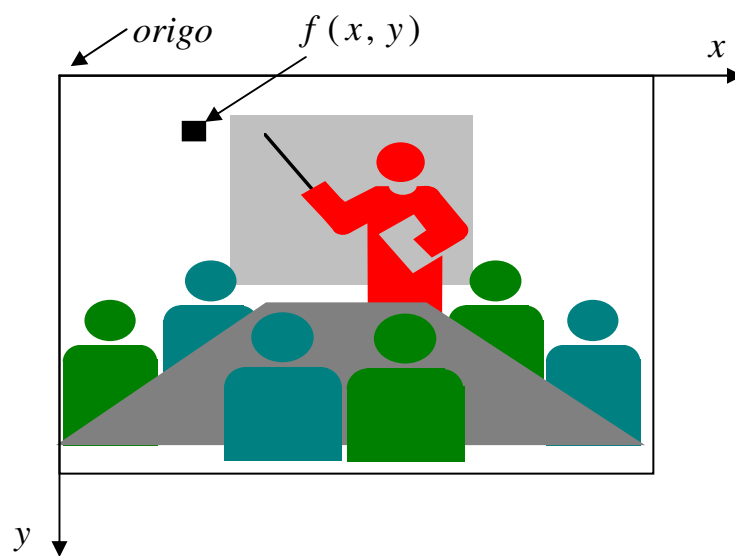


DIGITAALINEN KUVANKÄSITTELY



JUKKA JAUHAINEN 2006

1	<i>Johdanto</i>	3
1.1	Digitaalisen kuvan määritelmä	3
1.2	Kuvankäsittelyn vaiheet	3
1.3	Kuvankäsittelyjärjestelmän osat	4
2	<i>Perusteita</i>	5
2.1	Ihmissilmän rakenne	5
2.2	Fotometria	7
2.3	Silmän mukautuminen ja kirkkauden erotuskyky	8
2.4	Yksinkertainen kuvamalli	11
3	<i>Pikselien väliset relaatiot</i>	13
3.1	Pikselin naapurit	13
3.2	Viereisyys ja liittyvyys	13
3.3	Liittyneiden komponenttien nimeäminen	15
3.4	Relaatiot, ekvivalenssi ja transitiivinen sulkeuma	16
3.5	Etäisyysmitat	18
3.6	Aritmeettis-loogiset operaatiot	20
3.7	Maskioperaatiot	22
4	<i>Kuvan ehostus</i>	23
4.1	Pisteprosessointi kuvatasossa	24
4.2	Histogrammi ja sen prosessointi	26
4.3	Kuvien vähennys	29
4.4	Kuvien keskiarvoistus	30
4.5	suodatus kuvatasossa	31
4.5.1	Alipäästösuodattimet (smoothing filters)	31
4.5.2	Ylipäästösuodattimet (sharpening filters)	32
4.5.3	Derivaattasuodattimet (derivative filters)	33
4.6	Kuvan ehostus taajuustasossa	36
4.6.1	Kaksiulotteinen Fourier-muunnos	36
4.6.2	Konvoluutioteoreema	37
4.6.3	Alipäästösuodattimet	38
4.6.4	Ylipäästösuodattimet	40

Tämä luentomateriaali perustuu kirjaan:

Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing, Addison Wesley Publishing Corporation (1993).

1 JOHDANTO

Kiinnostus digitaalista kuvankäsittelyä kohtaan juontaa juurensa kahdesta sovellusalueesta. Nämä ovat kuvanlaadun parantaminen kuvien tulkinnan helpottamiseksi, sekä konenäön ja robotiikan sovellutukset. Edellä mainitussa kuvia muokataan ihmissilmälle luettavampaan muotoon ja jälkimmäisessä pyritään automatisoimaan esimerkiksi teollisuuden prosesseja. Kuvankäsittely tieteenalana on kehittynyt voimakkaasti 60-luvulta alkaen. Tähän on vaikuttanut luonnollisesti tietokoneiden, ohjelmointikielten ja algoritmien voimakas kehitys. Vaikka sovellusalueet vaihtelevat avaruusohjelmista ja satelliittikuvista röntgenkuvaukseen, on kaikkien taustalla samat kuvankäsittelyn perustekniikat.

1.1 DIGITAALISEN KUVAN MÄÄRITELMÄ

Harmaasävykuvalla tai yksinkertaisesti kuvalla tarkoitetaan kaksiulotteista valon intensiteettifunktiota $f(x,y)$, missä x ja y ovat paikkakoordinaatit ja funktion f arvo on verrannollinen kuvan *kirkkauteen* (brightness) eli *harmaasävyyden* (gray level) kussakin kuvan pisteessä. Otsikkosivulla on esitetty tyypillisesti käytetty koordinaatiston valinta. Kuvan origo, eli koordinaatti $(0,0)$ on vasemmassa yläkulmassa. Gonzalez&Woodsissa käytetään poikkeuksellista koordinaattijärjestelmää, jossa x -koordinaatti kasvaa ylhäältä alas ja y -koordinaatti vasemmalta oikealle. *Digitaalisessa kuvassa* kuva-ala muodostuu matriisista, jossa $f(x,y)$ voi saada vain diskreettejä arvoja. Digitaalisen kuvamatriisin yhtä kuva-alkiota sanotaan *pikseliksi* (picture element, pixel, *pel*). Digitaalisen kuvan koko vaihtelee sovelluksen mukaan. Usein kuitenkin käytetään neliömatriisia, jossa on yhtä monta riviä ja saraketta. Lisäksi harmaasävyjen määrä per pikseli on yleensä $2:n$ potenssi. Esimerkiksi tyypillinen kuvakoko, joka vastaa mustavalko-TV:n kuvanlaatua olisi 512×512 -matriisi 128 harmaasävyllä.

1.2 KUVANKÄSITTELYN VAIHEET

Tarkastellaan seuraavaksi, mitä vaiheita digitaaliseen kuvankäsittelyyn kaikkienensa liittyy. Ensimmäinen vaihe on *kuvan kerääminen* (image acquisition). Tähän tarvitaan jokin sensori, esimerkiksi

skanneri tai digitaalikamera. Seuraavaksi kuva *esikäsitellään* (preprocessing). Sen avulla voidaan muokata kuvan kontrastia, poistaa siitä kohinaa. Kolmas vaihe on nimeltään *segmentointi*. Sen avulla kuva jaetaan osiin tai siitä eristetään mielenkiintoalueita. Esimerkkinä mainittakoon kirjainten tunnistus skannatulta A4-sivulta. Segmentoinnin lopputuloksena saadaan yleensä raakaa pikselidataa, joka esittää joko mielenkiintoalueen reunaviivan tai kaikki mielenkiintoalueen sisään jäävät pisteet. Reunaviiva halutaan yleensä silloin, kun ollaan kiinnostuneita alueen muodosta. Sisään jäävä alue halutaan vastaavasti silloin, kun ollaan kiinnostuneita objektin sisäisestä rakenteesta. Joissain sovellutuksissa tarvitaan molempia segmentointitapoja yhtä aikaa. Segmentoinnin pohjalta tehdään neljäntenä vaiheena hahmontunnistus (feature selection). Sen avulla kuvataan, mitä piirteitä segmentoitu objekti sisältää ja miten se on erotettavissa toisista segmentoiduista objekteista. Esimerkiksi kirjainten tunnistuksen yhteydessä määritellään, missä kohtaa mitäkin kirjainta on reikiä ja missä vastaavasti yhtenäisiä alueita. Kuvankäsittelyn viimeinen vaihe koostuu objektin tunnistuksesta (recognition) ja tulkinnasta (interpretation). Esimerkiksi edellä tehdyn hahmontunnistusalgoritmin tuloksena saadut objektit luokitellaan ja nimetään eri kirjaimiksi.

Yleensä kuvankäsittelyssä tarvitaan etukäteistietoa käsiteltävästä kohteesta. Yksinkertaisimmillaan tämä ns. tietämuskanta (knowledge base) voi olla esimerkiksi tieto siitä, että skannattu A4-sivu sisältää tekstiä, siis kirjaimia. Huomattavan paljon monimutkaisemmasta tietämuskannasta voisi olla esimerkiksi radiologinen kuvantulkinta, jossa tehdään diagnoosia röntgenkuvasta. Tietämuskanta ohjaa kaikkia edelläkuvattuja kuvankäsittelyn vaiheita.

1.3 KUVANKÄSITTELYJÄRJESTELMÄN OSAT

Yleiskäyttöisen kuvankäsittelyjärjestelmän osat ovat 1) kuvankeräys 2) tallennus 3) prosessointi 4) tiedonsiirto ja 5) näyttö.

Kuvankeräysosa koostuu kahdesta osasta. Ensiksikin tarvitaan jokin laite, jolla kuva saadaan tallennettua. Laite on herkkä jollekin osalle sähkömagneettisen säteilyn spektriä. Sen on toisin sanoen pystyttävä taltioimaan esimerkiksi röntgen-, UV-, näkyvän valon- tai infrapuna-alueen säteilyä. Se tuottaa sähköisen signaalin, jonka intensiteetti on verrannollinen saapuneen säteilyn määrään. Toinen osa on digitoija, jolla analoginen sähköinen signaali muutetaan digitaaliseksi tietokoneen ymmärtämään muotoon. Ajatellaanpa esimerkiksi röntgenkuvauslaitteiston periaatetta. Siinä röntgenputkesta saatava säteily osuu kohteeseen ja vaimenee kudoksen tiheyseroista johtuen eri lailla eri kudoksissa. Kohteen

takapuolella on jokin media johon kuva tallentuu. Tallennusmedia voi olla filmi, Valomonistinputki-TV-kamera-yhdistelmä, tai valoherkkä puolijohdemateriaali.

8-bittinen 1024x1024 kokoinen kuva vaatii noin 1 Mt tallennustilaa ($1024 \cdot 1024 = 1048576$). Digitaalisten kuvien vaatima tilantarve on siis suuri. Tässä on oletettu, että yksi tavu koostuu kahdeksasta bitistä. Tallennusovellutukset voidaan jakaa kahteen luokkaan, prosessoinnin aikana tapahtuvaan väliaikaiseen tallennukseen, sekä pitkäaikaiseen tallennukseen. Kuvista voidaan myös muodostaa laajempia kuva-arkistoja, jolloin itse kuvadatan lisäksi voidaan tietokantamuodossa tallettaa myös muuta kuvaan liittyvää tietoa organisoidussa muodossa.

Kuvaa voidaan pitää lyhytaikaisesti tietokoneen RAM-muistissa. Nykyisin kuitenkin kaikissa koneissa on erillinen näytönohjainkortti, jossa voi olla satoja megatavuja muistia, joka hoitaa kuvankäsittelyn ilman että operaatio kuluttaa pääprosessorin tehoa. Näytönohjaimen avulla päästään nykyään jo huikkeisiin kuvankäsittelynopeuksiin. Pitempiäaikaiseen varastointiin on kehitetty ja kehitetään uusia tekniikoita perinteisen magneettinauhatalennuksen rinnalle. Nykyään jo tavallisten kiintolevyjen kapasiteetit yltyvät 150 Gt:un. Lisäksi on tarjolla kirjoittavat CR-ROM asemat sekä DVD-asetat.

Digitaalisen kuvan prosessointi tarkoittaa yleensä menetelmiä, jotka voidaan ilmaista algoritmeina. Suurin osa prosessoinnista voidaankin hoitaa ohjelmallisesti. Yleisin syy rautatason prosessointiin on nopeusvaatimukset joissain erikoisovellutuksissa, kuten pelit tai DSA (digitaalinen subtraktioangiografia).

