Konenäkö ja kuva-analyysi

Tuomo Rossi
Jyväskylän yliopisto
Tietotekniikan laitos

9. syyskuuta 2008

Sisältö

Luku 1

Matemaattisia esitietoja

1.1 Lineaariset suotimet ja konvoluutio

Useita tärkeitä kuvamuunnoksia voidaan tehdä saman yksinkertaisen mallin avulla. Aluksi konstruoidaan uusi taulukko, joka on samankokoinen kuin käsiteltävä pikselikuva. Uuden taulukon alkion arvoksi asetetaan painotettu summa käsiteltävän kuvan vastinpikselinaapuruston pikseliarvoista. Tämän naapuruston määrää *suodinmaskin* muoto ja koko. Yleensä maskin muoto on neliö ja sen koko on $(2k+1) \times (2k+1)$ pikseliä, missä $k = 0, 1, \ldots$ on tarkoitukseen soveltuva kokonaislukuvakio. Mielikuvana voi ajatella suodinmaskia liikuteltavan pikseli pikseliltä yli käsiteltävän kuvan ja maskin keskipistettä vastaava tulospikselin arvo saadaan skaalaamalla maskia vastaavat pikseliarvot maskin painoilla ja summaamalla ne yhteen. Huomaa: Tässä tapauksessa käytetään koko ajan kaikkia pikselinaapurustoja käsiteltäessä *samoja painokertoimia*, jolloin prosessi on *lineaarinen*, eli lopputulos kahden alkuperäisen kuvan summaa käsiteltäessä on sama asia kuin jos summattaisiin erikseen käsiteltyjen kuvien lopputulokset. Lisäksi vakiokertoimella skaalatun alkuperäisen kuvan käsittelyn lopputulos saadaan yhtä hyvin skaalaamalla samalla vakiolla alkuperäisestä kuvasta saatu lopputulos.

Erilaisia painokerroinjoukkoja käyttäen saadaan tietenkin erilaisia kuvankäsittelyoperaatioita. Painokertoimet voivat esimerkiksi olla arvoltaan suuria lähellä suodinmaskin keskipistettä ja ne vaipuvat nopeasti nollaan etäännyttäessä keskipisteestä. Tällainen suodin voisi esimerkiksi mallittaa epäfokuksessa olevan linssijärjestelmän aiheuttamaa kuvan sumenemista. Painokertoimia ei ole rajoitettu positiivisiksi. Esimerkiksi erotusosamäärää vastaavat, suodinmaskin alueella positiivisesta negatiivisiksi muuttuvat painokertoimet mallittavat matemaattista gradienttioperaattoria, jota voi hyödyntää esimerkiksi haluttaessa korostaa käsiteltävästä kuvasta kohtia joissa intensiteettivaihtelu on suurta. Tällaisia sovelluksia ovat esimerkiksi erilaisten reunojen etsintä kuvasta.

Toinen tärkeä ominaisuus tällaisella mallilla on sen *siirtoinvarianttius*, joka johtuu siitä, että lopputulospikselin arvo ei riipu sen paikasta kuvassa, vaan sen käsiteltävän kuvan vastinpikselin naapurien arvoista. Jos tarkasteltava objekti liikkuu kuvasarjassa, liikkuu sen suodatettu lopputulema vastaavalla tavalla lopputuloskuvasarjassa. Tällaiset ominaisuudet omaavaa muunnosta kutsutaan *lineaariseksi suotimeksi*. Lineaaristen suotimien matemaattiseen tarkasteluun käytetään usein *konvoluutioksi* nimettyä matemaattista operaatiota.

1.1.1 Konvoluutio

Suodinmaskin painokertoimet määräävät suotimen *ytimen* (eng. kernel), ja suotimen soveltamista yllä kuvatulla tavalla nimetään yleensä konvoluutioksi. Syystä joka selviää myöhemmin on luontevaa kirjoittaa prosessi matemaattiseksi kaavaksi epäilmeisellä tavalla: Olkoon annettu suotimen ydin H. Ytimen H konvoluutio käsiteltävän kuvan F kanssa on lopputuloskuva R. Kuvan R i,j:nnen pikselin arvo saadaan kaavasta

$$R_{i,j} = \sum_{u,v} H_{i-u,j-v} F_{u,v}. \tag{1.1}$$

Tässä on tietoisesti vältetty kirjaamasta summausrajat. Sen sijaan oletetaan, että summausalue on riittävän laaja huomioimaan kaikki ytimen nollasta poikkeavat arvot.

1.1.2 Esimerkki: silottaminen keskiarvoistamalla

Kuvilla on tyypillisesti se ominaisuus, että pikselin väri on samankaltainen sen naapuripikselien värin kanssa. Joskus kuvaan on sitä otettaessa tullut jostakin syystä kohinaa kuten satunnaisia "kuolleita"pikseleitä, tai pieniä nollakeskiarvoisia satunnaislukuja on jostakin teknisestä syystä summautunut pikseliarvoihin. Tällaisissa tapauksissa on luontevaa yrittää vähentää kuvan häiriöitä korvaamalla jokaisen pikselin arvo sen naapuripikselien arvojen painotetulla keskiarvolla. Tätä prosessia kutsutaan kuvan *silottamiseksi*, engl. smoothing, tai *sumeuttamiseksi*, engl. blurring.

Ensimmäinen yritelmä silotusoperaation malliksi on käyttää normaalia aritmeettista pikseliarvojen keskiarvoa yli kiinnitetyn suodinmaskialueen. Voimme esimerkiksi laskea keskiarvon kaikista $2k+1 \times 2k+1$ pikselistä tarkastelupikselin ympäriltä. Käsiteltävälle kuvalle F tämä määrää lopputuloksen

$$R_{i,j} = \frac{1}{(2k+1)^2} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} F_{u,v}. \tag{1.2}$$

Tarkastelemalla huolellisesti summausrajat, huomataan että tämä operaatio todellakin vastaa kaavan (1.1) mukaista konvoluutiota, kunhan ytimen alkioiden arvoksi asetetaan vakio $1/(2k+1)^2$.

Tämä tapa on kuitenkin huono kuvan silottamiseen. Lopputulos ei vastaa epäfokuksessa olevan kameran ottamaa kuvaa. Syy on selvä. Oletetaan että lähtökuvan kaikki pikselit paitsi keskimmäinen on arvoltaan 0, ja keskimmäinen on arvoltaan 1. Painottamaton keskiarvosuodatus tuottaa lopputuloksen, jossa kuvan keskellä on harmaa neliö, mutta epäfokuksessa oleva kamera ei toimi näin. Kameran linssit ovat yleensä ympyränmuotoisia, joten halutun silotusprosessin tulisi muuttaa hyvin pieni kirkas valopiste ympyränmuotoiseksi sumeaksi läiskäksi joka on keskeltä kirkas ja vaimenee säteen suuntaisesti mustaksi. Tällaiseksi silotusytimeksi valitaan hyvin usein Gaussin ydin.

1.1.3 Esimerkki: silottaminen Gaussin ytimellä

Hyvä formaali matemaattinen malli edellä mainitulle "sumealle läiskälle"on symmetrinen Gaussin ydin

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{1.3}$$

Tässä σ on Gaussin ytimen keskihajonta ja yksiköt ovat pikselien välisiä etäisyyksiä. Vakiotermi takaa sen, että Gaussin ytimen integraali yli koko avaruuden $\mathbb{R}^2$ on yksi. Tällä ytimellä on haluttuja ominaisuuksia:

- Jos keskihajonta on hyvin pieni (pienempi kuin yksi pikseli), on silotusvaikutus hyvin vähäinen koska painokertoimet keskipistettä lukuun ottamatta ovat hyvin pieniä.

