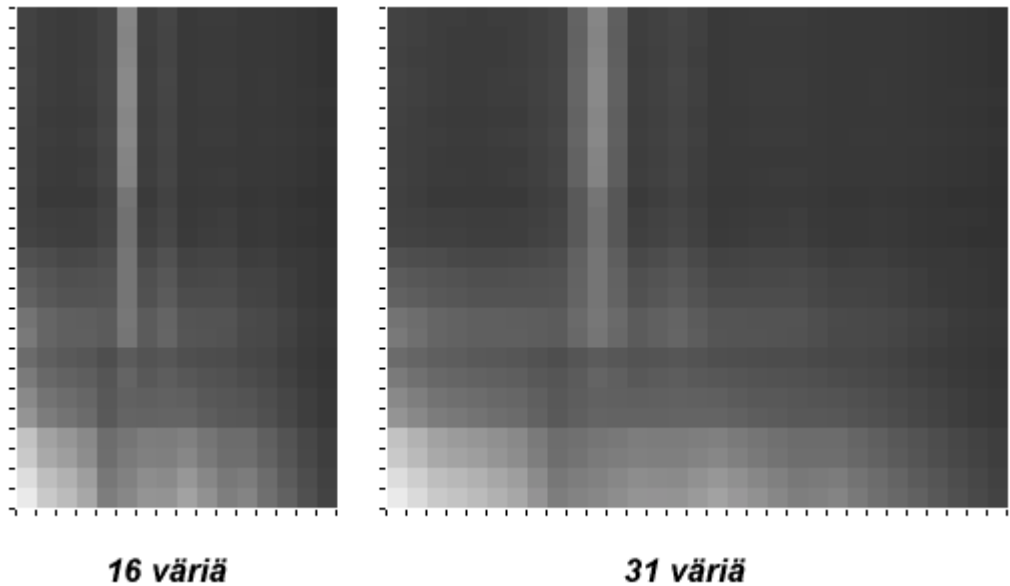
SOM koko	2048	1024	512	256	128	64	32	16
Taulukon 3 tiivistyssuhde	6.27	9.78	14.06	18.85	24.08	30.06	37.55	48.04
Som tavuina (algoritmi 4)	25000	12500	6400	2900	1550	830	500	300
Indeksi tavuina (algoritmi 4)	26950	24200	20600	16600	13700	10700	8300	5900
Tiivistyssuhde (algoritmi 4)	9.43	13.35	18.15	25.13	32.13	42.50	55.68	79.03

Kartan ja indeksikuvan tiivistämisen avulla saavutetaan n. 30-40% parannus tiivistyssuhteeseen verrattuna algoritmiin 2.

6.2. Muita parannusyriytyksiä

Itseorganisoituva kartta muodostaa väritaulukon siten, että toisiaan lähellä olevat spektrit ovat myös taulukossa lähellä toisiaan. Tätä voidaan hyödyntää laajentamalla värien määrää siten, että alkuperäisistä spektreistä saadaan jollakin menetelmällä moninkertainen määrä uusia, hyödynnettäviä spektrejä. Yksinkertaisin menetelmä on spektrien lineaarinen interpolaatio, jolla lisäspektrit muodostuvat alkuperäisten spektrien väliin asteittain.

Menetelmän avulla saatavien ylimääräisten spektrien käyttökelpoisuutta kokeiltiin interpoloimalla 16-spektrisestä kartasta 31-spektrinen kartta. Spektrikuva tiivistettiin käyttämällä saatua karttaa ja tiivistetystä kartasta laskettiin uusien, interpoloimalla saatujen spektrien osuus. Kuvan 24 oikeassa reunassa olevan kartan uudet, lineaarisesti interpoloidut spektrit (15 kpl) muodostavat 42 % tiivistetyn spektrikuvan (aisa140x140) spektreistä, joten uudet spektrit ovat varsin käyttökelpoisia.



Kuva 24. Alkuperäinen SOM ja siitä lineaarisesti interpoloitu SOM.

Interpoloinnin etuna on, että talletettavan kartan koko pysyy samana, mutta käytettävien spektrien määrä kasvaa. Haittapuolena on se, että spektrien määrän kasvaessa myös indeksikuvan viemä tila kasvaa. Myöskään interpoloidut spektrit eivät ole aivan yhtä hyviä kuin saman kokoisessa opetetussa kartassa. Taulukossa 5 on 64:n spektrin kartasta interpoloituja 127:n, 255:n ja 511:n spektrin karttoja vertailtu opetettuihin 128:n ja 256:n spektrin karttoihin. Karttojen koon kasvaessa interpoloinnilla saavutetut edut pienenevät, interpoloidut värit eivät enää ole hyödyllisiä. 511-spektrinen interpoloitu kartta vastaa virheeltään 128-spektristä opetettua karttaa tiivistyssuhteen ollessa huomattavasti pienempi. Tiivistyssuhteen ero johtuu siitä, että vaikka 511-spektrinen interpoloitu kartta on vain puolet 128-spektrisen opetetun kartan koosta, indeksikuvan koko on $n \cdot 1.29$ -kertainen.