2 PERUSTEITA

2.1 IHMISSILMÄN RAKENNE

Silmämunan seinämässä on kolme kerrosta. Edessä muodostaa uloimman kerroksen läpinäkyvä verisuoneton *sarveiskalvo* (*cornea*), sivuilla ja takana läpinäkymätön *kovakalvo* (*sclera*). Kovakalvon sisäpuolella on *suonikalvo* (*choroidea*), jossa on runsaasti verisuonia. Sisinnä on valoon reagoiva verkkokalvo (*retina*). Suonikalvon jatkeena on rengasmaisen paksuuntuma, *sädekehä* (*corpus ciliare*), jossa puolestaan on kiinni *värikalvo* eli iiris (*iris*). Värikalvon keskellä on pyöreä reikä, *mustuainen* eli pupilli

(*pupilla*), jonka läpi valo kulkee silmänpohjaan. Värikalvo muistuttaa toiminnaltaan kameran himmennintä, sillä sen liikkeet suurentavat ja pienentävät mustuaista ja säätelevät siten silmään tulevan valon määrää. Silmään tuleva valo taittuu voimakkaasti sarveiskalvon ulkorajalla. Silmän mykiö eli linssi taittaa valoa vielä lisää, ja katsottavasta kohteesta muodostuu kuva verkkokalvoon. Linssin taittokyky on pienempi kuin sarveiskalvon, mutta linssin taittovoima muuttuu katseluetäisyyden mukaan, niin että ihminen pystyy näkemään tarkasti sekä lähelle että kauas.

Ihminen pystyy näköaistillaan havaitsemaan sähkömagneettista säteilyä, jonka aallonpituus on 397 - 723 nm. Valoon reagoi verkkokalvo. Suuri osa ns. reseptoreiden vastaanottamasta näköinformaatiosta muokkautuu jo verkkokalvossa tavalla, joka on yleensä ominaista aivoille. Näköreseptoreja on kahta tyyppiä, sauvoja ja tappeja. Tappeja on n. 6-7 miljoonaa kappaletta. Ne sijaitsevat verkkokalvon keskikuopan läheisyydessä ja ovat erittäin herkkiä havaitsemaan eri värejä mutta eivät toimi hämärässä. Ihminen pystyy havaitsemaan pieniä yksityiskohtia, koska kukin tappisolu on yhteydessä omaan hermopäätteeseensä. Silmän lihakset liikuttavat silmämunaa siten, että kohteen kuva osuu keskikuoppaan. Tappisolujen näkökyky on niin sanottua fotooppista- eli kirkasnäkökykyä. Sauvasolujen määrä on huomattavasti suurempi, n. 75 - 100 miljoonaa ja ne ovat jakaantuneet tasaisesti retinaan. Monta sauvaa on yhteydessä yhteen hermopäätteeseen. Siksi niiden avulla ei voida nähdä kovin tarkasti. Sauvojen avulla voidaan havaita "yleiskuva" näkökentän alueelta. Tappi- ja sauvasolujen erilaisesta herkkydestä eri valaistusvoimakkuuksille johtuu, että sama kohde näyttää eri väriseltä päivänvalossa ja hämärässä. Sauvojen näkökyky on luonteeltaan skotooppista- eli hämäränäkökykyä.

Havaintoalueensa laajuudessa ihmissilmä voittaa monet tekniset mittauslaitteet. On laskettu, että valoistimuksen voi edullisissa olosuhteissa saada aikaan yksi ainoa sauvasoluun osunut foton. Toisaalta vielä 10^{10} kertaa suurempi valomäärä aistitaan valona. On arvioitu, että ihminen pystyy havaitsemaan 160 erilaista väriä. Näiden lisäksi pystymme havaitsemaan erilaisia väri-intensiteetin (valoisuuden) asteita sekä erilaisia värikylläisyyden asteita. Nämä tekijät voivat yhdistyä eri tavoin, joten erilaisia havaittavia värivaihteita on hyvin suuri määrä.

Verkkokalvon tappeja on kuitenkin vain kolmenlaisia. Kussakin tappilajissa on yhtä valoon reagoivaa väriainetta. On tapana puhua siniherkistä, viherherkistä ja punaherkistä tappeista. Useimmissa tapauksessa väriin reagoi ainakin kaksi ellei kaikki kolmekin tappilajia. Ne toimivat kuitenkin eri tilanteissa erilaisella suhteellisella voimakkuudella. Tämä riittää aivoissa tapahtuvaan värianalyysiin.

2.2 FOTOMETRIA

Silmä on optinen koje, jonka polttoväliä voidaan tarvittaessa säädellä hyvinkin laajalla alueella. Fotometria käsittelee valaistukseen liittyviä suureita, ottaen huomioon silmän erilaisen herkkyyden eri aallonpituusalueille. Silmä on hyvässä valaistuksessa herkin valolle, jonka aallonpituus on 555 nm (keltainen). Hämärässä herkkyysmaksimi on n. 510 nm. Herkkyys heikkenee kohti näkyvän valon alueen ääriä. Tätä sanotaan silmän spektriherkkyydeksi.

Koska valonlähteet säteilevät yleensä koko avaruuteen tai osaan avaruutta, tarvitaan valaistussuureiden määrittelyyn kolmiulotteinen avaruuskulma Ω . Se määritellään kulman pallosta leikkaaman pinnan pinta-alan A ja pallon säteen r suhteena. Avaruuskulman yksikkö on steradiaani (sr). Koko avaruus vastaa avaruuskulmaa 4π sr.

$$\Omega = \frac{A}{r^2}$$

Fotometriaan liittyvä SI-järjestelmän perussuure on *valovoima* I . Se kuvaa valon voimakkuutta tarkastelusuunnassa siten, että siinä on otettu huomioon silmän spektriherkkyys. Valovoiman yksikkö on *kandela* (cd). Valovoima on valonlähteeseen, ei havaitsijaan, liittyvä suure. Kandela on saanut nimensä englannin kielen sanasta candle (kynttilä). Yhden steariini-kynttilän valovoima on n. 1 cd.

Valovirta on valonlähteen valotehoa kuvaava suure. Pistemäisen, kaikkiin suuntiin tasaisesti säteilevän valonlähteen synnyttämä valovirta Φ avaruuskulmaan Ω on kyseiseen avaruuskulmaan osuva silmän spektriherkkyydellä painotettu säteilyteho. Valovirta saadaan kertomalla valovoima I avaruuskulmalla:

$$\Phi = I \Omega$$

Valovirran yksikkö on *luumen* (lm). Valonlähteen *valotehokkuus* η määritellään valovirran Φ ja sähkötehon P suhteena

$$\eta = \frac{\Phi}{P}$$

Valotehokkuuden yksikkö on lm/W. Hehkulampun valotehokkuus on alle 20 lm/W, parhaiden loistelamppujen lähes 100 lm/W. Näkyvän alueen ulkopuoliselle säteilylle se on 0. Suuretta, joka kuvaa pintaan osun valovirran tiheyttä, kutsutaan *valaistusvoimakkuudeksi* E . Se on pinnalle osuva valovirta pinta-alayksikköä kohti

$$E = \frac{\Phi}{A} = \frac{I}{r^2} sr$$

Yhtälö pätee tilanteelle, jossa valo tulee kohtisuorasti pinnalle. Valaistusvoimakkuuden yksikkö on *luksi* (lx). Se kertoo, kuinka "hyvin" tarkasteltava pinta on valaistu. Esimerkiksi ulkotiloissa pihojen suositeltava valaistusvoimakkuus on 30 luksia, keittiön yleisvalaistus 150 luksia, luokkahuoneiden 300 - 500 luksia ja erittäin suurta tarkkuutta vaativa työ 2000 - 5000 luksia.

Edellä on käsitelty ainoastaan pistemäisiä valonlähteitä. Kun tarkastelu laajennetaan säteileville pinnoille, määritellään suure *luminanssi* eli *kirkkaus* (brightness) L . Se kuvaa säteilevän tai heijastavan pinnan kirkkautta, eli pinnan valovoimaa tarkastelusuuntaa vastaan kohtisuoraa pinta-alayksikköä kohti.

$$L = \frac{I}{A_{\perp}}$$

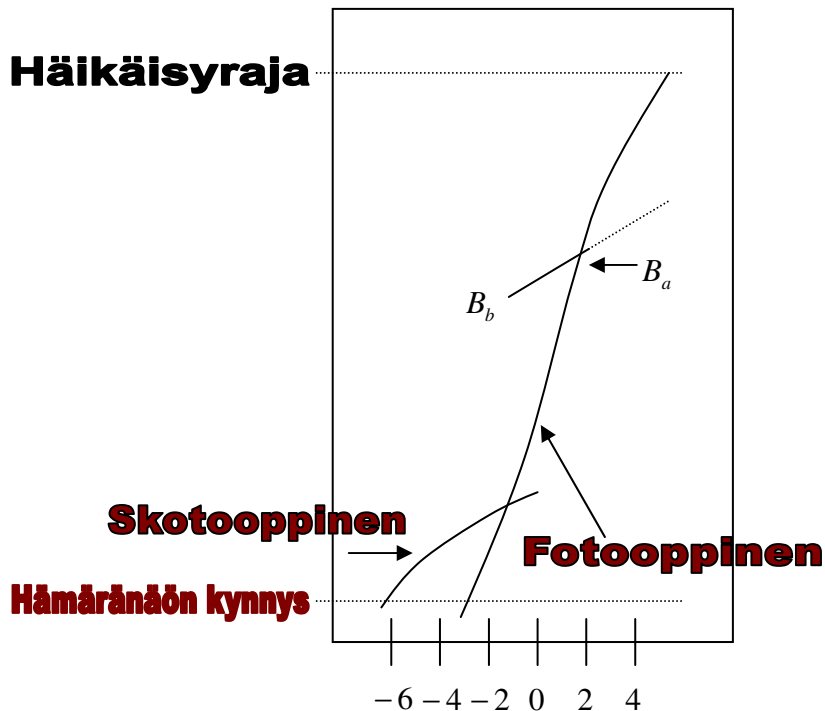
Luminanssin yksikkö $[L]=\text{cd}/\text{m}^2$. Auringon luminanssi on $10^{10} \text{ cd}/\text{m}^2$, pilvettömän taivaan keskipäivällä n. $4 \times 10^3 \text{ cd}/\text{m}^2$, täysikuun $3 \times 10^3 \text{ cd}/\text{m}^2$ ja pilvettömän kuuttoman yötaivaan $10^{-3} \text{ cd}/\text{m}^2$. 100 W:n hehkulampun luminanssi on kirkkaalla kuvalla noin $7 \times 10^6 \text{ cd}/\text{m}^2$ ja himmennetyllä $2 \times 10^5 \text{ cd}/\text{m}^2$. Tietokonemonitorien luminanssit vaihtelevat sadan cd/m^2 :n kahta puolen.

Mitkä seikat sitten vaikuttavat kohteen näkemiseen? Pelkkä valaistusvoimakkuus ei riitä. Jos pinta heijastaa huonosti, se näyttää kirkkaassakin valossa mustalta. Näköhavainnon kannalta luminanssi on keskeinen suure, koska se sisältää valaistusvoimakkuuden lisäksi myös pinnan heijastussuhteen. Toinen merkittävä tekijä on kappaleen ja taustan välinen *kontrasti*. Tarkastellaan sitä jatkossa tarkemmin.