- Suuremmalla keskihajonnalla lähinaapuripisteet vaikuttava lopputulokseen suuremmalla painoarvolla, jolloin keskiarvoistus johtaa lähinaapurien samankaltaistumiseen. Tämä on hyvä estimaatti pikselin arvolle, ja häiriökohina poistuu suurelta osin kuvan lievän sumeutumisen kustannuksella.

- Hyvin suuri keskihajonta johtaa tilanteeseen, jossa suurin osa yksityiskohdista katoaa kohinan mukana.

Sovelluksia varten täytyy Gaussin ydin diskretoida. Tämä tapahtuu muodostamalla diskreetti ydin konstruoimalla $2k+1 \times 2k+1$ -taulukko, jonka arvot ovat

$$H_{i,j} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}\right). \tag{1.4}$$

Keskihajonnan σ valinnalla on vaikutusta tarvittavaan taulukon kokoon k. Jos σ on liian pieni, on käytännössä ainoastaan taulukon keskimmäinen alkio nollasta poikkeava. Jos σ puolestaan on suuri, täytyy koon k olla riittävän suuri, jottei vielä kohtuullisen suuria painokertoimia leikkautuisi pois.

1.2 Siirtoinvariantit lineaariset systeemit

Useimmat kuvantamisjärjestelmät ja -systeemit käyttäytyvät pitkälti siten, että niillä on kolme tärkeää ominaisuutta:

- **Superpositioperiaate**: Oletetaan, että

  $$R(f+g) = R(f) + R(g),$$

  eli syötteiden summan vaste on erillisten vasteiden summa.

- **Skaalautuvuus**: Nollasyötteen vaste on nolla, ja skaalatun syötteen vaste on skaalattu versio alkuperäisen syötteen vasteesta, eli

  $$R(kf) = kR(f).$$

  Systeemi joka toteuttaa superpositioperiaatteen ja on skaalautuva on *lineaarinen*.

- **Siirtoinvarianttius**: Siirtoinvariantissa systeemissä vaste siirrettyyn syötteeseen on alkuperäisen syötteen vasteen vastaava siirtymä. Kameran tapauksessa tämä tarkoittaa yksinkertaisesti esimerkiksi sitä, että jos keskelle kameran näkökenttää asetettu pieni valopiste näkyy kameran ottamassa kuvassa keskellä kuvaa kirkkaana läiskänä, ja valopistettä siirretään reunaa kohti, tulisi kuvassa näkyä edelleen sama läiskä siirtyneenä.

Laitetta, järjestelmää tai systeemiä joka on lineaarinen ja siirtoinvariantti kutsutaan *lineaariseksi siirtoinvariantiksi* laitteeksi, järjestelmäksi tai systeemiksi. Mikä on tärkeää on se, että lineaarisen siirtoinvariantin systeemin vaste annettuun syötteeseen saadaan määrättyä konvoluution avulla. Ensin tämä seikka todennetaan systeemeille, joiden syötteet ja vasteet ovat diskreettejä (lukuvektoreita tai taulukkoja) ja sen jälkeen yleistetään tilanne systeemeihin joiden syötteet ja vasteet ovat jatkuvia 1d- tai 2d-reaaliarvoisia funktioita.

1.2.1 Diskreetti konvoluutio

Tarkastellaan tilannetta ensin yksiulotteisessa tapauksessa ja oletetaan, että on annettu lineaarinen siirtoinvariantti systeemi jonka syötteet ja vasteet ovat vektoreita. Tässä oletetaan että syöte- ja vastevektorit sisältävät (numeroituvasti) äärettömän määrän alkioita. Täten voidaan toistaiseksi välttää pienet tarkennusta vaativat seikat syötteiden ja vasteiden alku- ja loppukohdissa. Näihin palataan myöhemmin.

Yksiulotteinen diskreetti konvoluutio

Oletetaan, että on annettu syötevektori f, joka on ääretön ja indeksoitu kokonaisluvuilla (eli on olemassa -1:s alkio jne.) ja jonka i:nnettä alkiota merkitään f_i:llä, $i = \ldots -2, -1, 0, 1, 2 \ldots$. Syötevektori f voidaan esittää painotettuna summana kantavektoreista. Sopivat kantavektorit ovat sellaisia, joiden yksi alkio on arvoltaan 1 muiden ollessa arvoltaan 0. Merkitään

$$e_i = \ldots 0, 0, 1, 0, 0 \ldots$$

joka on siis vektori jonka alkio i on 1 muiden alkioiden ollessa 0.

Määritellään siirto-operaattori $\mathrm{Shift}(f,i)$, joka palauttaa vektorin jonka j:s alkio on vektorin f $j-i$:s alkio: $\mathrm{Shift}(f,i)_j = f_{j-i}$. Esimerkiksi vektorin $\mathrm{Shift}(e_0, 1)$ alkio numero 1 on 1, muut alkiot ovat nollia. Siirto-operaattorin avulla vektori f voidaan esittää kantavektorin e_0 avulla seuraavasti:

$$f = \sum_i f_i \,\mathrm{Shift}(e_0, i). \tag{1.5}$$

Merkitään lineaarisen siirtoinvariantin systeemimme vastetta syötevektoriin f merkinnällä $R(f)$. Koska systeemi on siirtoinvariantti, täytyy olla voimassa

$$R(\mathrm{Shift}(f,k)) = \mathrm{Shift}(R(f),k),$$

ja lisäksi koska se on lineaarinen on voimassa

$$R(kf) = kR(F), \quad R(f+g) = R(f) + R(g).$$

Nämä seikat yhdessä tarkoittavat sitä, että

$$\begin{aligned} R(f) &= R\left(\sum_i f_i \operatorname{Shift}(e_0, i)\right) \\ &= \sum_i R(f_i \operatorname{Shift}(e_0, i)) \\ &= \sum_i f_i R(\operatorname{Shift}(e_0, i)) \\ &= \sum_i f_i \operatorname{Shift}(R(e_0), i). \end{aligned} \tag{1.6}$$

Mutta tämä tarkoittaa sitä, että voimme määrätä systeemin vasteen mihin tahansa datavektoriin f, kunhan vain tunnemme sen vasteen vektoriin e_0. Tätä vastetta $R(e_0)$ kutsutaan yleensä systeemin *(yksikkö)impulssivasteeksi*. Yksikköimpulssivasteita mitataan monista laitteista. Näitä mm. kultakorvahifistit tutkailevat tarkasti vahvistintestiraporteista ja päättelevät laitteen sopivuutta oman laitteiston osaksi.