Taulukko 5. Lineaarisesti interpoloitujen karttojen tunnuslukuja.

	Kartan värien määrä	Suurin virhe	Keskiarvovirhe	Tiivistyssuhde (algoritmi 2)	Tiivistyssuhde (algoritmi 4)
Opetetut kartat	256	57	2.66	18.85	25.13
	128	58	2.78	24.08	32.13
	64	59	2.92	30.06	42.50
Interpoloidut kartat	127 (64)	59	2.82	26.13	33.72
	255 (64)	59	2.79	23.11	28.11
	511 (64)	60	2.77	20.72	22.79

Karttaa interpoloimalla saavutettu etu siis menetetään indeksikuvan koon kasvun takia. Varsinkin pienillä kartoilla interpoloinnilla saavutetaan hyviä tuloksia virheen suhteen, mutta kartan tiivistyssuhteen paraneminen on vähäistä kartan pienen koon takia ja indeksin koon kasvaminen yhdellä bitillä vaikuttaa enemmän kuin suurilla kartoilla, jolloin taas virheen pieneneminen on vaatimatonta.

6.3. Vertailua kartan ja indeksin tiivistämiseen muilla menetelmillä

Opetetun kartan ja indeksikuvan tiivistämiseksi kokeiltiin myös muita hukkaamattomia menetelmiä, niin kuvien tiivistämiseen tarkoitettuja kuin yleisiäkin algoritmeja. Yleiset algoritmit eivät osanneet hyödyntää kuvan 2-ulotteista rakennetta ja kuvaformaateista GIF toimi vain, jos kartassa oli alle 256 spektriä. GIF myöskin tallensi turhaa informaatiota eikä tiivistyssuhteissa päässyt lähellekään käytettyä menetelmää. FELICS- CALIC- tai TMW-algoritmia /5,21,13/ käyttäviä valmiita ohjelmia ei ollut käytettävissä, eikä niiden koodaamista itse nähty tarkoituksenmukaiseksi, joten vertailussa käytettiin apuna kuvaa, jolle oli laskettu algoritmien tiivistyssuhde.

Kuvankäsittelyssä yleisesti testauksessa käytetylle harmaatasokuvalle (kuva 25) laskettiin tiivistyssuhde 2-ulotteisella ennakoivalla huffmankoodauksella ja verrattiin sitä samalla kuvalla muilla algoritmeilla saatuihin tiivistyssuhteisiin.



Kuva 25. Lena-kuva.

Lena-kuva ei täysin vastaa tyypillisiä opetettuja karttoja tai indeksikuvia, mutta tuloksien (taulukko 6) perusteella menetelmä tiivistäisi luonnollisia kuviakin hyvin.

Taulukko 6. Lena-kuvan tiivistyssuhteita eri menetelmillä.

Menetelmä	Bittejä/pikseli
GIF	8.07
Unix compress	7.5
Gnu zip	7.1
Pkzip	6.9
2d ennakova Huffman (algoritmi 4)	4.7
FELICS	4.7
CALIC	4.4
TMW	4.3

7. YHTEENVETO

Tässä diplomityössä tutkittiin spektrikuvan tiivistämistä ja kehitettiin uusia menetelmiä käyttäen hyväksi itseorganisoituvaa karttaa. Kehitetyillä menetelmillä voidaan tiivistää spektrikuvia huomattavasti. Tässä työssä esitettyä menetelmiä on verrattu julkaisussa /20/ julkaisun /9/ menetelmiin. Käytetyllä testiaineistolla saavutetut tiivistyssuhteet olivat 3-4 kertaa parempia kuin julkaisussa /9/ esitetyt.

Menetelmiä voitaisiin käyttää esimerkiksi raakavedosten siirtämiseksi verkossa, jolloin alkuperäistä spektrikuvaa ei tulisi siirrettyä vain, jotta havaittaisiin kuvan olevan käyttötarkoituksiin soveltumaton, tai muuten tarpeeton. Prosessointi voitaisiin suorittaa raakavedoksella ja tarkemmat tulokset voitaisiin haluttaessa laskea alkuperäisellä kuvalla. Menetelmällä tiivistettyjä spektrikuvia voitaisiin myös käyttää sovelluksissa, joissa pienet spektrikuvan virheet olisi hyväksyttävissä.

Menetelmää voitaisiin kehittää monin tavoin. Samankaltaisia kuvia käytettäessä voitaisiin opettaa vain yksi itseorganisoituva kartta, jonka perusteella kaikki indeksikuvat muodostettaisiin. Tällöin tiivistyssuhde saataisiin yksittäistä kuvaa kohti vieläkin paremmaksi, koska jokaisen kuvan mukana ei tarvitsisi tallettaa erillistä spektrit sisältävää opetettua karttaa. Kartta voitaisiin opettaa antamalla sille syötteenä useita kuvia tai useita jo opetettuja karttoja. Vaikka yhteinen kartta ei minkään yksittäisen spektrikuvan tapauksessa toimisi niin hyvin kuin sitä varten opetettu kartta, voisi yhteinen kartta olla kooltaan paljon suurempi, jolloin vääristymätkin vähenisivät.