2.3 SILMÄN MUKAUTUMINEN JA KIRKKAUDEN EROTUSKYKY

Digitaaliset kuvat esitetään joukkona diskreettejä kirkkausarvoja. Siksi on tärkeää tietää silmän kyky erotella eri kirkkaustasoja toisistaan. Ihmissilmä pystyy mukautumaan (adaptaatio) suunnattoman laajaan

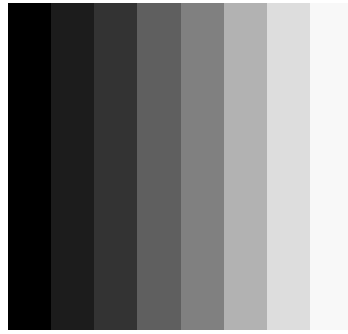
valon intensiteettitason vaihteluun, suuruusluokaltaan 10^{10} . Alarajana on hämäränäön kynnyks, eli skotooppinen kynnyks ja ylärajana häikäisyraja (glare limit). Kokeellinen aineisto tukee sitä otaksumaa, että subjektiivinen kirkkaus (ihmisen näköjärjestelmän havaitsema kirkkaus) on silmään tulevan valon intensiteetin logaritmin funktio. Oleellinen piirre on, että ihmisen visuaalinen järjestelmä ei kykene toimimaan näin laajalla dynaamisella alueella samanaikaisesti. Sen sijaan järjestelmä mukautuu tietylle, suhteellisen pienelle, kirkkausalueelle kerrallaan. Tiettyjen olosuhteiden vallitessa visuaalisen järjestelmän herkkyytaso kuvaa suure kirkkauden mukautumistaso (brightness adaptation level) , joka kuvan tapauksessa vastaa kirkkaustasoa B_a . Lyhyt leikkaava käyrä kuvaa niitä subjektiivisen kirkkauden arvoja, jotka silmä voi havaita silloin, kun se on mukautunut esitetulle kirkkaustasolle. Tämä alue on varsin rajallinen; kuvan tasoa B_b matalammat kirkkaustasot havaitaan kaikki mustina. Käyrän katkoviivoitettua yläosaa ei ole rajoitettu, mutta jos käyrä jatkuu liian kauas, B_a nousee vastaavasti ylemmäs.



Silmän kyky erotella (diskriminoida) *kirkkauden* muutoksia tietyllä mukautumistasolla on myös tärkeä ominaisuus. Klassinen koe, jolla silmän kykyä erotella eri kirkkausarvoja voidaan testata, tapahtuu siten, että koehenkilölle näytetään tasaisesti valaistua laajaa aluetta, joka täyttää koko näkökentän (esimerkiksi himmeät suojalasit, joita valaistaan ulkopäin). Valonlähteen intensiteettiä I voidaan vaihdella. Tähän valaistukseen lisätään toinen, lyhyistä välähdyksistä koostuva, valonlähde ΔI keskelle näkökenttää. Jos ΔI ei ole tarpeeksi kirkas, koehenkilö vastaa "EI", eli hän ei havaitse valonvälähdyistä. Kun muutos ΔI käy voimakkaammaksi, hän jollain kirkkaustasolla antaa vastaukseksi "KYLLÄ". Riittävän suurella ΔI :n arvolla koehenkilö vastaa aina "KYLLÄ". Suhdetta $\Delta I_c/I$, missä ΔI_c on se lisäys kirkkaudessa, joka havaitaan puolessa kokeista, sanotaan *Weberin suhteeksi*. Pieni Weberin suhde tarkoittaa, että pieni prosentuaalinen muutos valon kirkkaudessa on havaittavissa, jolloin kirkkauden erotuskyky on hyvä. Vastaavasti suuri Weberin suhde tarkoittaa, että vaaditaan suuri suhteellinen kirkkauden muutos ja kirkkauden erotuskyky on huono. Kuvassa on esitetty $\log(\Delta I_c/I)$ $\log(I)$:n funktiona. Kuvasta nähdään, että kirkkauden erotuskyky on huono matalilla kirkkauden arvoilla (suuri Weberin suhde) ja paranee huomattavasti (suhde pienenee) kun taustan kirkkaus kasvaa. Käyrän kaksiahaarainen luonne on suoraa seurausta siitä, että pienillä kirkkauden arvoilla näkemisestä ovat vastuussa sauvat ja suurilla kirkkausarvoilla tapit.

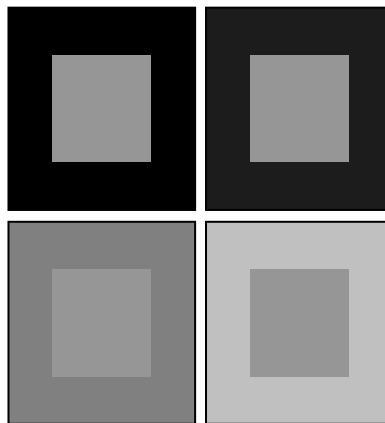
Jos taustan kirkkaus pidetään vakiona ja lisävalonlähteen kirkkautta kasvatetaan tasaisesti, voidaan havaita, että koehenkilö pystyy havaitsemaan 10 - 25 eri harmaasävyä. Tämä on karkeasti ottaen se määrä sävyjä, jonka ihminen pystyy havaitsemaan yhdellä kertaa harmaasävykuvassa. Tämä EI kuitenkaan tarkoita, että harmaasävykuva voitaisiin esittää näin pienellä sävyjen määrällä ! Kun silmä kohdistuu eri kohtaan kuvaa, sen mukautumistaso (B_a) muuttuu vastaavasti, jolloin saadaan aina uusi joukko harmaasävyjä, jotka kyetään erottelemaan. Tästä on seurauksena, että silmä pystyy käsittelemään paljon suuremman määrän harmaasävyjä.

Ihmisen silmän havaitsema kohteen kirkkaus ei ole valon intensiteetin (fotonien lukumäärä aikayksikössä) yksinkertainen funktio. Tämä tarkoittaa sitä, että aivot tekevät silmän havaitsemaan dataan omaa prosessointiaan, joka joissain tapauksessa saattaa vääristää havainnon ! Tästä kaksi esimerkkiä ovat *Machin vyökuvio* (Mach band pattern) ja *samanaikainen kontrasti* (simultaneous contrast).



Kuva x: Machin ryökuvio. Kuussa kaikki kahdeksan eri harmaasävyjuovaa ovat kukin vakiosävyisiä. Tummempaa juovaa vasten oleva sivu näyttää tummemmalta kuin vaaleampaa juovaa vasten oleva sivu !

Samanaikainen kontrasti tarkoittaa sitä, että alueen kirkkaus ei riipu ainoastaan alueen absoluuttisesta intensiteetistä, vaan myös *taustan* intensiteetistä. Esimerkiksi valkea paperi näyttää pöydällä lampun valossa valkoiselta, mutta mustalta jos sitä katsoo vasten kirkasta taivasta.



Kuva x. Samanaikainen kontrasti. Kaikissa pienissä neliöissä on sama harmaasävyarvo, mutta ne näyttävät sitä tummemmilta, mitä vaaleampi tausta on.

2.4 YKSINKERTAINEN KUVAMALLI

Tämän materiaalin alussa kuva määriteltiin kaksiulotteisena intensiteettifunktiona $f(x,y)$. Funktio f palauttaa arvonaan kuvan intensiteetin (kirkkauden) pisteessä (x,y) . Koska valo on fyysikaalisestikin energiaa, on oltava voimassa

$$0 < f(x,y) < \infty.$$

Ihminen havaitsee kohteen, koska valo heijastuu siitä. Funktio $f(x,y)$ voidaan jakaa valaistus- ja heijastuskomponenttiin, $i(x,y)$ ja $r(x,y)$. Valaistuskomponentti on itse asiassa luminanssi (yksikkö kandela/m²) ja jälkimmäinen kuvaa sitä, kuinka suuri osa valosta heijastuu kohteen pinnasta. Heijastuskomponentti voi saada arvoja väliltä 0 (täydellinen absorptio) - 1 (täydellinen heijastus). Valaistuskomponentti voi puolestaan saada arvoja väliltä $0 < i(x,y) < \infty$. Intensiteetti-, heijastus- ja valaistuskomponenttien välillä on yhteys

$$f(x,y) = i(x,y)r(x,y)$$

Kuvan intensiteetti tietyssä pisteessä määritellään *harmaasävynä* (gray level) l . Se voi saada arvoja *harmaasävyskaalalla* väliltä $[L_{min}, L_{max}]$ siten, että arvo $l=L_{min}$ vastaa täysin mustaa pikseliä ja $l=L_{max}$ täysin valkoista pikseliä. Näiden ääriarvojen väliin sijoittuvat l :n arvot ovat eri harmaan sävyjä.

Kuvan intensiteettifunktio $f(x,y)$ voidaan esittää $N \times M$ -matriisina:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix}$$

Yhtälön oikea puoli esittää digitaalista kuvaa. Kukin matriisin elementti on yksi pikseli. Pikselien arvot ovat kokonaislukuja, vaikka alkuperäisessä kuvainformaatiossa valon intensiteettijakauma on jatkuva. Siten se, kuinka hyvin digitaalinen kuva toistaa alkuperäisen kuvan, riippuu kuvamatriisin koosta (rivien N ja sarakkeiden M määrästä) sekä harmaasävyjen määrästä G . Yleensä sekä N , M että G ovat 2:n potensseja, eli

$$N=2^n, M=2^k \text{ ja } G=2^m.$$

Digitaalisen kuvan tallennukseen vaadittava bittimäärä $b = NMm$. Siten 64 harmaasävyn 128x128-kuvamatriisilla talletettu digitaalinen kuva vaatii 98 304 bittiä. Kuvan *resoluutiolla* tarkoitetaan sitä, kuinka pieniä yksityiskohtia kuvasta voidaan erottaa. Mitä suurempi matriisi ja mitä enemmän harmaasävyjä, sitä paremmin digitaalinen kuva muistuttaa alkuperäistä, mutta samalla tallennuskapasiteetin määrä valitettavasti kasvaa rajusti. "Hyvää" kuvanlaatua on hyvin vaikea määrittellä. Kauneus on tässäkin

tapauksessa katsojan silmässä. Harjoituksissa tutkitaan, miten kuvanlaatu muuttuu eri parametrien arvoilla.

On syytä huomioida, että joissakin tapauksissa subjektiivinen kuvanlaatu paranee, jos harmaasävyjen määrää *vähennetään*, koska kuvan kontrasti näyttää paranevan. Vastaavasti joskus kuvissa, joissa on suuri määrä yksityiskohtia, tarvitaan joskus ainoastaan muutama harmaasävy, eikä subjektiivinen kuvanlaatu parane sävyjen määrää lisäämällä.

3 PIKSELIN VÄLISET RELAATIOIT

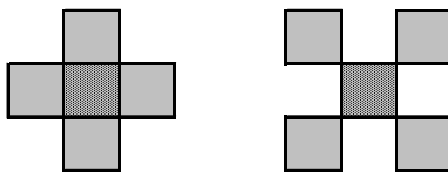
3.1 PIKSELIN NAAPURIT

Pisteessä (x,y) olevalla pikselillä p on neljä horisontaalista ja neljä vertikaalista naapuria, joiden koordinaatit ovat $(x+1,y)$, $(x-1,y)$, $(x,y+1)$ ja $(x,y-1)$.

Tätä pikselijoukkoa kutsutaan pikselin p *4-naapureiksi* ja merkitään $N_4(p)$. Kukin pikseleistä on yksikköetäisyyden (unit distance) päässä pisteestä (x,y) ja jotkut naapureista voivat olla kuva-alan ulkopuolella. Pikselin p *diagonaalinaapureilla* $N_D(p)$ on koordinaatit

$$(x+1, y+1), (x+1, y-1), (x-1, y+1) \text{ ja } (x-1, y-1).$$

Pisteitä, jotka kuuluvat joko joukkoon $N_4(p)$ tai joukkoon $N_D(p)$ kutsutaan pikselin p *8-naapureiksi* $N_8(p)$.



Kuva x: Pikselin p sekä sen 4- ja diagonaalinaapurit.

3.2 VIEREISYYS JA LIITTYVYYS

Jos kaksi pikseliä ovat *viereisiä*, on niiden oltava naapureita (esim. 4-naapureita) ja niiden harmaasävyjen on lisäksi oltava joko täsmälleen tai tietyissä rajoissa samat. Esimerkiksi binäärikuvassa

(vain sävyarvot 0 ja 1) kaksi pikseliä voivat olla 4-naapureita, mutta ne eivät ole viereisiä jos toisen arvo on 0 ja toisen 1.

Merkitään symbolilla V joukkoa harmaasävyarvoja, jotka määrittävät viereisyyden. Esimerkiksi binäärikuvalla $V=\{1\}$ määrittää viereisyyden niiden pikseleiden välillä, joiden arvo on 1. Vastaavasti, jos halutaan tutkia viereisyyttä harmaasävykuvalla väliltä 32 -> 64, $V=\{32, 33, \dots, 63, 64\}$.