Merkitään nyt systeemin yksikköimpulssivastetta $g = R(e_0)$. Nyt siis

$$R(f) = \sum_i f_i \operatorname{Shift}(g, i) = g * f. \tag{1.7}$$

Tämä määrittää yksiulotteisen diskreetin konvoluutio-operaation, jota merkitään operaattorisymbolilla $*$. Jos tarkastellaan vasteen $R(f)$ alkiota j, saadaan lauseke

$$R(f)_j = \sum_i g_{j-i} f_i, \tag{1.8}$$

joka on analoginen kaavan (1.1) esittämän kaksiulotteisen konvoluutiolausekkeen kanssa, ja itse asiassa selittää sen alkuperän. Lineaarisen siirtoinvariantin systeemin **vaste** syötteeseen **saadaan** siis **konvoluutio-operaatiolla syötteestä ja systeemin yksikköimpulssivasteesta.**

Kaksiulotteinen diskreetti konvoluutio

Tässä tapauksessa sekä syöte että vaste ovat äärettömiä kaksiulotteisia taulukoita, joiden rivejä ja sarakkeita indeksoidaan kokonaisluvuin. Merkitään tällaisen taulukon F alkiota i,j $F_{i,j}$:llä. Kaksiulotteisessa diskreetissä tapauksessa yksikköimpulssi on ääretön taulukko E, jonka alkiot ovat kaikki nollia paitsi $E_{0,0}=1$. Jos $G=R(E)$ on systeemin R yksikköimpulssivaste, johtaa vastaava päättely kuin edellä yksiulotteisessa tapauksessa tulokseen, jonka mukaan systeemin vaste syötteeseen F on

$$R(F)_{i,j}=\sum_{u,v}G_{i-u,j-v}F_{u,v}=G**F. \tag{1.9}$$

Merkinnällä $**$ täsmennetään kyseessä olevan kaksiulotteinen konvoluutio-operaatio.

1.2.2 Jatkuva konvoluutio

Useat lineaariset siirtoinvariantit systeemit tuottavat jatkuvan vasteen jatkuvaan syötteeseen. Esimerkkinä vaikka kameran linssi joka kokoaa valoinformaation joka voidaan mallittaa kaksiulotteisen funktiona, ja tuottaa filmipinnalle siitä toisen representaation joka edelleen voidaan mieltää toisena kaksiulotteisena funktiona. Useat linssijärjestelmät ovat likimäärin siirtoinvariantteja. Tällaisia systeemejä tutkimalla saamme selville sen mitä informaatiota katoaa, kun jatkuvaa kaksiulotteista funktiota (linssistön filmipinnalle tai ccd-kennolle kuvaamaa valoinformaatiota) approksimoidaan diskreetillä funktiolla (ccd-kennon näytteistämillä vakioilla pikseliarvoilla).

Luonnollinen systeemin toiminnan kuvaus pohjautuu sen vasteeseen melko outoon otukseen, n.k. δ-funktioon, joka itse asiassa ei ole funktio perinteisessä mielessä.

Yksiulotteinen jatkuva konvoluutio

Johdetaan jatkuva tapaus lähtien liikkeelle aiemmin esitetystä diskreetistä tapauksesta. Otetaan lähtökohdaksi diskreetti syöte ja muodostetaan siitä paloittain vakio ja joka pisteessä määritelty laatikoista koottu porrasfunktio. Sen jälkeen pienennetään ja pienennetään laatikoiden leveyttä ja tutkitaan mitä raja-arvona saadaan.

Systeemimme ottaa syötteekseen yksiulotteisen reaaliarvoisen funktion ja palauttaa vasteenaan toisen vastaavanlaisen funktion. Merkitään jälleen $R(f(x))$:llä systeemin vastetta syötefunktioon $f(x)$, ja korostamme vasteen olevan funktion merkitsemällä sitä tarvittaessa $R(f(x))(u)$:lla. Lineaarisuus tarkoittaa tätä notaatiota käytettäessä ominaisuutta

$$R(f(x)+g(x))(u)=R(f(x))(u)+R(g(x))(u), \quad R(kf(x))(u)=kR(f(x))(u),$$

missä k on jokin reaalilukuvakio. Siirtoinvarianttiutta varten määritellään siirto-operaattori Shift, joka kuvaa funktiot funktioiksi seuraavasti

$$\mathrm{Shift}(f,c)(u) = f(u-c),$$

eli esimerkiksi $\mathrm{Shift}(f,1)(1) = f(0)$. Siirto-operaattorin avulla lausuttuna siirtoinvarianttiusominaisuus tarkoittaa, että (funktioiden argumentit poistettu selkeyden vuoksi)

$$R(\mathrm{Shift}(f,c)) = Shift(R(f),c).$$

Määritellään laatikkofunktio box_ε seuraavasti

$$\mathrm{box}_\varepsilon(x) = \begin{cases} 0, & \|x\| \geq \varepsilon/2, \\ 1, & \|x\| < \varepsilon/2, \end{cases}$$

Olkoon nyt jatkuva syötefunktio $f(x)$. Konstruoidaan tasavälinen pisteistö x_i, missä $x_{i+1} - x_i = \varepsilon$. Muodostetaan diskreetti vektori f, jolle $f_i = f(x_i)$, ja joka edustaa funktiota $f(x)$.

Muodostetaan jatkuvalle funktiolle $f(x)$ porrasfunktioapproksimaatio

$$f \approx \sum_i f_i \,\mathrm{Shift}(\mathrm{box}_\varepsilon, x_i), \tag{1.10}$$

joka annetaan syötteeksi lineaariselle siirtoinvariantille systeemillemme, jonka vaste on tällöin painotettu summa siirretyistä box_ε-funktion vasteista:

$$\begin{aligned} R\left(\sum_i f_i \,\mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)\right) &= \sum_i R(f_i \,\mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)) \\ &= \sum_i f_i R(\mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)) \\ &= \sum_i f_i \,\mathrm{Shift}(R(\frac{\mathrm{box}_\varepsilon}{\varepsilon}\varepsilon), x_i) \\ &= \sum_i f_i \,\mathrm{Shift}(R(\frac{\mathrm{box}_\varepsilon}{\varepsilon}), x_i)\varepsilon \end{aligned} \tag{1.11}$$

Tähän saakka on noudatettu diskreetin tapauksen menettelytapaa. Huomaa, että lopputulos muistuttaa porrasfunktion integraalia. Esitellään nyt raja-arvona n.k. δ-funktio, jonka avulla hoidetaan termi $\mathrm{box}_\varepsilon / \varepsilon$:

$$\delta(x) = \lim_{\varepsilon \to 0} \frac{\mathrm{box}_\varepsilon}{\varepsilon}. \tag{1.12}$$

Eräs mielenkiintoinen δ-funktion ominaisuus on se, että käytännön siirtoinvarianteilla systeemeillä on olemassa vaste δ-funktiolle ja vasteella on kompakti kantaja (eli vaste on nolla poislukien äärellistä määrää äärellisiä välejä). Hyvä mielikuvamalli δ-funktiolle kaksiulotteisessa tapauksessa on äärimmäisen kirkas äärimmäisen pieni valopiste. Jos valopistettä pienennetään samalla valon intensiteettiä nostaen energian säilyessä vakiona odotamme kameramme kuvassa näkyvän pienen, mutta äärellisen valoläiskän epäfokuksessa olevan linssin takia. δ-funktio on luonnollinen vastine diskreetille yksikköimpulssille e_0 jatkuvassa tapauksessa.

Nyt lauseke

$$\sum_i f_i \,\mathrm{Shift}(R(\frac{\mathrm{box}_\varepsilon}{\varepsilon}), x_i)\varepsilon$$

muuttuu integraaliksi raja-arvona, kun $\varepsilon \to 0$. Samalla porrasfunktioapproksimaatiomme lähestyy raja-arvonaan syötefunktiota f ja

$$R(f)(x) = \int R(\delta)(x-x')f(x')\,dx' = \int g(x-x')f(x')\,dx', \tag{1.13}$$

missä olemme esittäneet yksikköimpulssivasteelle merkinnän $g = R(\delta)$ ja jättäneet merkitsemättä integrointirajat. Integraali voi olla yli koko reaaliakselin tai äärellisen välin jos g:n ja h:n kantajat ovat kompakteja. Tämä operaatio on nimeltään jatkuva konvoluutio, ja sen avulla saadaan lineaarisen ja siirtoinvariantin systeemin vaste syötteestä ja systeemin yksikköimpulssivasteesta

$$R(f) = g * f = \int g(x-x')f(x')\,dx',$$

käyttäen jälkeen $*$-merkintää konvoluutio-operaattorille. Konvoluutio on symmetrinen eli

$$(g*h)(x) = (h*g)(x)$$

ja assosiatiivinen tarkoittaen, että

$$(f*(g*h)) = ((f*g)*h).$$

Assosiatiivisuuden takia voimme löytää yhden lineaarisen siirtoinvariantin systeemin, joka toimii kuin kahden systeemin kooste. Tämä on hyödyllinen ominaisuus tutkittaessa näytteistystä.

