

Kuvankäsittelyä

Mittausten virheet

Useat virhelähteet vaikuttavat havaintoihin, kuten CCD-kuviin:

- Diffraktio ja seeing levittävät kuvaa (blurring).
- Kuvausjärjestelmän optiikka aiheuttaa geometrista vääristymistä.
- Laitteisto itsessään aiheuttaa kohinaa (noise). Kohinan määrää kuvaa signaalin ja kohinan suhde (signal to noise ratio). Suhdetta voidaan parantaa toistamalla havainto useampaan kertaan ja laskemalla saatujen tulosten keskiarvo.
- Signaalin diskretoinnista aiheutuu virhettä. Riippuu A/D-muuntimen bittimäärästä. Kyseessä on satunnainen virhe, joka ei vaikuta monen mittauksen keskiarvoon. Satunnaisen virheen vaikutus hallittavissa tilastollisilla menetelmillä.
- Eri kameroiden herkkyys eri aallonpituuksilla on erilainen. Kyseessä on systemaattinen virhe, joka selviää vasta vertaamalla tuloksia muilla laitteilla tehtyihin. Samalla kameralla tehdyt havainnot voivat olla systemaattisesti vääriä, mutta silti vertailukelpoisia keskenään. Verrattaessa eri laitteilla saatuja tuloksia jokainen laite on kalibroitava erikseen.
- Kuvassa voi olla kirkkaiden kohteiden (Kuu tms.) aiheuttamia heijastuksia. Ongelmana on eri havaintojen vertailukelvottomuus, joka johtuu olosuhteiden muuttumisesta. Havaittaessa muuttuvaa tähteä eri korkeuksilla horisontista ilmakehä aiheuttaa eri havaintoihin erisuuruisia virheitä. Vertailuhavaintojen avulla tästä aiheutuva virhe voidaan poistaa ainakin osittain.
- Kohde on niin kirkas, että kuva saturoituu, jolloin ei saada lainkaan mittauservoa; tiedämme vain, että kirkkaus on jotakin rajaa

- Muut häiriöt (kosmiset säteet, huonosta sähköisestä eristyksestä johtuvat häiriöt).
- Datassa voi esiintyä myös yksittäisiä muista poikkeavia arvoja (outliers). Ne voivat olla mittausvirheitä, mutta voivat myös johtua jostakin ulkopuolisesta häiriöstä. Tällaisten poikkeamien syy olisi pyrittävä selvittämään, ettei vahingossa tulisi poistaneeksi poikkeavia mutta todellisia havaintoja.
- Moniin havaintomenetelmiin liittyy omia erityisiä toimenpiteitä, joilla alkuperäisestä raakadatasta saadaan varsinainen käsiteltävä aineisto. Tällainen on esimerkiksi kuvankäsittelyyn liittyvä kameran optiikan aiheuttamien geometristen vääristymien korjaaminen.

Kuvankäsittelyn tarkoitus on korjata näitä vikoja. Siihen kuuluu mm.

- datan hankkiminen ja tallennus
 - analogisen signaalin digitointi
 - datan tiivistäminen
 - datan tallennus (kuvien eri esitystavat)
- kuvan korjaaminen
 - geometriset muunnokset
 - kohinan vähentäminen
 - dekonvoluutio
 - yksityiskohtien vahvistaminen
- kuvan analyysi
 - reunojan havaitseminen
 - kohteiden tunnistaminen
 - luokittelu

Esitystavat

Kuva esitetään tavallisesti matriisina, jonka kukin alkio vastaa yhtä pikseliä. Toinen mahdollisuus olisi tallettaa esimerkiksi Fourier'n sarjan kertoimet. Molemmat ovat erikoistapauksia yleisestä kehitelmästä.