Viereisyyden määritelmä:

1. 4-vierisyys: Pikselit p ja q , joiden arvot toteuttavat ehdon V , ovat 4-viereisiä jos q kuuluu joukkoon $N_4(p)$.
2. 8-viereisyys: Pikselit p ja q , joiden arvot toteuttavat ehdon V , ovat 8-viereisiä jos q kuuluu joukkoon $N_8(p)$.
3. m -viereisyys (sekoittunut viereisyys, mixed adjacency): Pikselit p ja q , joiden arvot toteuttavat ehdon V , ovat m -viereisiä, jos
 - q kuuluu joukkoon $N_4(p)$ tai
 - q kuuluu joukkoon $N_D(p)$ ja leikkausjoukko $N_4(p) \cap N_4(q)$ on tyhjä (tämä on pikselijoukko, joka on sekä p :n että q :n 4- naapureita ja jonka arvot toteuttavat ehdon V).

m -viereisyys on 8-viereisyyden täydennys. jolla eliminoidaan useat vaihtoehtoiset polut pikselien välillä.

Kaksi alijoukkoa $S1$ ja $S2$ ovat viereisiä, jos jokin alijoukon $S1$ pikseli on viereinen jonkin joukon $S2$ pikselin kanssa. *Polku* (path) pikselistä p , jonka koordinaatit ovat (x,y) , pikseliin q , jonka koordinaatit ovat (s,t) on järjestetty joukko peräkkäisiä pikseleitä, joiden koordinaatit ovat

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n),$$

missä $(x_0, y_0) = (x, y)$ ja $(x_n, y_n) = (s, t)$. Polulla oleva pikseli (x_i, y_i) on viereinen pikselille (x_{i-1}, y_{i-1}) , $1 \leq i \leq n$, missä n on *polun pituus*. Riippuen vierekkäisyyden tyypistä voidaan määrittellä 4-, 8-, tai m -*polkuja*. Jos p ja q kuuluvat alijoukkoon S , ne ovat liittyneitä, jos niiden välillä on olemassa polku, jonka kaikki pikselit kuuluvat joukkoon S . Nämä pikselit muodostavat *liittyneen komponentin* (connected component). Kaikki liittyneeseen komponenttiin kuuluvat pikselit ovat liittyneitä toisiinsa.

3.3 LIITTYNEIDEN KOMPONENTTIEN NIMEÄMINEN

Tarkastellaan tilannetta, jossa kuvaa käydään pikseli kerrallaan läpi vasemmalla oikealle ja ylhäältä alas. Tarkastelun kohteena on tässä ainoastaan 4-liittyneet komponentit. Merkitään p :llä tietyllä hetkellä tarkastelun kohteena olevaa pikseliä ja olkoot r ja t , jotka ovat p :n ylä- ja vasemmanpuoleiset naapurit. Edellä kiinnitetty kuvan läpikäyntijärjestys takaa, että pikseleissä r ja t on jo käyty, kun saavutaan p :hen.

Nimeämisalgoritmin kulku:

- Jos $p=0$, siirry seuraavaan pikseliin.
- Jos $p=1$, tutki r ja t .
 - Jos sekä r että t ovat nollia, merkitse p uudeksi komponentiksi (tämä on ensimmäinen kerta, kun tämä komponentti tulee vastaan !)
 - Jos joko $r=1$ tai $t=1$, nimeä sen tunnus myös p :n tunnukseksi.
 - Jos $r=1$ ja $t=1$, ja
 - jos niillä on sama tunnus, nimeä myös p kyseisellä tunnuksella.
 - jos niillä on eri tunnus, nimeä p jommankumman tunnuksen mukaan ja tee muistiinpano siitä, että r :n ja t :n tunnukset ovat ekvivalentteja (ts. pikselit r ja t ovat liittyneet p :hen).

Kun kaikki pikselit on läpikäyty, kaikki ne, joiden arvo oli 1, on tullut nimetyksi jollain tunnuksella, mutta tunnuksia saattaa olla useita. Jotkut tunnuksista ovat kuitenkin ekvivalentteja eli samanarvoisia (*ne kuuluvat samaan liittyneeseen komponenttiin* !), jolloin ne voidaan yhdistää yhdeksi tunnukseksi. Kaikki ekvivalentit tunnusparit lajitellaan ekvivalenssiluokkiin, niille annetaan uusi, yhteinen tunnus, jonka jälkeen käydään kuva uudelleen läpi ja nimetään ekvivalentit tunnukset uudella yhteisellä tunnuksella.

Jos 4-liittävyyden sijaan edellinen tarkastelu suoritetaan 8-liittävyydelle, täytyy lisäksi tutkia myös kahden ylädiagonaalinaapurien q ja s arvot. Jälleen, läpikäyntijärjestys takaa, että nämä naapurit on jo

käyty läpi ja mahdollisesti nimetty jollain tunnuksella ennen p :tä. Jätetään tämän algoritmin toteutus harjoitustehtäväksi ☺.

3.4 RELAA TIOT, EKVIVALENSSI JA TRANSITIIVINEN SULKEUMA

Joukkoon \mathbf{A} kohdistuva *binäärinen relaatio* \mathbf{R} määritellään arvoparien joukoksi \mathbf{A} :han kuuluvia elementtejä. Jos alkio a ja b kuuluvat joukkoon \mathbf{A} , niiden välistä binääristä relaatiota merkitään aRb . Tämä voidaan lukea muodossa "alkio a on yhteydessä alkioon b :hen".

Esimerkki: Tarkastellaan joukkoa $A = \{p_1, p_2, p_3, p_4\}$, joka on järjestetty seuraavasti:

$$\begin{array}{cc} p_1 & p_2 \\ & p_3 \\ & & p_4 \end{array}$$

Olkoon relaationa "4-liittyvyys". Tässä tapauksessa \mathbf{R} on niiden arvoparien muodostama joukko, jotka toteuttavat 4-liittyvyyden, eli $\mathbf{R} = \{(p_1, p_2), (p_2, p_1), (p_1, p_3), (p_3, p_1)\}$. Toisin sanoen p_1 on yhteydessä (relaatioissa) p_2 :een (nehän ovat 4-naapureita), samoin p_1 on yhteydessä p_3 :een ja päinvastoin, mutta p_4 ei ole yhteydessä mihinkään muuhun joukon \mathbf{A} alkioon. Tässä tilanteessa edellisen kappaleen nimeämisalgoritmi siis tuottaa kaksi tunnusta: yhden p_1, p_2 ja p_3 :n muodostamalle liittyneelle komponentille ja toisen p_4 :lle. □

Binäärinen relaatio \mathbf{R} joukkoon \mathbf{A} sanotaan olevan

- Refleksiivinen, jos jokaisella alkioilla a joukossa \mathbf{A} , on voimassa aRa ;
- Symmetrinen, jos jokaiselle alkioille a ja b joukossa \mathbf{A} , $aRb = bRa$;
- Transitiiivinen, jos on voimassa aRb ja bRc niin aRc .

Relaatio, joka täyttää nämä kolme ominaisuutta sanotaan olevan *ekvivalenssirelaatio*. Ekvivalenssirelaation tärkeä ominaisuus on, että jos \mathbf{R} on joukon \mathbf{A} ekvivalenssirelaatio, joukko \mathbf{A} voidaan jakaa k :hon toisiinsa liittymättömään alijoukkoon (komponenttiin). Näitä alijoukkoja kutsutaan *ekvivalenssiluokiksi*. Kaksi alkioita a ja b kuuluvat samaan alijoukkoon (komponenttiin) vain, mikäli ne kuuluvat samaan ekvivalenssiluokkaan.

Esimerkki: Relatio $R = \{(a,a), (a,b), (b,d), (d,b), (c,e)\}$ voidaan esittää binäärimatriisina,

$$\mathbf{B} = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

missä matriisin alkioksi on asetettu 1, jos riveiltä ja sarakeilta luetut alkiot ovat yhteydessä keskenään. Jos esitetty relaatio olisi refleksiivinen, kaikki diagonaalelementit olisivat ykkösiä. Jos relaatio olisi symmetrinen, \mathbf{B} olisi symmetrinen matriisi. \square

Transitiivisuus tarkoittaa sitä, että jos aRb ja bRc niin silloin myös aRc . Edellisessä esimerkissä a on yhteydessä b :hen ja b on yhteydessä d :hen, koska (a,b) ja (b,d) kuuluvat joukkoon \mathbf{R} . Nyt kuitenkin huomataan, että relaatio (a,d) **ei kuulu** \mathbf{R} :ään. Joukkoa, joka sisältää myös nämä puuttuvat relaatiot, sanotaan \mathbf{R} :n *transitiiviseksi sulkeumaksi* (transitive closure) ja sitä merkitään \mathbf{R}^+ . Tässä $\mathbf{R}^+ = \{(a,a), (a,b), (a,d), (b,b), (b,d), (d,b), (d,d), (c,e)\}$. Se, miksi myös (b,b) ja (d,d) kuuluvat \mathbf{R}^+ :een, johtuu transitiivisuuden määritelmästä. Koska bRd ja dRb , niin bRb ja vastaavasti koska dRb ja bRd niin dRd . Matriisimuodossa,

$$\mathbf{B}^+ = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

missä ykkösellä merkityt alkiot ovat transitiivisen sulkeuman jäseniä. Matriisi \mathbf{B}^+ voidaan määrittää seuraavasti. Olkoon \mathbf{B} $n \times n$ binäärimatriisi joka esittää relaatioita \mathbf{R} joukosta \mathbf{A} . Matriisi \mathbf{B}^+ lasketaan kaavalla

$$\mathbf{B}^+ = \mathbf{B} + \mathbf{B} \mathbf{B} + \mathbf{B} \mathbf{B} \mathbf{B} + \dots + (\mathbf{B})^n,$$

missä $(\mathbf{B})^n = \mathbf{B} \mathbf{B} \mathbf{B} \dots \mathbf{B}$ (n kertaa). Matriisitulon laskeminen tapahtuu kuten matematiikassa on (toivottavasti) opittu, paitsi että kertolasku korvataan JA-operaatiolla ja yhteenlasku TAI-operaatiolla. Esimerkiksi Matlab laskee sen näppärästi myös. Laskenta on luonteeltaan rekursiivinen, eli kussakin vaiheessa saatu tulosmatriisi kerrotaan seuraavalla kierroksella \mathbf{B} :llä. Jätetään tämäkin harjoitustehtäväksi.

Edellä kuvattu menetelmä on laskennallisesti kallis, sillä se vaatii n^3 laskutoimitusta. Warshall on kehittänyt tehokkaamman algoritmin, joka vaatii ainoastaan TAI-operaatioita niiden matriisielementtien välillä, joiden arvo on 1. Ohessa algoritmin toteutus pseudokoodina:

Step 1: set $j=1$.

Step 2: for $i=1, 2, \dots, n$, if $b(i, j)=1$ then, for $k=1, 2, \dots, n$

set $b(i, k) = b(i, k) + b(j, k)$.

Step 3: $j=j+1$.

Step 4: if $j \leq n$ goto Step 2, otherwise goto Step 5.

Step 5: Stop. The result is \mathbf{B}^+ in place of \mathbf{B} .

Käytännössä aina oletetaan, että relaatiot ovat ekvivalenssirelaatioita, jolloin matriisi \mathbf{B} on symmetrinen ja sen diagonaalelementit ovat ykkösiä.

3.5 ETÄISYYSMITAT

D on pikselien p, q , ja z , joiden koordinaatit ovat (x, y) , (s, t) , ja (u, v) *etäisyysfunktio* (metric), jos

- $D(p, q) \geq 0$ (HUOM: $D(p, q)=0$ jos ja vain jos $p=q$).
- $D(p, q) = D(q, p)$ ja
- $D(p, z) \leq D(p, q) + D(q, z)$

Pikselien p ja q välinen *Euklidinen etäisyys* määritellään kaavalla

$$D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Tämähän on "normaali" geometriasta tuttu kahden pisteen välinen etäisyys.