Myös käytettyä opetetun kartan ja indeksikuvan tiivistämistä voitaisiin kehittää. Käyttämällä adaptiivisia ennustajia seuraavan pikselin ennustaminen kuvaa läpi käytäessä saataisiin parempia tuloksia, samoin vaihtamalla Huffman-koodaus

aritmeettiseen koodaukseen saavutettaisiin muutaman prosentin parannus tiivistyssuhteeseen.

Vaikka uudet kuvien tiivistämiseen kehitetyt hukkaamattomat algoritmit CALIC /21/ ja TMW /13/ tiivistävät vieläkin tehokkaammin, ongelmana on valmiin ohjelman tai koodin saatavuus näille algoritmeille. Myös uudet lähes hukkaamattomat algoritmit voivat paremmalla tiivistyssuhteellaan kompensoida aiheuttamaansa virhettä, varsinkin indeksikuvan tapauksessa, jossa pieni virhe indeksissä ei välttämättä ole kovin haitallista itseorganisoituvan kartan topologian säilyttävän luonteen johdosta.

Kehitettyjen menetelmien puutteista voisi niiden tietoa hukkaavan luonteensa lisäksi mainita niiden vaatiman suuren prosessointitarpeen, joka Matlab-koodina toteutettuna vaatii suhteettoman paljon aikaa ja muistia.

Työssä esitettyjä menetelmiä voidaan siis kehittää monella tavalla. Kuitenkin nykyisetkin esitetyt menetelmät antavat hyvin lupaavia tiivistyssuhteita. Samalla osoitetaan, että perinteisiä tiivistystekniikoita ja neuraalilaskentaa kannattaa yhdistää. Työn tuloksena on syntynyt uusi menetelmä spektrikuvien tiivistämiseksi.

LÄHTEET

1. N. Ekstrand. *Some Results on Lossless Compression of Grayscale Images*. Teknisk licentiat thesis, University of Lund, Sweden, 1998.
2. R. Fisher, S. Perkins, A. Walker and E. Wolfart. *Hypermedia Image Processing Reference*. J.Wiley & sons, 1996.
3. R. M. Friedhoff, W. Benzon. *Visualization*. Harry N. Abrams Inc, 1989.
4. W. Fulco. *Vector Quantization Compression Algorithms*. Multimedia Systems Design, Volume 2, 1998.
5. P. G. Howard. *The Design and Analysis of Efficient Lossless Data Compression Systems*. Brown University, Technical Report Number CS-93-28, June 1993.
6. D. A. Huffman. *A Method for the Construction of Minimum Redundancy Codes*. Proceedings of the Institute of Radio Engineers, Volume 40, Number 9, pp. 1098-1101, September 1952.
7. R. W. G. Hunt. *Measuring Colour*. 2. painos. Ellis Horwood, 1992.
8. *Introduction to Self-Organizing Map*. Internet:
(<http://www.cis.hut.fi/projects/somtoolbox/somalg.html>)
9. A. Kaarna, P. Zemcik, H. Kälviäinen, J. Parkkinen. *Multispectral Image Compression*. Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98), pp. 1264-1267, Australia, 1998.

10. T. Kohonen. *Self-Organizing Maps*. 2.painos. Springer, 1997.
11. P. Koikkalainen. *Neurolaskennan mahdollisuudet*. Tekes julkaisu nro. 43, 1994.
12. S.P. Lutrell, *Hierarchical Vector Quantization*, IEE Proceedings, Volume 136, Number 6, pp. 405-413, 1989.
13. B. Meyer, P. Tischer. *TMW - a New Method for Lossless Image Compression*. Proceedings of International Picture Coding Symposium PCS97 conference , 1997.
14. *Munsell Book of Color*, Matte Finish Collection, Munsell Color, Baltimore, USA, 1976.
15. A. N. Netravali, B. G. Haskell. *Digital Pictures, Representation and Compression*. 2. painos. Plenum Press, 1988.
16. J. Ruokolainen, M. Gröhn. *Tieteellinen visualisointi*. CSC - Tieteellinen laskenta Oy, 1996.
17. J. C. Russ. *The Image Processing Handbook*. 3. painos. Springer-Verlag 1999.
18. A.D. Ryer. *Light Measurement Handbook*. International Light Inc., 1998.
19. C. E. Shannon, W. Weaver. *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, 1949.
20. P. J. Toivanen, A. Lehtinen, H. Kälviäinen. *Two-Stage Multispectral Image Compression Using the Self-Organizing Map*. Proceedings of the 11th Scandinavian Conference on Image Analysis (SCIA'99), Volume 2, pp. 903-909, Greenland, 1999.

21. X. Wu, N. Memon. *Context-based, Adaptive, Lossless Image Codec*. IEEE Trans. On Communications, Volume 45, Number 4, April 1997.

22. J. Ziv, A. Lempel. *Compression of individual Sequences via Variable Rate Coding*. IEEE Trans. Information Theory, Volume 24, pp. 530-536, September 1978.