Kaksiulotteinen jatkuva konvoluutio

Kaksiulotteisessa tapauksessa "laatikkofunktio"määritellään tulona $\mathrm{box}_\varepsilon(x,y) = \mathrm{box}_\varepsilon(x)\,\mathrm{box}_\varepsilon(y)$ ja δ-funktio raja-arvona

$$\delta(x,y) = \lim_{\varepsilon\to 0} \frac{\mathrm{box}_\varepsilon(x,y)}{\varepsilon^2}.$$

Vastaava (teknisempi) tarkastelu johtaa lopputulokseen

$$R(f)(x,y) = \int\int g(x-x', y-y') f(x',y')\, dx' dy' = g(x,y) ** f(x,y), \qquad (1.14)$$

missä $g(x,y) = R(\delta(x,y))$ on systeemin R yksikköimpulssivaste tai *pisteen levitysfunktio* (engl. point spread funtion), joka nimi tulee kameran optiikan epätäydellisyyden aiheuttamasta etäisen pienen kirkkaan kohteen kuvautumisesta sumeaksi läiskäksi. Tämä on hyvä mielikuva yksikköimpulssivasteen olemuksesta.

1.2.3 Reunailmiöt diskreeteissä konvoluutioissa

Käytännön tietokonelaskennassa ei voida käyttää äärettömiä datavektoreita tai taulukoita. Tämän takia täytyy jotenkin huolehtia äärellisen (kuva)datataulukon käsittelystä taulukon reunojen lähistöllä. Reunojen lähistöllä ja reunoilla konvoluution laskennassa tarvittaisiin arvoja, joita ei ole olemassa lainkaan. Tätä ongelmaa voi lähestyä usealla vaihtoehtoisella tavalla:

- **Konvoluution tulosta ei lasketa lainkaan reunan lähistöllä.** Tämä tarkoittaa sitä, että konvoluution tulos lasketaan vain niille tulostaulukon alkioille, joiden laskennassa tarvittava kaikki lähtödata on saatavilla. Tällä tavalla ei sorruta keinotekoiseen peukalointiin, mutta haittana on se, että tulostaulukon koko on pienempi kuin syötteen. Toistuva laskenta kutistaa lopputuloksen huomattavasti.

- **Laajennetaan lähtödataa keinotekoisesti vakioarvoilla.** Tässä tapauksessa todellisen syötedatan vaikutus vähenee merkittävästi reunan lähistöllä. Tämä voi merkittävästi vääristää tulosta, erityisesti se voi vaikuttaa merkittävästi tulosdatan derivaattaan. Etuna on tietenkin se, että tulostaulukon koko säilyy syötetaulukon kokoisena.

- **Laajennetaan lähtödataa muulla tavalla.** Lähtödata voidaan esimerkiksi mieltää tuplaperiodiseksi. Jos lähtödatataulukon koko on $m \times m$, asetetaan laskennassa tarvittava sarake $m+1$ vastaamaan saraketta 1 jne. Tämä taktiikka puolestaan saattaa vaikuttaa merkittävästi tulosdatan toiseen derivaattaan reunan lähettyvillä.

1.2.4 Oikeat kuvantamisjärjestelmät vs siirtoinvariantit lineaariset systeemit

Kuvantamisjärjestelmät ovat vain likimäärin lineaarisia. Valokuvafilmi ei ole lineaarinen. Se ei ole herkkä erityisen heikolle valolle ja se saturoituu (ylivalottuu) liian kirkkaassa valaistuksessa. Poislukien nämä ääripäät on lineaarinen malli kuitenkin yleensä käyttökelpoinen. CCD-kamerat ovat lineaarisia normaalityöskentelyalueella. Lämpökohinan takia ne kuitenkin antavat hyvin pienen vasteen nollasyötteelle (tämän takia astronomit jäähdyttävät kameroitaan) ja myös ne saturoituvat hyvin kirkkaalla syötteellä. Useat ccd-kamerat sisältävät elektroniikkaa jolla jäljitellään perinteistä filmiä, koska ihmiset ovat tottuneet perinteisiin valokuviin. Siirtoinvarianssikin on vain approksimaatio todellisuudesta, koska linssistöillä on taipumus vääristää kuvaa etenkin lähellä reuna-alueita.

1.3 Spatiaalinen taajuus ja Fourier-muunnos

Edellä käytettiin trikkiä, jossa signaali $f(x,y)$ esitettiin painotettuna summana hyvin suuresta (tai äärettömästä) määrästä hyvin pieniä (tai äärettömän pieniä) laatikkofunktioita. Tämä tapa korostaa ajatusta siitä, että signaali on *vektoriavaruuden alkio, eli vektori*. Laatikkofunktiot muodostavat erään kannan tähän vektoriavaruuteen ja painokertoimet muodostavat vektorin alkioittaisen esityksen tässä "laatikkokannassa". Tarvitsemme vielä uuden työkalun käsitelläksemme jäljellä olevat kaksi avointa kysymystä:

- Vaikka onkin selvää, ettei kuvan diskreetti pikselirepresentaatio voi sisältää jatkuvan kuvasignaalin kaikkea informaatiota, emme vielä tiedä mikä osa informaatiosta katoaa.

- On myös selvää, ettemme voi tiputtaa diskreetin pikselirepresentaation resoluutiota yksinkertaisella tavalla ottamalla mukaan vain joka k:nnen pikselin (shakkilaudan kuvasta voisi tulla kokomusta tai kokovalkoinen). Kuinka resoluutiota sitten turvallisesti ja minimivahingoin pudotetaan?

Nämä molemmat avoimet ongelmat liittyvät kuvasignaalissa esiintyviin nopeisiin muutoksiin. Esimerkiksi resoluution pudottaminen alkeellisesti saattaa hukata nopeita vaihteluita, koska ne tapahtuvat matalaresoluutioversioon valittujen pikselien välisessä osassa alkuperäistä dataa.

Koska voimme tulkita kuvasignaalimme vektoriavaruuden alkioina, voimme myöskin tehdä vektoriavaruuteen *kannanvaihdon*, ja tutkia saman signaalin esitystä uudessa kannassa. Sopiva uusi kanta koostuu jatkuvassa tapauksessa äärettömästä määrästä eri suuntiin etenevistä ja eri taajuisista sinitasoaalloista. Kun signaali esitetään tällaisessa kannassa, on nopeasti muuttuva informaatio helppo havaita, sillä korkeataajuisia tasoaaltokantavektoreita vastaavat signaalivektorin alkiot ovat tällöin suuria. Vektoriesityksen alkioiden arvot ovatkin suoraan verrannollisia vastaavan sinitasoaaltokantafunktion merkittävyyteen signaalin esittäjänä.