D_4 -etäisyys ("city-block distance") määritellään kaavalla

$$D_4(p, q) = |x - s| + |y - t|$$

Tässä tapauksessa pikselit, jotka ovat etäisyyden r säteellä keskipisteestä (x, y) , muodostavat timanttikuvion.

Esimerkki: Pikselit, joille $D_4 \leq 2$ (lukuarvo kertoo etäisyyden):

$$\begin{array}{ccccc} & & & & 2 \\ & & & & 2 & 1 & 2 \\ & & & & 2 & 1 & 0 & 1 & 2 \\ & & & & 2 & 1 & 2 \\ & & & & 2 \end{array}$$

Selvästikin pikselit, joille $D_4=1$ ovat keskipikselin 4-naapureita.

D_8 -etäisyys ("chessboard distance") määritellään kaavalla:

$$D_8(p, q) = \max(|x - s|, |y - t|)$$

Pikselit, jotka ovat D_8 -etäisyydellä keskipikselistä (x, y) muodostavat neliön. Pikselit, jotka ovat etäisyydellä $D_8=1$, muodostavat keskipikselin 8-naapuruston.

Esimerkki: Pikselit, joille $D_8 \leq 2$:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

D_4 -etäisyys on lyhin kahden pikselin välinen *4-polku*. Vastaavasti D_8 -etäisyys on lyhin pikselien välinen *8-polku*. Etäisyys voidaan määrittää pikseleiden välille, vaikka ne eivät olisikaan liittyneitä komponentteja. M-liittyvyydelle pikselien välinen etäisyys sen sijaan riippuu pikselien ja niiden naapurien arvoista.

Esimerkki: Tarkastellaan oheisen kuvan mukaista pikselijoukkoa:

$$\begin{array}{cc} & p3 & p4 \\ p1 & & p2 \\ p & & \end{array}$$

Oletetaan, että p_1 , p_2 ja p_4 arvo on 1 ja että p_3 ja p_4 voivat saada joko arvon 0 tai 1. Jos sallitaan ainoastaan 1-arvoisten pikselien liittyvyys, ja $p_1=p_3=0$, pikselien p ja p_4 välinen m-etäisyys on 2. Jos joko $p_1=1$ tai $p_3=1$, etäisyys on 3. Jos molemmat ovat 1, etäisyys on 4. Toisin sanoen, jos vaaditaan m-liittyvyys, polku liittyneiden pikselien välillä kuljetaan ylä-ala-sivusuunnassa eikä diagonaaleja pitkin!

3.6 ARITMEETTIS-LOGISET OPERAATIOIOT

Aritmeettiset operaatiot:

- Yhteenlasku: $p+q$
- Vähennyslasku: $p-q$
- Kertolasku: $p*q$ (muuta merkintätapoja pq , $p \times q$)
- Jakolasku: $p \div q$

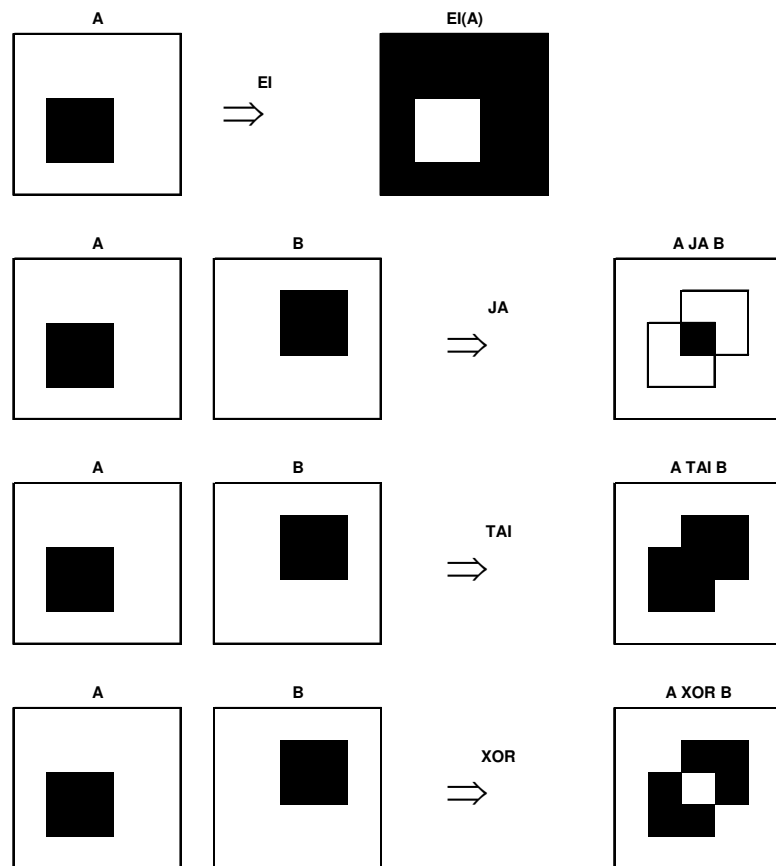
Aritmeettiset operaatiot suoritetaan pikseli pikseliltä ja yleensä koko kuvalle. Yhteenlaskua käytetään mm. kohinanpoistoon (keskiarvoistus). Vähennyslaskua käytetään mm. digitaalisessa subtraktioangiografiassa staattisen taustakuvan poistoon, ts. vähennetään ennen ja jälkeen varjoainetta otetut kuvat toisistaan. Kerto- ja jakolaskua voidaan käyttää esimerkiksi korjaamaan kuvan intensiteettivaihteluita. Tällainen tilanne syntyy usein magneettikuvauksessa, jossa lähellä kelaa olevasta kohteesta saadaan vahvempi signaali kuin kauempana olevasta.

Loogiset operaatiot:

- JA (AND): Tosi, jos p on tosi JA q on tosi ($p \bullet q$).
- TAI (OR): Tosi, jos p on tosi TAI q on tosi ($p + q$).
- KOMPLEMENTTI: EI q (\bar{q}).

Näistä perusoperaatioista saadaan muut loogiset operaatiot yhdistelemällä niitä tarpeen mukaan. Erityisen tärkeä lisä on **XOR**-(exclusice-OR)-funktio, joka palauttaa 1:n jos *joko* $p=1$ *tai* $q=1$ mutta *EI* molemmat.

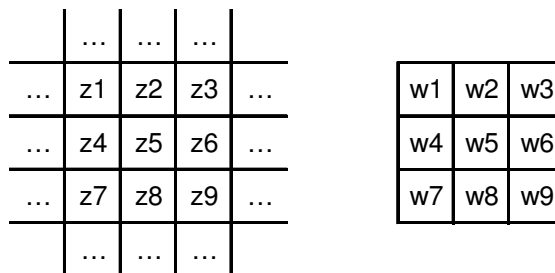
Loogisia operaatioita voidaan käyttää ainoastaan binäärisiin kuviin. Niitä käytetään perustyökaluina mm. maskauksessa, hahmontunnistuksessa ja muotoanalyysissä. Loogiset operaatiot suoritetaan pikseli pikseliltä kahden kuvan välillä ja tulos tallettuu jommankumman kuvan paikalle tai uudeksi kuvaksi. Esimerkiksi JA-operaation tulos on 1 vain, jos molemmissa alkuperäisissä kuvissa vastinpikselien arvot ovat = 1. Seuraavassa on esitetty muutamia esimerkkejä kuvien A ja B välisistä loogisista operaatioista, jossa kylläkin poikkeuksellisesti musta ruutu=1 ja valkea=0.



3.7 MASKIOPERAATIOIT

Pikseli pikseliltä prosessoinnin lisäksi aritmeettis-loogisia operaatioita käytetään hyvin paljon "naapurusto-orientoituneeseen" prosessointiin. Tämä tarkoittaa sitä, että tietyn laskutoimituksen tai operaation tulos riippuu paitsi itse pikselin arvosta, myös sitä ympäröivien pikselien arvoista. Maskista käytetään - hieman asiayhteydestä riippuen - myös signaalinkäsittelystä tutumpia termejä ikkuna (window) tai suodatin (filter). Englanninkielisessä kirjallisuudessa käytetään tässä yhteydessä paljon myös termiä *template*.

Esimerkki: Tarkastellaan oheisen kuvan mukaista tilannetta.



Halutaan korvata vasemmanpuoleisessa kuvassa pikselin z_5 arvo keskiarvolla, joka on laskettu ympäröivän 3×3 -alueen pikseliarvojen avulla. Tämähän voidaan tehdä siten, että lasketaan keskiarvo

$$z = \frac{1}{9}(z_1 + z_2 + \dots + z_9) = \frac{1}{9} \sum_{i=1}^9 z_i$$

ja sijoitetaan tulos z z_5 :n paikalle. Yleisempi tapa on käyttää oikeanpuoleisessa kuvassa esitettyä maskia. Keskiarvon laskenta tapahtuu nyt siten, että keskitetään maski z_5 :n päälle ja kerrotaan kukin maskin alle jäävä pikseli maskissa määritellyllä kertoimella w_i ($i=1 \dots 9$).

$$z = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

Jos asetetaan $w_i=1/9$, saadaan sama tulos kuin edellä. Maskin toimintaperiaatteen ymmärtäminen on tärkeää! Sopivalla maskin ja sen kertoimien valinnalla voidaan suorittaa lähes kaikki tavanomaiset kuvankäsittelyoperaatiot. On kuitenkin huomattava, että maskaus pikseli pikseliltä on laskennallisesti kallis operaatio. Esimerkiksi 3×3 -maski laskettuna 512×512 -matriisiin vaatii 9 kertolaskua ja 8 yhteenlaskua jokaista pikseliä kohti. Tästä syntyy kaikkiaan 2 359 296 kertolaskua ja 2 097 152 yhteenlaskua.

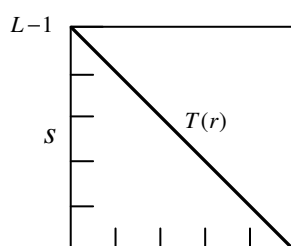
4 KUVAN EHOSTUS

Kuvan ehostuksella (image enhancement) pyritään prosessoimaan kuvaa siten, että siitä saadaan halutut piirteet ja / tai ominaisuudet esille. Se, mitä missäkin tilanteessa halutaan, riippuu sovellutuksesta. Tässä luvussa käsitellyt menetelmät voidaan jakaa kahteen luokkaan: Kuva- ja taajuustason menetelmiin. Kuvatasossa manipuloidaan suoraan itse kuvaa ja sen pikseleitä. Taajuustason menetelmät operoivat sen sijaan kuvan Fourier-muunnokseen. Varsin tavallista on, että sovelluksessa hyödynnetään näitä molempia. Tässä luvussa tarkastellaan ensin kuvatason pikseli kerrallaan (1×1 -

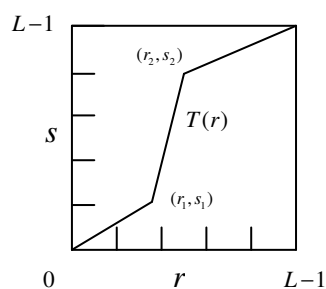
maski) tapahtuvaa prosessointia, sitten useamman pikselin yhteistuotoksena tapahtuvaa ($n \times n$ -maski) prosessointia ja lopuksi taajuustason menetelmiä.

4.1 PISTEPROSESSOINTI KUVATASOSSA

Yksipiste-prosessointi (single-point processing) on yksinkertainen menetelmä kuvan ehostukseen. Seuraavassa symbolilla r tarkoitetaan pikseli-intensiteettiä ennen jotain kuvankäsittelyoperaatiota, ja symbolilla s jälkeen kyseisen operaation. T on muunnosoperaattori, joka suorittaa kyseisen operaation. Pisteprosessoinnin tapauksessa T :tä kutsutaan *harmaasävymuunnokseksi* (gray-level transformation) $s=T(r)$.



Kuvan negatiivi: Kuvan negatiivi saadaan oheisen kuvan mukaisella transformaatiolla $s=T(r)$. Kuvassa L on harmaasävyn lukumäärä. Ideana on, että harmaasävyskala käännetään siten, että pienintä arvoa alkuperäisessä kuvassa vastaa suurin arvo prosessoidussa kuvassa ja päinvastoin.



Kontrastin venytys (contrast stretching): Laajennetaan alun perin liian "tasasävyn" kuvan kontrastia, jolloin saadaan enemmän yksityiskohtia esille. Pisteiden (r_1, s_1) ja (r_2, s_2) paikat kontrolloivat transformaatiofunktion muotoa. Jos $r_1=s_1$ ja $r_2=s_2$, kyseessä on *lineaarinen transformaatio*, joka ei muuta kontrastia mitenkään. Jos taas $r_1=r_2$, $s_1=0$ ja $s_2=L-1$, kyseessä on *kynnystäminen* (thresholding function), joka tuottaa binäärisen kuvan. Kaikki kynnyksen alapuolella olevat pikselit asetetaan nolliksi, ja kaikki yläpuoliset harmaasävyn maksimiarvoon. Näiden kahden ääritapauksen välissä saadaan erilaisia

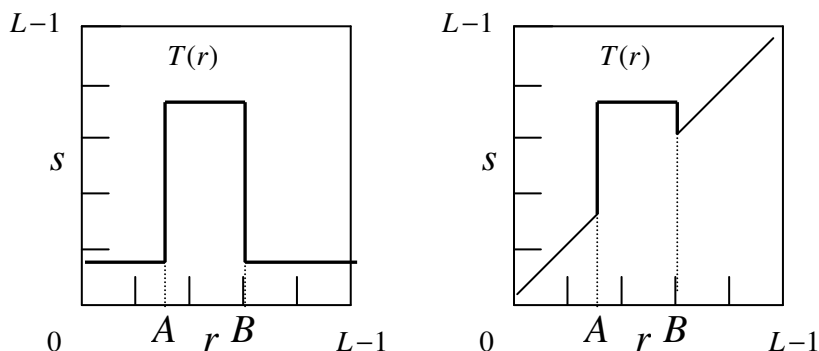
kontrasteja em. parametrien valinnoista riippuen. Yleensä kuitenkin tässä tehdään se oletus, että $r_1 \leq r_2$ ja $s_1 \leq s_2$, jolloin $T(r)$ on yksiarvoinen, monotonisesti kasvava funktio. Tämähän tarkoittaa vain sitä, että muunnoksessa harmaasävyjen järjestys mustasta valkoiseen säilyy, toisin kuin negatiivikuvan tapauksessa, jossa se kääntyy!

Dynaamisen harmaasävy skaalan kompressio (dynamic range compression): Joskus kuva sisältää paljon enemmän sävyjä, kuin siitä on mahdollista joko näyttää tai nähdä. Esimerkiksi kuvan Fourier-muunnosta tarkasteltaessa ainoastaan keskellä olevat matalan paikkataajuuden arvot ovat kirkkaita, muut osat näyttävät mustilta. Tehokas tapa skaalan kompressioon on suorittaa logaritmin avulla intensiteetitransformaatio:

$$s = c \log(1 + |r|),$$

missä c on skaalauskerroin.

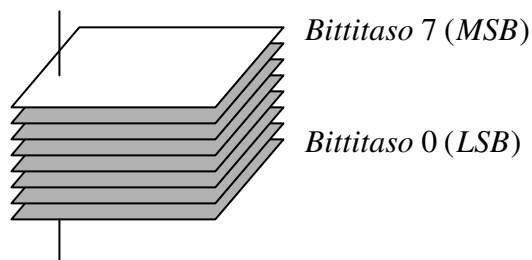
Harmaasävyjen viipalointi (Gray-level slicing): Joskus kuvasta pitää korostaa vain tiettyjä harmaasävyjä. Tähän on olemassa kaksi saman periaatteen variaatiota. (1) Tietyllä sävyvälillä olevat arvot asetetaan lähelle maksimiarvoa (kuvassa väli A - B) ja muut lähelle nollaa. Tuloksena on binäärinen kuva. (2) Korostetaan halutulla välillä olevia harmaasävyarvoja, mutta säilytetään kuvan kontrasti ja harmaasävyt alkuperäisinä tämän välin ulkopuolella.



Bittitason viipalointi (bit-level slicing): Oletetaan, että kukin kuvan pikseli on esitetty 8 bitillä. Jaetaan kuva kahdeksaksi 1.-bittiseksi kuvatasoksi siten, että tasossa 0 on kuvan vähiten merkitsevä bitti (LSB) ja tasossa 7 eniten merkitsevä bitti (MSB). Esimerkkikuvasta havaitaan, että viisi ylintä bittiä sisältää kuvan kannalta oleellista informaatiota ja alimmat kolme näyttävät lähinnä kohinalta. Lisäksi havaitaan,

että ylin bitti vastaa kynnystämistä raja-arvolla 128. Tämä johtuu tietenkin siitä, että suurin seitsemällä bitillä esitettävä luku on $2^7=128$.

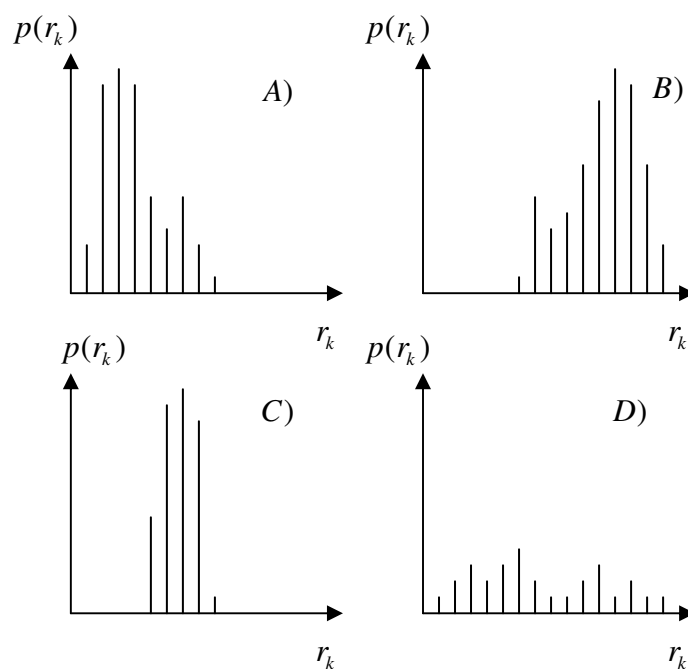
Yksi 8-bittinen tavu



4.2 HISTOGRAMMI JA SEN PROSESSOINTI

Digitaalisen kuvan, jonka harmaasävyt ovat välillä $[0, L-1]$, histogrammilla tarkoitetaan diskreettiä funktiota $p(r_k)=n_k/n$, missä r_k on k :s harmaasävy, n_k on niiden pikselien lukumäärä, joissa on kyseinen harmaasävy, n on pikselin kokonaismäärä ja $k=0, 1, 2, \dots, L-1$.

Funktio $p(r_k)$ toisin sanoen kuvaa tietyn harmaasävyn r_k esiintymistodennäköisyyttä kuvassa. Histogrammin muoto antaa hyödyllistä tietoa kuvan luonteesta. Oheisessa kuvassa on esitetty neljä histogrammin perustyyppiä. A)-kohdan histogrammista nähdään, että harmaasävyt ovat keskittyneet pieniin lukuarvoihin plotin vasempaan reunaan. Tästä voidaan heti päätellä, että kyseessä on varsin tummansävyinen kuva. Kuvassa B) tilanne on juuri päinvastainen. Kuvan C) histogrammin sävyt ovat keskittyneet suppealle harmaasävyvälille. Tällaisella kuvalla on *matala kontrasti* (vähän sävyjä). Puhutaan että kuvan dynaaminen alue on suppea. Sen sijaan histogrammin D) sävyt ovat jakautuneet koko harmaasävyskaalalle. Tällaisella kuvalla on laaja dynaaminen alue ja *korkea kontrasti*. On tietenkin syytä muistaa, että histogrammi ei itsessään kerro mitään siitä, mitä kuva esittää! Histogrammin muotoilu kuitenkin tarjoaa erittäin tehokkaan keinon kuvan kuvan kontrastin parantamiseen.



Histogrammiekvalisointi:

Esittäkään jatkuva muuttuja r kuvan normalisoituja harmaasävyjä siten, että $r=0$ vastaa mustaa ja $r=1$ valkoista pikseliä. Myöhemmin formulointi laajennetaan diskreettiin tapaukseen, jolloin pikseliarvot voivat olla välillä $[0, L-1]$. Tarkastellaan harmaasävymuunnosta $s=T(r)$, joka tuottaa yksikäsitteisen harmaasävyn s kullekin alkuperäiselle r :n arvolle. Oletetaan, että $T(r)$ on monotonisesti kasvava ja että sen arvot ovat välillä $[0,1]$. Näitten ehtojen ollessa voimassa, on myös olemassa *käänteismuunnos*

$$r=T^{-1}(s).$$

Sekä alkuperäisille että muunnetuille harmaasävyille voidaan määritellä todennäköisyystiheysfunktiot $p_r(r)$ ja $p_s(s)$, jotka kuvaavat sitä, millä todennäköisyydellä tietty sävy kuvassa esiintyy. Niiden välillä on yhteys

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

Muunnos voidaan esittää integraalina

$$s = T(r) = \int_0^r p_r(w)dw, \quad 0 \leq r \leq 1$$

missä w on mielivaltainen integroimismuuttuja. Yhtälön oikea puoli on r :n todennäköisyystiheyden kertymäfunktio. Derivoimalla s r :n suhteen saadaan $ds/dr = p_r(r)$. Kun se sijoitetaan $p_s(s)$:n kaavaan, saadaan

$$p_s(s) = \left[p_r(r) \frac{1}{p_r(r)} \right]_{r=T^{-1}(s)} = 1$$

Tämä on esittää todennäköisyystiheysjakaumaa, joka saa vakioarvon 1 välillä $0 \leq s \leq 1$. Muunnoksen jälkeen siis kaikki harmaasävyarvot s ovat yhtä todennäköisiä, eli on suoritettu alkuperäisten harmaasävyarvojen *histogrammiequalisointi*. Tulos on riippumaton käänteismuunnoksesta $T^{-1}(s)$. Ekvalisoinnin tuloksena saadaan kuva, jolla on alkuperäistä laajempi dynaaminen alue (suurempi kontrasti).

Esimerkki: Alkuperäiset pikseliarvot noudattavat todennäköisyysfunktioita. Ekvälisöinnin tekemiseksi riittää määrittää $T(r)$. Tässä tapauksessa se voidaan laskea varsin helposti analyttisesti:

$$p_r(r) = \begin{cases} -2r + 2, & 0 \leq r \leq 1 \\ 0, & \text{muualla} \end{cases}$$

$$\Rightarrow s = T(r) = \int_0^r (-2w + 2)dw = -r^2 + 2r.$$

Jätetään harjoitustehtäväksi osoittaa, että $p_s(s)$ on todellakin tasaisesti jakautunut. \square

Jotta tällä teorialla olisi käytännön merkitystä, sitä on pystyttävä käyttämään digitaalisena versiona. Kuvalle, jonka harmaasävyt saavat diskreettejä arvoja saadaan todennäköisyysfunktion kaavalla

$$p_r(r_k) = \frac{n_k}{n}, \quad 0 \leq r_k \leq 1 \quad \text{ja} \quad k = 0, 1, \dots, L-1$$

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j)$$

$$r_k = T^{-1}(s_k)$$

missä L on harmaasävyjen määrä, $p_r(r_k)$ on k :nnen harmaasävyyn todennäköisyys, n_k kertoo, kuinka monesti kyseinen sävy esiintyy kuvassa, n on pikselien lukumäärä.