1.3.1 Fourier-muunnos

Keskitytään aluksi jatkuvaan Fourier-muunnokseen, koska sitä tarvitaan lähinnä käsitteellisenä apuneuvona tässä vaiheessa. Haluttu kannanvaihto taajuusrepresentaatioon tehdään Fourier-muunnoksella. Määritellään signaalin $g(x,y)$ Fourier-muunnokseksi tuplaintegraali

$$F(g(x,y))(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\, e^{-i2\pi(ux+vy)}\, dxdy. \tag{1.15}$$

Oletetaan, että integraali on hyvin määritelty, kuten se on sovelluksissamme (tarkempaa detaljitietoa matematiikan kursseilta). Fourier-muunnos kuvaa kompleksiarvoisen funktion $g(x,y)$ toiseksi kompleksiarvoiseksi funktioksi $F(g(x,y))(u,v)$ (kuvasignaalit $g(x,y)$ ovat kompleksisia signaaleja, niiden imaginaariosa sattuu vain olemaan 0).

Kiinnitetään piste (u,v) ja tutkitaan muunnoksen arvoa tässä pisteessä. Eksponenttifunktiolauseke voidaan aukikirjoittaa muotoon

$$e^{-i2\pi(ux+vy)} = \cos(2\pi(ux+vy)) - i\sin(2\pi(ux+vy)).$$

Sekä reaalitermi että imaginaaritermi ovat xy-tason sinitasoaaltoja, joiden suunta ja taajuus määräytyy parametreista u ja v. Jos tutkitaan esimerkiksi reaalitermiä, havaitaan sen olevan vakio kunhan lausekkeen $ux+vy$ arvo on vakio, eli pitkin xy-tason suoria joiden suunnalle pätee $\tan\theta = v/u$. Reaalitermin gradienttivektori on kohtisuorassa tätä suoraa vastaan, ja reaalitermiä vastaavan sinitasoaallon taajuus on $\sqrt{u^2+v^2}$. Tällaisia sinitasoaaltoja kutsutaan *spatiaalisiksi taajuuskomponenteiksi*.

Integraali (1.15) tulee tulkita *sisätulona*. Jos u ja v kiinnitetään, on integraalin arvo x ja y-koordinaattien määräämään sinitasoaallon ja alkuperäisen signaalin sisätulo. Analogia on hyödyllinen, koska sisätulothan mittaavat yhden vektorin "määrää"toisen vektorin suuntaan. Aivan samoin muunnoksen arvo tietyillä u ja v voidaan tulkita mittaustuloksena siitä kuinka paljon annetun taajuista ja suuntaista sinitasoaaltoa signaalimme sisältää. Signaalin Fourier-muunnos koostaa tämän mittaustulosinformaation kaikilla mahdollisilla arvoilla u ja v, eli kaikkiin eri suuntiin eteneville eritaajuisille sinitasoaalloille!

Fourier-muunnos on lineaarinen, eli:

$$F(g(x,y)+h(x,y)) = F(g(x,y))+F(h(x,y)), \quad F(kg(x,y)) = kF(g(x,y)),$$

ja sillä on käänteismuunnos, jolla alkuperäinen signaali voidaan palauttaa Fourier-muunnoksestaan. Tämä on vain toinen kannanvaihto vektoriavaruudessa ja se on muotoa:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(g(x,y))(u,v)\, e^{i2\pi(ux+vy)}\, du dv. \tag{1.16}$$

Fourier-muunnokset tunnetaan suljetussa muodossa useassa hyödyllisessä tapauksessa. Muutamia näistä on taulukoitu taulukkoon 1.1. Viimeinen taulukon muunnospari tunnetaan myös *konvoluutioteoreemana*: konvoluutio paikkatasossa on kertolasku taajuustasossa ja kertolasku paikkatasossa on konvoluutio taajuustasossa! Tämä on tärkeää useassa tapauksessa. Esim diskreettiä Fourier-muunnosta soveltamalla voidaan n.k. FFT-algoritmin avulla laskea tehokkaasti konvoluutioita erityisesti, jos maskin koko on suuri.

Funktio	Fourier-muunnos
$\delta(x,y)$	1
$\frac{\partial f}{\partial x}(x,y)$	$uF(f)(u,v)$
$e^{-\pi(x^2+y^2)}$	$e^{-\pi(u^2+v^2)}$
$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)$	$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(u-i,v-j)$
$\mathrm{box}_1(x)\,\mathrm{box}_1(y)$	$\frac{\sin(\pi u)}{\pi u}\frac{\sin(\pi v)}{\pi v}$
$(f**g)(x,y)$	$F(f)F(g)(u,v)$

Taulukko 1.1: Eräitä Fourier-muunnospareja

Fourier muunnoksen arvo tietyssä pisteessä u, v riippuu koko muunnettavasta funktiosta. Tämä on selvää, koska integraali lasketaan yli muunnettavan funktion koko määrittelyjoukon. Tästä seuraa mm. se, että pienikin paikallinen muutos muunnettavassa funktiossa (esim. diskreetissä tapauksessa yksittäisen pikseliarvon muuttaminen) muuttaa kaikkia Fourier-kertoimia. Tämän takia Fourier-kertoimet ovatkin aika hankala datan representaatiokeino. On esimerkiksi hyvin vaikeaa päätellä Fourier-kertoimista onko alkuperäisessä muunnettavassa kuvasignaalissa joku tietty kuvio.

1.3.2 Esimerkki: silmälasien välttämättömyys

Konvoluutioteoreeman ja toiseksi viimeisen taulukkorivin muunnosparin avulla saamme idealisoidun matemaattisen todistuksen sille miksi silmälasit ovat välttämättömät taittovirheestä kärsiville! Taittovirheen takia kohde ei tarkennu täsmälleen verkkokalvolle vaan se tarkentuu joko verkkokalvon eteen tai taakse. Kummassakin tapauksessa yksittäinen piste projisoituu verkkokalvolle kiekoksi, jota tässä approksimoidaan laatikkofunktiona. Taittovirheinen silmä on siis tässä eittämättä hieman keinotekoisessa esimerkissä lineaarinen siirtoinvariantti systeemi, jonka hypoteettinen yksikköimpulssivaste on $g(x,y) = R(\delta(x,y)) = \mathrm{box}_1(x)\,\mathrm{box}_1(y)$. Systeemin vaste syötteeseen saatiin yksikköimpulssivasteen ja syötesignaalin $f(x,y)$ konvoluutiona. Konvoluutioteoreeman mukaan vastaava tulos saadaan kertomalla syötteen Fourier-muunnos yksikköimpulssivasteen Fourier-muunnoksella ja ottamalla tuloksesta käänteismuunnos

$$f(x,y) ** g(x,y) = F^{-1}(F(f(x,y))F(g(x,y))) = F^{-1}(F(f(x,y))\frac{\sin(\pi u)}{\pi u}\frac{\sin(\pi v)}{\pi v}).$$

Koska funktio $sin(x)/x$ saa arvon 0 kun $x = \ldots -2\pi, -\pi, \pi, 2\pi \ldots$ kuvautuu osa alkuperäisen signaalin Fourier-kertoimista nollaksi. Tämä on peruuttamaton vahinko, sillä alkuperäistä informaatiota ei voi enää tämän jälkeen palauttaa takaisin (se vaatisi nollalla jakamisen, ja edes aivot eivät ole siihen miljoonien vuosien evoluution aikana oppineet)!