4.3 KUVIEN VÄHENNYS

Kahden kuvan $f(x,y)$ ja $b(x,y)$ erotus määritellään kaavalla

$$g(x,y) = f(x,y) - b(x,y)$$

Erotus lasketaan kuvien vastinpikselien (joilla on samat koordinaatit) välillä. Kuten edellä jo todettiin, vähennyskuvien tärkeä sovellus on DSA. Tässä tapauksessa $b(x,y)$ on ns. *maskikuva*, joka on otettu kohteesta ennen varjoaineen injektointia. Samanaikaisesti varjoaineen ruiskutuksen kanssa otetaan sarja kuvia, jotka ovat $f(x,y)$. Varjoaine kulkeutuu pitkin verisuonia. Kun kukin $f(x,y)$ vähennetään maskikuvasta $b(x,y)$, kaikki muut kudokset paitsi suonet nollautuvat, koska varjoaine ei vaikuta niiden intensiteettiin. Käytännössä kuvauksessa käytetään kuvanvahvistin-TV-ketjua, joka pystyy tuottamaan vähennyskuvia niin nopeasti, että saadaan aikaan elokuvamainen kuvasarja.

4.4 KUVIEN KESKIARVOISTUS

Tarkastellaan kohinaista kuvaa $g(x,y)$. Sen voidaan ajatella koostuvan kohinasta $n(x,y)$ sekä alkuperäisestä kohinattomasta kuvasta $f(x,y)$.

$$g(x,y) = f(x,y) + n(x,y)$$

Kohinan keskiarvo on nolla ja se on luonteeltaan täysin satunnaista. Jos summataan M kappaletta samasta kohteesta otettua kohinaista kuvaa, niin tuloksena saatava kuva

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y)$$

lähestyy odotusarvonaan alkuperäistä (kohinatonta) kuvaa $f(x,y)$, kun M kasvaa:

$$E\{\bar{g}(x, y)\} = f(x, y)$$

Summakuvan *varianssi* voidaan esittää kohinan varianssin avulla:

$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{M} \sigma_{n(x,y)}^2$$

Keskiahjonta on varianssin neliöjuuri, eli

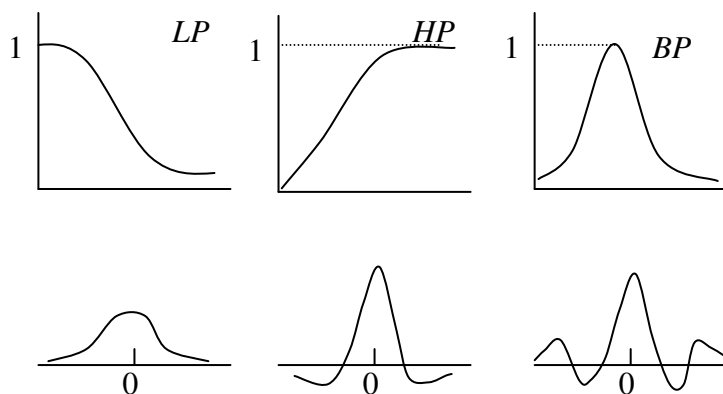
$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{n(x,y)}$$

Edellä olevista kaavoista nähdään, että kun M kasvaa, kuvapisteiden välinen hajonta pienenee. Siis mitä useampia kuvia summataan, sitä kohinattomampi, yhä paremmin alkuperäistä kuvaa muistuttava, summakuva saadaan. Käytännössä yksittäiset kuvat täytyy *rekisteröidä* ennen kuin ne voidaan summata, muuten tuloksena on summakuvan sumentuminen. Rekisteröinnillä kuvankäsittelyssä tarkoitetaan operaatiota, jolla eri kuvissa olevat kohteet siirretään (joko käsin tai automaattisesti) samalle kohtaa ennen summaamista.

4.5 SUODATUS KUVATASOSSA

Muissa kursseissa ollaan toivottavasti jo opittu, että lineaarisia suodattimia on neljää perustyyppiä: alipäästö-, ylipäästö-, kaistanpäästö- ja kaistanestosuodattimet. Alipäästösuodattimet vaimentavat korkeita taajuuksia ja päästävät matalat läpi. Kuvassa korkeat (paikka)taajuudet edustavat teräviä yksityiskohtia ja reunapintoja, joissa kontrasti muuttuu nopeasti. Vastaavasti matalat taajuudet kuvaavat hitaasti muuttuvia ominaisuuksia, kuten yleiskontrastia ja keskimääräistä intensiteettiä. Siten alipäästösuodattimella on aina kuvaa sumentava vaikutus. Kuvankäsittelyssä alipäästösuodattimia käytetään ensisijaisesti kuvan *tasoitukseen* (smoothing) ja ylipäästösuodattimia *reunankorostukseen* (edge enhancement). Kaistanpäästö- ja kaistanestosuodattimia käytetään etupäässä kuvan entisöinnissä (restoration). Siihen ei tällä kurssilla puututa.

Suodatin voidaan toteuttaa joko kuva- tai taajuustasossa. Oheisessa kuvassa on esitetty alipäästö (LP)-, ylipäästö(HP)- ja kaistanpäästösuodattimien (BP) poikkileikkaukset taajuus (ylärivi) - ja kuvatasoissa (alarivi). On huomattava, että kun näitä suodattimia sovelletaan kuvaan, ne ovat kaksiulotteisia.



4.5.1 ALIPÄÄSTÖSUODATTIMET (SMOOTHING FILTERS)

Keskiarvoistus: Suodatus voidaan tehdä laskemalla *keskiarvo* pikselin $n \times n$ -naapurustosta. Saatu arvo sijoitetaan alkuperäisen pikselin arvoksi, jonka jälkeen siirrytään laskemaan seuraavaa pikseliä. Näin käydään koko kuva läpi. Yksinkertaisin tällainen suodatin saadaan laskemalla pikselin 3×3 -naapurusto kaavalla

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

jossa kaikki maskin kertoimet w_i saavat arvon $1/9$ ja z_i ovat vastaavat pikseli-intensiteetit. On myös mahdollista määritellä suurempia maskeja, kuten 5×5 , 7×7 jne. Oleellista on, että maskin kertoimien summan on oltava 1, jolloin pikseliarvot pysyvät suodatuksen jälkeen alkuperäisellä skaalalla. 5×5 -maskin kertoimet ovat siis $1/25$ ja 7×7 maskin $1/49$. Lisäksi on huomattava, että kaikki maskin kertoimet ovat positiivisia. Näiden suodattimien ongelma on, että mitä suurempi maski, sitä enemmän kuva sumenee.

Mediaanisuodatus: Jos tarkastellaan jotain joukkoa lukuarvoja, mediaani m on luku, joka on lukujoukon "puolivälissä", ts. puolet joukon arvoista on suurempia kuin m ja puolet pienempiä. Mediaanisuodatusta kannattaa käyttää, jos halutaan poistaa kuvasta kohinaa *ilman* että se sumenee terävistä reunoista. Keskiarvon sijaan kunkin pikselin arvo korvataan mediaanilla. Menetelmä on erityisen tehokas, jos kuvassa on voimakkaita, teräviä kohinapiikkejä, joiden intensiteetti poikkeaa paljon ympäristön intensiteetistä.

Käytännössä mediaanisuodatus tapahtuu siten, että pikselin $n \times n$ -ympäristön pikseliarvot lajitellaan suuruusjärjestykseen, määritetään "keskimmäinen" arvo ja sijoitetaan se tarkasteltavana olevan pikselin arvoksi.

Esimerkki: Ohessa on esitetty kehystetyn pikselin 3×3 -ympäristö. Lajitellut pikseliarvot saadaan

10	20	20
20	15	20
20	25	100

järjestykseen $\{10, 15, 20, 20, \mathbf{20}, 20, 20, 25, 100\}$. Keskimmäinen arvo on 3×3 -ympäristön tapauksessa joukon 5. arvo (4 suurempaa ja 4 pienempää), joka on lihavoitu yllä. Tässä tapauksessa mediaani on siis 20, joka sijoitetaan kehystetyn pikselin paikalle. Jätetään harjoitustehtäväksi tutkia keskiarvoistuksen ja mediaanisuodatuksen ominaisuuksia eri maskin koolla. □

4.5.2 YLIPÄÄSTÖSUODATTIMET (SHARPENING FILTERS)

Maskin keskellä olevat kertoimet ovat positiivisia ja reunoilla olevat negatiivisia. Yksinkertaisin tällainen maski on siten oheisen kuvan kaltainen:

$$\begin{array}{ccc} & -1 & -1 & -1 \\ 1/9 \times & -1 & 8 & -1 \\ & -1 & -1 & -1 \end{array}$$

Huomaa, että maskin kertoimien summa on nolla. Maski ei siis muuta kuvan intensiteettiä sellaisilla alueilla, joissa on matalia paikkataajuuksia, ts. hitaasti paikan mukana muuttuvia harmaasävyjä. Tämä suodatin eliminoi kuvasta nollataajuisen (DC) komponentin. Siten suodatin hävittää kuvasta kontrastin lähes kokonaan. Tuloksena on odotettavissa teräviä piirteitä lähes mustalla taustalla. Lisäksi on huomattava, että koska maskin kertoimien summa on nolla, on tuloskuvan keskimääräinen intensiteetti nolla ! Tämähän tarkoittaa, että kuvassa on *negatiivisia* pikseliarvoja ! Ne saattavat aiheuttaa mielenkiintoisia efektejä kuvaan, koska tietokoneen binääriaritmetiikka tulkitsee pienen negatiivisen luvun suureksi positiiviseksi luvuksi. Siksi joudutaankin tapauskohtaisesti päättämään, miten negatiiviset pikseliarvot käsitellään.

Korkeataajuuspainotteinen suodatus (High-boost filtering): Ylipäästösuodatettu (HP) kuva voidaan laskea alkuperäisen kuvan (O) ja alipäästösuodatetun (LP) kuvan erotuksena: $HP=O-LP$. Jos alkuperäinen kuva kerrotaan jollain vahvistuskertoimella $A>1$, saadaan suodatin, jossa osa alkuperäisen kuvan intensiteetistä lisätään "takaisin" ylipäästösuodatettuun kuvaan. Näin voidaan palauttaa takaisin osa siitä kontrastista, joka suodatuksessa menetettiin:

$$HB=A \cdot O-LP=(A-1) \cdot O+O-LP=(A-1) \cdot O+HP$$

Tuloksena saadaan korkeataajuuspainotteinen "high-boost"-suodatin (HB). Reunankorostuksen suuruus riippuu A:sta. Mitä suurempi A, sitä enemmän korostusta. $A=1$ vastaa standardia edellä käsiteltyä ylipäästösuodatinta. Yleisesti operaatiota, jossa alkuperäisestä vähennetään sumennettu kuva, sanotaan *epäteräväksi maskaukseksi (unsharp masking)*. Tällä menetelmällä on keskeinen merkitys paino- ja julkaisuteollisuudessa sekä digitaalisessa radiografiassa. Tutkitaan harjoituksissa näitä tarkemmin.

4.5.3 DERIVAATTASUODATTIMET (DERIVATIVE FILTERS)

Funktion $f(x,y)$ *gradientti* määritellään 2D-tapauksessa vektorina

$$\nabla \bar{f} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Vektorin itseisarvo

$$|\nabla \bar{f}| = \nabla f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Tarkastellaan oheisen kuvan mukaista maskia, missä z_i :t ovat kuvan harmaasävyjä:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Gradientti voidaan estimoida pikselille z_5 monella eri tavalla. Yksinkertaisin tapa on laskea erotus $(z_5 - z_8)$ x-suunnassa ja erotus $(z_5 - z_6)$ y-suunnassa, jolloin

$$\nabla f \approx \sqrt{(z_5 - z_8)^2 + (z_5 - z_6)^2}$$

Neliöiden ja juurien sijasta voidaan käyttää myös itseisarvoja

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6|$$

Eräs tapa on käyttää "ristierotuksia"

$$\nabla f \approx \sqrt{(z_5 - z_9)^2 + (z_6 - z_8)^2}$$

Tai itseisarvoilla

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8|$$

Edellä kuvatut maskit voidaan implementoida 2x2-kokoisina. Esimerkiksi ristierotukset voidaan laskea käyttäen oheisia maskeja ja summaamalla niiden antamat tulokset yhteen. Näitä maskeja kutsutaan *Robertsin ristigradientioperaattoreiksi*.

$$\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & -1 & -1 & 0 \end{array}$$

Roberts

2x2-maskin implementointi on ohjelmallisesti hankalaa. Helpompaa on approksimoida gradientti pisteessä z_i käyttäen 3x3-ympäristöä esimerkiksi seuraavasti:

$$\nabla f = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_{24} + z_7)|$$

Siis 1. ja 3. rivin erotus approksimoi x:n suuntaista derivaattaa ja 3. ja 1. sarakkeen erotus vastaavasti derivaattaa y-suuntaan. Näitä oheisen kuvan mukaisia maskeja kutsutaan *Prewittin operaattoreiksi*.

$$\begin{array}{cccccc} -1 & -1 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 & 0 & 1 \end{array}$$

Prewitt

Gradientti voidaan approksimoida myös *Sobelin operaattoreilla*:

$$\begin{array}{cccccc} -1 & -2 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -2 & 0 & 2 \\ 1 & 2 & 1 & -1 & 0 & 1 \end{array}$$

Sobel

Huomaa, että kaikkien näiden maskien summa on nolla, jos niillä operoidaan alueeseen, jonka intensiteetti on vakio. Matematiikasta toivottavasti muistetaan, että niinhän vakion derivaatan toki pitää ollakin ...

4.6 KUVAN EHOSTUS TAAJUUSTASOSSA

4.6.1 KAKSIULOTTEINEN FOURIER-MUUNNOS

Tässä materiaalissa ei perehdytä Fourier-muunnoksen sielunelämän perusteisiin. Se oletetaan tunnetuksi aikaisemmilta kursseilta ☺. Kuvatasossa operoidaan kaksiulotteisella Fourier-muunnoksella: Jos $f(x,y)$ on jatkuva ja integroitava ja $F(u,v)$ on integroitava, ne muodostavat Fourier-muunnosparin

$$\begin{cases} \mathfrak{F}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \exp(-j2\pi(ux+vy)) dx dy \\ \mathfrak{F}^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v) \exp(j2\pi(ux+vy)) du dv \end{cases}$$

missä u ja v ovat taajuustason muuttujia. Kuten yksiulotteisessa tapauksessakin, myös nyt voidaan määrittää muunnokselle itseisarvo, vaihe ja teho kaavoilla

$$\begin{aligned} |F(u,v)| &= \sqrt{R^2(u,v) + I^2(u,v)} \\ \Phi(u,v) &= \arctan \left[\frac{I(u,v)}{R(u,v)} \right] \\ P(u,v) &= |F(u,v)|^2 = R^2(u,v) + I^2(u,v) \end{aligned}$$

Yksiulotteisessa tapauksessa diskreetti Fourier-muunnos saadaan ottamalla jatkuvasta funktiosta $f(x)$ N kappaletta näytteitä Δx :n välein:

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + [N-1]\Delta x)\}$$

Näytteistys on helposti laajennettavissa myös kaksiulotteiseen tapaukseen, jolloin saadaan diskreetti Fourier-muunnospari:

$$\begin{cases} F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux/M + vy/N)] \\ f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[j2\pi(ux/M + vy/N)] \end{cases}$$

missä $u=0, 1, 2, \dots, M-1$ ja $v=0, 1, 2, \dots, N-1$. Tässä M ja N ovat siis kuvamatriisin rivien ja sarakkeiden lukumäärä. Kaavoista nähdään, että ne ovat samat sekä taajuustasossa $F(u, v)$ että kuvatasossa $f(x, y)$. Näytteenottovälit Δx ja Δy vastaavat kuvan pikselikokoa ja ne voidaan ilmaista normaalisti pituuden yksiköissä, esimerkiksi millimetreinä. Pikselikoot kuva- ja taajuustasossa ovat yhteydessä toisiinsa kaavoilla

$$\Delta u = \frac{1}{M\Delta x}, \quad \Delta v = \frac{1}{N\Delta y}$$

Tällä kurssilla ei ole tarkoitus laskea Fourier-muunnoksia käsin, vaan käytetään apuna tietokonetta, joka osaa laskea nopean Fourier-muunnoksen eli FFT:n, vaikka käyttäjä ei välttämättä osaisikaan. On se viksu vekotin. Harjoituksissa tutustutaan tarkemmin siihen, miltä kuvan Fourier-muunnos näyttää. Fourier-muunnoksen käyttö kuvan ehostukseen on yksinkertaista, kunhan käytössä on tarvittavat ohjelmointityökalut, esimerkiksi Matlab. Ehostus tapahtuu kolmessa vaiheessa. Kuvasta 1) lasketaan FFT, 2) kerrotaan se suodattimen taajuusvasteella ja 3) otetaan käänteismuunnos.

4.6.2 KONVOLUUTIOOTEOREEMA

Taajuustason suodatuksen peruskivi on *konvoluutioteoreema*: Jos kuva $f(x, y)$ konvoloidaan paikkainvariantilla operaattorilla $h(x, y)$, saadaan tuloksena kuva $g(x, y)$ siten että

$$g(x, y) = h(x, y) * f(x, y),$$

missä $*$ viittaa konvoluutio-operaatioon. Konvoluutioteoreeman mukaan näiden funktioiden Fourier-muunnokset voidaan esittää tulona

$$G(u, v) = H(u, v)F(u, v),$$

missä G , H ja F ovat kuvataso-funktioiden g , b ja f Fourier-muunnokset. Huomaa, että jälkimmäisessä tapauksessa H ja F kerrotaan keskenään. *Siis konvoluution Fourier-muunnos on alkuperäisten funktioiden Fourier-muunnosten tulo.* Mitä järkeä tässä on? Monesti on paljon nopeampi ja helpompi laskea Fourier-muunnokset ja kertoa ne keskenään, kuin laskea sama asia konvoluution kautta.

Kuvankäsittelyssä voidaan ajatella, että $f(x,y)$ esittää "todellista", häiriintymätöntä kohdetta. Saatu kuva on kuitenkin vääristynyt monestakin eri tekijästä, kuten kuvauslaitteen tai kameran epätarkkuuksista johtuen. Kaikki nämä kuvanlaatua heikentävät tekijät kootaan funktion $b(x,y)$ alle ja sitä kutsutaan *pisteenlevenemäfunktioksi* (Point Spread Function, PSF). Lopputuloksena saadaan kuva $g(x,y)$, jossa *kukin alkuperäisen kuvan pikseli on vääristynyt PSF:n vaikutuksesta.* Tätähän konvoluutio nimenomaan tarkoittaa! Funktiota $H(u,v)$ kutsutaan (tässä yhteydessä) systeemin optiseksi siirtofunktioksi ja sen itseisarvoa *modulaationsiirtofunktioksi* (Modulation Transfer Function, MTF). Lineaaristen systeemien teoriassa PSF:stä käytetään nimitystä impulssivaste.

4.6.3 ALIPÄÄSTÖSUODATTIMET

Edellä maskien yhteydessä jo mainittiin, että reunat ja terävät siirtymät harmaasävystä toiseen, kuten kohina, vaikuttavat taajuusalueessa korkeisiin taajuuskomponentteihin. Siten kuvan harmaasävytasoisuus voidaan tehdä vaimentamalla korkeat taajuudet. Tehtävänä on valita siirtofunktio $H(u,v)$ siten, että alkuperäisen kuvan Fourier-muunnoksesta $F(u,v)$ on poistettu korkeat taajuudet. Kertomalla nämä keskenään saadaan $G(u,v)$ ja siitä edelleen haluttu suodatettu kuva $g(x,y)$ Fourier-käänteismuunnoksella. Seuraavaksi tarkastellaan suodattimia, jotka vaikuttavat muunnoksen reaali- ja imaginääriosiin samalla lailla. Tällaisia suodattimia kutsutaan *nollavaihesiirtosuodattimiksi* (zero-phase-shift filters).

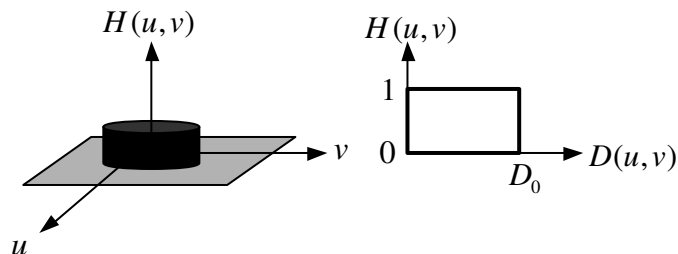
Ideaalinen suodatin: Siirtofunktio on muotoa

$$H(u, v) = \begin{cases} 1, & \text{jos } D(u, v) \leq D_0 \\ 0, & \text{jos } D(u, v) > D_0 \end{cases}$$

missä D_0 on (u,v) -tasossa määritelty säde (rajataajuus) ja $D(u,v)$ on taajuustason origon etäisyys pisteestä (u,v)

$$D(u, v) = \sqrt{u^2 + v^2}$$

Oheinen 3D-perspektiivikuva esittää $H(u,v)$:n käyttäytymisen. Kuten nimitys ideaalinen osoittaa, tämä suodatin päästää läpi kaikki taajuudet, jotka ovat D_0 -säteisen ympyrän sisäpuolella ja vaimentaa kaikki ulkopuolella olevat täysin.

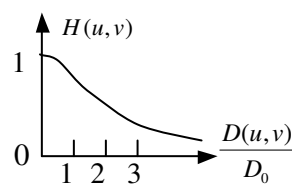


Kaikki jatkossa tarkasteltavat suodattimet ovat radiaalisymmetrisiä. Siksi riittää tarkastella ainoastaan projektiota johonkin suuntaan, kuten oikeanpuoleisessa kuvassa. Tämän tyyppinen oletus edellyttää, että nollataajuus on keskitetty keskelle kuvaa. Näin ei välttämättä ole automaattisesti. Monet ohjelmat laskevat FFT:n siten, että nollakohta on vasemmassa yläreunassa.

Ylläolevan kuvan kaltainen porraskuva taajuustasossa synnyttää sinc-funktion kuvatasoon. Kahdessa ulottuvuudessa sinc-funktio näyttää samalta, kuin aaltorintamat vedessä. Tämä ns. Gibbsin ilmiö eli rengastuminen (ringing) on haitallinen ja sitä voidaan eliminoida käyttämällä loivemmin muuttuvaa $H(u,v)$:ta.

Butterworth-suodatin: Suodattimen siirtofunktio $H(u,v)$ saadaan allaolevasta kaavasta. Siirtofunktion taajuusvasteen poikkileikkauskuvaa on esitetty oheisessa kuvassa.

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0} \right]^{2n}}$$

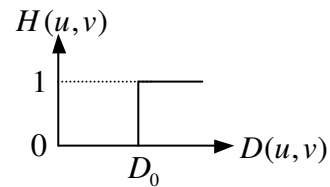


Tällä suodattimella ei ole terävää katkaisutaajuutta, joten myöskään rengastumista ei havaita. Haittapuolena on, että alipäästö ei ole kovin jyrkkä; suodatin päästää läpi myös katkaisutaajuuden yläpuolisia taajuuksia. Tavallisesti rajataajuus määritetään -3dB:n rajojen avulla. Rajataajuushan on se taajuus, jolla vaste on pienentynyt kertoimella 0.7 maksimiarvosta.

4.6.4 YLIPÄÄSTÖSUODATTIMET

Reunankorostus voidaan saada aikaan vahvistamalla korkeita paikkataajuuksia. Idealiselle ylipäästösuodattimelle määritellään siirtofunktio oheisella kaavalla ja sen taajuusvaste näyttää seuraavanlaiselta:

$$H(u, v) = \begin{cases} 0, & \text{jos } D(u, v) \leq D_0 \\ 1, & \text{jos } D(u, v) > D_0 \end{cases}$$



Rengastuminen on samanlaista kuin alipäästösuodattimellakin. Butterworth-ylipäästösuodatin määritellään kaavalla

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}}$$