1.4 Näytteistäminen ja aliasoituminen

Paneudutaan nyt kysymykseen siitä mikä on jatkuvan ja diskreetin kuvasignaalin ero. Erityisesti yritetään selvittää mitä informaatiota hukkuu, kun kuva näytteistetään ja siitä muodostetaan diskreetti pikselitaulukko. Hyvä yksinkertainen esimerkki on shakkiruudukon jatkuva kuvasignaali. Oletetaan että neliömäisen ruudun koko on kaksi yksikköä suuntaansa. Jos naapurinäytteiden etäisyys on yksi yksikkö, voidaan näytteistysruudukko oikein asemoimalla taata, että jokaista valkoista ja mustaa ruutua vastaa neljä näytepikseliä. Jos näytteet otetaan kahden yksikön välein, saadaan siltikin yksi näyte per ruutu. Jos näytteitä otetaankin kolmen yksikön välein käy väistämättä siten, ettei kaikista ruuduista saada näytettä ja neljän yksikön välein näytteistetty pikselikuva voi olla kokomusta tai kokovalkoinen. Ongelma näyttäisi liittyvän näytteiden lukumäärään suhteessa näytteistettävään funktioon. Tämä voidaan formalisoida melko tarkasti ja tästä voidaan johtaa melko kattava malli.

1.4.1 Näytteistäminen

Näytteistämisellä tarkoitetaan jatkuvan signaalifunktion arvojen keräämistä diskreetin hilan hilapisteissä.

Yksiulotteisessa tapauksessa signaalifunktio on kuvaus reaaliluvuilta reaaliluvuille, ja näytteistäminen tarkoittaa diskreetin arvojoukon poimintaa tällaisesta kuvauksesta diskreetissä määrittelyjoukon pisteistössä. Tärkein erikoistapaus on kerätä näytteet tasaväliseltä pisteistöltä, joten voimme olettaa että näytteet kerätään reaalilukuakselin kokonaislukupisteissä. Tämä tarkoittaa sitä, että käytössämme on operaatio $\mathrm{Sample}_{1\mathrm{d}}$, jonka syöte on funktio $f(x)$ ja joka palauttaa kokonaisluvuilla indeksoitavan reaalilukuvektorin f:

$$f = \mathrm{Sample}_{1\mathrm{d}}(f(x)), \quad f_i = f(i), \quad i = \ldots -2, -1, 0, 1, 2 \ldots \tag{1.17}$$

Kaksiulotteinen tapaus on hyvin samankaltainen kuin yksiulotteinen. Vaikka näytteistys voisikin perustua epäsäännölliseen hilaan (kuten silmän verkkokalvolla), jatkamme sillä oletuksella, että näytteet kerätään tason kokonaislukukoordinaattipisteissä. Tämä on myös hyvä malli useimmille digitaalisille kameroille. Lopulliset näytteistetyt kuvat ovat suorakulmaisia äärellisen kokoisia kaksiulotteisia taulukoita, sillä kuva-alue on tällainen ja sen ulkopuolella näytteitä ei oteta (näytearvot asetetaan nolliksi). Kaksiulotteinen näytteistys voidaan esittää operaation $\mathrm{Sample}_{2\mathrm{d}}$ avulla. Tämän syöte on funktio $F(x,y)$ reaalilukutasolta reaalilukuakselille ja se palauttaa kaksiulotteisen taulukon $F_{i,j}$ jonka sekä rivejä että sarakkeita indeksoidaan kokonaisluvuilla:

$$F = \mathrm{Sample}_{2\mathrm{d}}(F(x,y)), \quad F_{i,j} = f(i,j), \quad i,j = \ldots -2, -1, 0, 1, 2 \ldots \tag{1.18}$$

Näytteistetyn signaalin integroituva malli

Tarvitsemme jatkoanalyysia varten integrotuvan mallin näytteistetystä signaalista. Tätä tarvitaan erityisesti Fourier-muunnosta varten, jossa mallimme ja kompleksisen eksponenttifunktion lausekkeen tulo täytyy pystyä integroimaan. Signaalin integraalin tulisi selvästikin yhtyä sen näytearvotaulukon alkioiden summaan. Emme siis voi mallittaa näytteistettyä signaalia funktiona, jonka arvo on nolla, paitsi kokonaislukukoordinaattipisteissä joissa sen arvo olisi näytearvo, koska tällaisen funktion integraali olisi 0.

Aiemmin esitetyllä δ-funktiolla on tärkeä ominaisuus, jota voidaan hyödyntää integroituvan mallin muodostuksessa:

$$\begin{aligned} \int_{-\infty}^{\infty} a\delta(x) f(x)\, dx &= a \lim_{\varepsilon\to 0} \int_{-\infty}^{\infty} \frac{\mathrm{box}_\varepsilon}{\varepsilon} f(x)\, dx \\ &= a \lim_{\varepsilon\to 0} \sum_{i=-\infty}^{\infty} \frac{\mathrm{box}_\varepsilon}{\varepsilon} f(i\varepsilon)\, \mathrm{Shift}(\mathrm{box}_\varepsilon, i\varepsilon)\varepsilon \\ &= af(0), \end{aligned} \tag{1.19}$$

eli δ-funktion ja jatkuvan funktion f tulon intergaalin arvo on $f(0)$.

Tästä saamme soveliaan mallin: asetamme jokaiseen näytteistyspisteeseen δ-funktion, jonka arvoa painotamme pisteeseen liittyvällä näytearvolla. Tämä saadaan aikaan kertomalla näytteistetty signaali joukolla δ-funktiota, joita on yksi per näytepiste. Näin ollen voimme uudelleen määritellä näytteistysoperaattorin yksiulotteisessa tapauksessa:

$$\mathrm{Sample}_{1\mathrm{d}}(f(x)) = \sum_{i=-\infty}^{\infty} f(i)\, \mathrm{Shift}(\delta(x), i) = f(x) \sum_{i=-\infty}^{\infty} \delta(x-i). \tag{1.20}$$

Summalauseketta $\sum_{i=-\infty}^{\infty} \delta(x-i)$ kutsutaan (ilmeisestä syystä) *kampafunktioksi* (engl. comb function). Sen kaksiulotteinen vastine on n.s. "*naulasänkyfunktio*" (engl. bed-of-nails function) jonka avulla kaksiulotteinen näytteistäminen voidaan muotoilla seuraavasti:

$$\begin{aligned} \mathrm{Sample}_{2\mathrm{d}}(F(x,y)) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F(i,j)\delta(x-i, y-j) \\ &= F(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j). \end{aligned} \tag{1.21}$$

Molemmilla näytteistysoperaattoreilla on nyt haluttu ominaisuus: ne palauttavat integroitavan "otuksen", jonka integraalin arvo on näytearvojen summa.

1.4.2 Aliasoituminen

Näytteistyksessä informaatiota katoaa. Näytämme miten liian hitaasti/harvasti näytteistämällä saatu näytearvojoukko tulkitsee alkuperäistä jatkuvaa signaalia väärin: Alkuperäisen signaalin korkeataajuiset spatiaaliset taajuuskomponentit ilmenevät matalataajuisina spatiaalisina taajuuskomponentteina näytteistetyssä signaalissa. Tämä ilmiö tunnetaan nimellä *aliasoituminen*.

Näytteistetyn signaalin Fourier-muunnos

Aiemmin johdimme näytteistetylle signaalille mallin, joka saadaan kertomalla alkuperäinen jatkuva signaali naulasänkyfunktiolla. Konvoluutioteoreeman mukaan tämän tulon Fourier-muunnos on näiden kahden funktion Fourier-muunnosten konvoluutio. Taulukon 1.1 mukaan naulasänkyfunktion Fourier-muunnos on toinen, taajuustason naulasänkyfunktio. Funktion konvoluutio naulasänkyfunktion kanssa muodostaa summan naulasänkyfunktion δ-piikkeihein siirretyistä funktion kopioista. Tässä tapauksessa konvoluutio lasketaan siis alkuperäisen signaalin Fourier-muunnoksen kanssa ja efekti on se, että tuloksena saadaan siirrettyjä Fourier-muunnoksia jotka summataan yhteen:

$$
\begin{aligned}
F(\mathrm{Sample}_{2\mathrm{d}}(F(x,y))) &= F\left(F(x,y)\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right) \\
&= F(u,v) ** F\left(\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right) \\
&= F(u,v) ** \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(u-i,v-j) \\
&= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}F(u-i,v-j),
\end{aligned}
\tag{1.22}
$$

missä funktion $F(x,y)$ Fourier-muunnosta on merkitty $F(u,v)$:llä.

Jos signaalin siirrettyjen Fourier-muunnosten kantajat eivät leikkaa toisiaan voidaan alkuperäinen signaali palauttaa täydellisesti takaisin leikkaamalla yksi Fourier-muunnoskopio talteen ja muuntamalla se takaisin käänteismuunnoksella.

Kuitenkin usein käy niin, että siirrettyjen Fourier-muunnosten kantajat leikkaavat toinen toisiaan. Tällaisessa tapauksessa emme pysty palauttamaan alkuperäistä signaalia täydellisenä takaisin. Tämä johtuu siitä, että kantajien leikkausalueella on summautunut yhteen saman Fourier-muunnetun signaalin matala- ja korkeataajuiset kertoimet. Jos tunnetaan vain summan arvo, ei tästä tiedosta pysty yksikäsitteisesti päättelemään summautuneita termejä. Tämä ilmiö tapahtuu *aina* kun näytteenottotaajuus on alle kaksi kertaa signaalin sisältämä korkein taajuuskomponentti. Tämä tulos tunnetaan kirjallisuudessa *Nyquistin teoreemana*.

1.4.3 Silottaminen ja uudelleennäytteistäminen

Nyquistin teoreema tarkoittaa, että on vaarallista pudottaa kuvan resoluutiota yksinkertaisesti ottamalla matalaresoluutiorepresentaation vain joka k:s pikseli. Ennen sitä lähtökuva tulee *alipäästösuodattaa* siten, että sen ne spatiaaliset taajuuskomponentit, jotka taajuudeltaan ylittävät uuden näytteenottotaajuuden, poistuvat. Tämän voisi tehdä tarkasti kuvasignaalin Fourier-muunnoksen avulla nollaamalla kaikki Fourier-kertoimet $F(u,v)$, jotka liittyvät liian korkeisiin taajuuksiin $\sqrt{u^2+v^2} > F_{tol}$ ja muuntamalla signaali käänteisellä Fourier-muunnoksella takaisin paikkatasoon. Yhtäpitävästi, muunnos olisi "mahdollista"tehdä suoraan paikkatasossa sellaisella konvoluutioytimellä $G(x,y)$ jonka painokertoimet ovat muotoa $(\sin x \sin y)/(xy)$. Menetelmä ei kuitenkaan ole käytännöllinen, koska kyseisen painokerroinfunktion kantaja on ääretön.

Tärkein tapaus on se, että haluamme puolittaa sekä kuvan vaaka- että pystyresoluution. Oletamme, että alkuperäinen kuva ei ole aliasoitunut (jos se olisi, ei mitään olisi tehtävissä — kun kuva on alunperin näytteistetty on mahdollinen aliasoituminen jo tapahtunut). Näytteistetyn kuvan Fourier-muunnos koostuu nyt tason kokonaislukukoordinaattipisteisiin keskittyneistä Fourier-muunnoskopioista.

Resoluution puolittaminen tarkoittaa sitä, että Fourier-muunnoskopiot keskittyvät tason pisteisiin, joiden koordinaatit ovat puolen yksikön monikertoja (tai vaihtoehtoisesti muunnokset efektiivisesti levittyvät kaksi kertaa laajemmalle alalle, koska puolta harvemmalla näytteenottotaajuudella aiempi korkein taajuus näyttäytyy efektiivisesti kaksi kertaa korkeampana). Tämä tarkoittaa myös sitä, että välttyäksemme aliasoitumiselta täytyy signaali ensin alipäästösuodattaa siten, että suodatuksen jälkeen signaalin korkein taajuus on enintään puolet aiemmasta korkeimmasta taajuudesta. Liiallista kaistan kavennusta tulee kuitenkin välttää, jottemme turhan paljon köyhdytä informaatiosisältöä.

Koska Gaussin ydin vaimenee nopeasti, ja Gaussin ytimen Fourier-muunnos on toinen, taajuustason Gaussin ydin (katso taulukko 1.1), voi sitä käyttää käytännön approksimaationa ideaaliselle alipäästösuotimelle. Ytimen koon määräävän keskihajonnan σ valinta on sovelluskohtaista. Jos σ on suuri, täytyy maskinkin olla suuri, ja vaikka aliasoituminen onkin tällöin vähäistä (koska ytimen alkiot ovat taajuusrajan jälkeen hyvin pieniä) katoaa informaatiota sen takia, ettei ydin ole tasainen säilytettävän taajuusrajan alapuolella. Vastaavasti valitsemalla pieni σ säilyy informaation taajuusrajan alapuolella melko koskemattomana, mutta aliasoituminen voi olla runsaampaa. Valitsemalla aluksi $\sigma = 1$ saa usein kelvon alkuarvauksen kokeelliseen keskihajonnan valintaan.

1.5 Kuvapyramidit ja skaalat

Kuva näyttää erilaiselta eri skaaloissa tarkasteltuna ja monissa tapauksissa on hyödyllistä esittää kuva esimerkiksi *kokoelmana eriresoluutioisista esityksistään*. Tällöin puhutaan kuvapyramideista (nimitys joka tulee visuaalisen analogian kautta).

1.5.1 Gaussin kuvapyramidi

Kuvapyramidi on kokoelma saman kuvan representaatioita eri resoluutioilla, eli n.k. *moniskaalarepresentaatio*. Tyypillisesti tietyn pyramidin tason kuvarepresentaation pysty- ja vaakaresoluutio on puolet edeltävän tason resoluutioista. Jos eri tasot mielletään pinotun päällekkäin, muodostuu pyramidi. *Gaussin kuvapyramidissa* jokaisen tason representaatio silotetaan symmetrisellä Gaussin ytimellä ja uudelleennäytteistetään seuraavan tason representaatioksi. Pyramidin muodostaminen on suoraviivaisinta, jos kuvan dimensiot ovat kakkosen potensseja tai niiden monikertoja. Mataläresoluutioisin esitys on tietenkin ankarimmin silotettu versio, ja yleisesti pyramidin tasoja kutsutaan *harvan skaalan* (engl. coarse scale) versioiksi alkuperäisestä kuvasta.

Kuvapyramidin muodostamisen esittämisen yksinkertaistamiseksi tehdään ensin se oletus, että alkuperäinen kuva on neliö ja kuva vaaka- ja pystydimensio on 2^k, missä k on kokonaisluku. Määritellään operaattori $\mathrm{S}^{\downarrow}$ joka puolittaa vaaka- ja pystyresoluution. Täsmällisemmin j,k:s pikseli mataläresoluutiokuvassa $\mathrm{S}^{\downarrow}(I)$ on alkuperäisen kuvan I pikseli $2j, 2k$. Lisäksi Gaussin pyramidin $P_{Gauss}(I)$ tasolle n käytetään merkintää $P_{Gauss}(I)_n$.

Gaussin pyramidin tasot määritellään lähtien liikkeelle alkuperäisestä kuvasta I seuraavasti:

$$P_{Gauss}(I)_1 = I, \quad P_{Gauss}(I)_{n+1} = \mathrm{S}^{\downarrow}(G_\sigma ** P_{Gauss}(I)_n) \tag{1.23}$$

Gaussin pyramidin sovelluksia

Gaussin pyramidin avulla voidaan kuvasta hakea eri tarkkuustason rakenteita tehokkaasti. Jos tavoitteena on esimerkiksi löytää seepran raidat, on yksi mahdollisuus soveltaa erilaisia suotimia kuvan täysresoluutioversion. Tällöin ongelmaksi tulee hyvin helposti suotimien vaste kaikenlaiseen epärelevanttiin kuvainformaatioon, kuten raidan yksittäisten karvojen suuntavaihteluihin tms. Tällaisten efektien mallittaminen suotimen painokertoimia säätämällä johtaa vaikeuksiin: suodinmaskien täytyy olla suuria, jolloin suotimen soveltaminen hidastuu. Lisäksi sopivien kertoimien hakeminen tulee hyvin haastavaksi ja laskennallisesti kalliiksi.

Käytännöllisempi ratkaisu onkin soveltaa pienempimaskisia suotimia vähemmän detaljeja sisältävään matalaresoluutioisempaan versioon kuvasta. Gaussin pyramidi tarjoaa valmiiksi kokoelman tällaisia versioita ja mahdollistaa eri tarkkuustason rakenteiden tehokkaan haun.

Toinen tärkeä sovellus on spatiaalinen haku. Tyypillinen tilanne on sellainen, että olemme paikallistaneet kiinnostavan pisteen yhdestä kuvasta ja tavoitteena on löytää vastinpiste toisesta kuvasta. Toinen kuva on esimerkiksi kuvattu toisesta suunnasta (stereonäkö) tai vastinpiste on liikkunut joko kameran tai kuvattavan kappaleen liikuttua (liikeanalyysi).

Vastinpisteen haku alkuperäisestä kuvaparista on epätehokasta, koska pahimmassa tapauksessa kaikkia vastinpistepareja tulee verrata keskenään samankaltaisuuden maksimoimiseksi. Nykyinen melko universaali ratkaisu on hakea ensin vastaavuuksia ankarasti silotetuista matalaresoluutioversioista ja sitten tarkentaa hakua siihen osaan kuvasta, josta paras matalaresoluutiovastaavuus löytyi. Tämä tehostaa hakua merkittävästi. Jos alkuperäinen kuvaresoluutio on 1024×1024 pikseliä, vastaa harvimman esitystason 4×4 resoluution yksi pikseli 256×256 pikselin neljännestä koko kuvassa.

1.5.2 Laplacen kuvapyramidi

Gaussin kuvapyramidi tallettaa redundanttia informaatiota, koska jokainen taso on alipäästösuodatettu versio edeltävästä tasosta. Tällöinhän matalataajuinen informaatio tulee talletettua useaan kertaan. Gaussin kuvapyramidin taso on samalla *ennuste* sitä *edeltävästä* suuriresoluutioisemmasta *tasosta*. Tämä ennuste ei tietenkään ole eksakti, mutta se samalla tarkoittaa sitä ettei ole välttämätöntä tallettaa kaikkea informaatiota korkeamman resoluution tasolla. Laplacen kuvapyramidin ideana onkin tallettaa ennusteen virhe. Laplacen pyramidia voi vielä täydentää orientaatioinformaatiolla.

Laplacen pyramidin muodostamiseksi tarvitsemme *resoluution nosto-operaattorin* (engl. upsampling operator). Selvästikään emme pysty luomaan kuvaan uutta informaatiota tyhjästä, mutta voimme helposti suurentaa resoluutiota *monista-*

malla pikseliarvoja. Määritellään seuraavaksi tällainen resoluution nosto-operaattori $\mathrm{S}^{\uparrow}$, joka ottaa syötteekseen pyramidin tason $n+1$ representaation ja tuottaa siitä korkearesoluutioisemman tason n esityksen. Tarkemmin sanottuna korkearesoluutioisen kuvan $\mathrm{S}^{\uparrow}(I)$ neljän pikselin $(2j-1, 2k-1)$, $(2j-1, 2k)$, $(2j, 2k-1)$ ja $(2j, 2k)$ arvoksi asetetaan kuvan I pikselin j, k arvo.

Laplacen kuvapyramidi muodostetaan valitsemalla sen matalaresoluutioisimmaksi tasoksi Gaussin kuvapyramidin matalaresoluutioisin taso. Jokainen korkearesoluutioinen taso Laplacen pyramidissa saadaan erotuksena vastaavan tason Gaussin pyramidin representaatiosta ja ennusteesta joka saadaan seuraavaksi matalamman resoluution Gaussin pyramidin tasosta resoluution nosto-operaattorilla. Aukikirjoitettuna tämä tarkoittaa:

$$\begin{aligned} P_{Laplace}(I)_n &= P_{Gauss}(I)_n, \\ P_{Laplace}(I)_k &= P_{Gauss}(I)_k - \mathrm{S}^{\uparrow}(P_{Gauss}(I)_{k+1}) \\ &= P_{Gauss}(I)_k - \mathrm{S}^{\uparrow}(\mathrm{S}^{\downarrow}(G_\sigma ** P_{Gauss}(I)_k)), \end{aligned} \tag{1.24}$$

misää n on matalimman resoluution taso.

Esitystavan nimi (Laplacen pyramidi) on vähän harhaanjohtava, koska derivaattaoperaattoreita ei konstruktiossa tarvita. Nimitys perustuu sille seikalle, että pyramidin jokainen taso vastaa likipitäen Gaussisten suodinten erotuksena saatavaa suodinta joka puolestaan on approksimaatio Gaussin suotimen toisena derivaattana (Laplace-operaattorilla) saatavasta suotimesta.

Laplacen kuvapyramidin jokainen taso voidaan mieltää kuvan kaistapäästösuodatettuna versiona, eli kuvan representaationa tietyn taajuusvälin spatiaalisten taajuuskomponenttien avulla.Tämä johtuu siitä, että tietyllä resoluutiolla esitetystä kuvainformaatiosta vähennetään matalamman resoluution version antama ennuste, joka vastaa matalataajuisia spatiaalisia taajuuskomponentteja. Tämä puolestaan antaa olettaa, että tietyt kuvan spatiaaliset taajuuskomponentit ilmenevät voimakkaina vasteina tietyllä pyramidin tasolla ja heikkoina vasteina muilla tasoilla.

Laplacen kuvapyramidilla on vielä yksi tärkeä ominaisuus: Alkuperäisen kuvan palauttaminen Laplacen pyramidistaan on helppoa. Tämä tehdään palauttamalla siitä Gaussin pyramidi, jonka suuriresoluutioisin taso on alkuperäinen kuva. Gaussin pyramidin palauttaminen on hetkessä hoideltu: Lähdetään liikkeelle matalimman resoluution tasosta, joka on samalla Gaussin pyramidin matalimman resoluution taso. Tämän resoluutio nostetaan seuraavaa tasoa vastaavaksi ja lisätään siihen seuraavan tason Laplacen pyramidin representaatio jne:

$$\begin{aligned}
Work_image_n &= P_{Laplace}(I)_n = P_{Gauss}(I)_n, \\
Work_image_k &= P_{Laplace}(I)_k + \mathrm{S}^{\uparrow}(Work_image_{k+1}), \quad k = n-1, \ldots, 1, \\
I &= Work_image_1.
\end{aligned} \tag{1.25}$$