Olkoon f neliöllisesti integroitava funktio, joka on määritelty jossakin alueessa S . Haluamme esittää funktion kantafunktioiden ϕ_{mn} , $m, n = 0, 1, \dots$, avulla. Kantafunktiot ovat ortonormaaleja, jos

$$\int \int_S \phi_{mn}(x, y) \phi_{rs}^*(x, y) dx dy = \delta_{mr} \delta_{ns}.$$

Haemme kehitelmää

$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \phi_{mn}(x, y)$$

siten, että virhe e^2 (sampling error) on mahdollisimman pieni:

$$e^2 = \int \int_S \left| f(x, y) - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \phi_{mn}(x, y) \right|^2 dx dy,$$

Tämä toteutuu, kun

$$a_{mn} = \int \int_S f(x, y) \phi_{mn}^*(x, y) dx dy.$$

Kantafunktioiden joukkoa sanotaan täydelliseksi, jos jokaiselle neliöllisesti integroitavalle funktiolle virhe $e \rightarrow 0$, kun termien määrä kasvaa.

Standardiesitys: Kantafunktiot ovat suorakaiteita:

$$\phi_{mn}(x, y) = \begin{cases} \sqrt{\frac{MN}{AB}}, & \text{jos } x \in [\frac{mA}{M}, \frac{(m+1)A}{M}) \text{ ja } y \in [\frac{nB}{N}, \frac{(n+1)B}{N}), \\ 0 & \text{muuten.} \end{cases}$$

Kerroin a_{mn} on yksinkertaisesti funktion f keskiarvo suorakaiteessa, jossa $\phi_{mn} \neq 0$.

Fourier'n esitys: Kantafunktiot ovat

$$\phi_{mn}(x, y) = \frac{1}{\sqrt{AB}} e^{2\pi i(mx/A + ny/B)},$$

missä A ja B ovat kuvan leveys ja korkeus.

Optimaalinen esitys: Onko olemassa kantaa, jossa virhe e^2 on mahdollisimman pieni.

Virheen minimointi johtaa ominaisarvotehtävään

$$\int \int R_{ff}(x, x', y, y') \phi_{m,n}(x', y') dx' dy' = \gamma_{mn} \phi_{mn}(x, y),$$

jonka ratkaisuna saadaan optimaaliset kantafunktiot. Ne tunnetaan Karhusen ja Loèven funktioina.

Kvantisointi

Jatkuvat intensiteettiarvot täytyy muuntaa äärelliseksi määräksi harmaasävyjä. Olkoot eri sävyjen rajat z_k , $k = 1, 2, \dots, K + 1$. Jos intensiteetti on välillä $[z_k, z_{k+1})$, pikseli saa arvon q_k . Jos $p(z)$ on todennäköisyys, että intensiteetti on z , kvantisointivirhe on

$$\delta_q^2 = \sum_{k=1}^K \int_{z_k}^{z_{k+1}} (z - q_k)^2 p(z) dz.$$

Minimoidaan tätä:

$$\begin{aligned} \frac{\partial \delta_q^2}{\partial z_k} &= (z_k - q_{k-1})^2 p(z_k) - (z_k - q_k)^2 p(z_k) = 0, k = 2, 3, \dots, K, \\ \frac{\partial \delta_q^2}{\partial q_k} &= -2 \int_{z_k}^{z_{k+1}} (z - q_k) p(z) dz = 0, k = 1, 2, \dots, K. \end{aligned}$$

eli

$$\begin{aligned} z_k &= \frac{q_{k-1} + q_k}{2}, \\ q_k &= \frac{\int_{z_k}^{z_{k+1}} z p(z) dz}{\int_{z_k}^{z_{k+1}} p(z) dz}. \end{aligned}$$

Jos $p(z)$ on vakio, saadaan tasaväliset rajat.

Käytännössä kvantisointi tapahtuu suoraan laitteistotasolla, jolloin rajat ovat yleensä tasavälisiä, vaikka se ei antaisikaan optimaalista lopputulosta.

Tiivistäminen

Tiivistämistä (compression) tarvitaan tilantarpeen vähentämiseksi ja myös tiedonsiirron nopeuttamiseksi.

Usein naapuripikselit ovat korreloituneita, joten tilaa hukataan tallettamalla tietoa, jonka informaatioisisältö on pieni.

Tiivistämistä voi tehdä monella tavalla:

- Transformaatiomenetelmissä kuva esitetään kombinaationa sopivista kantafunktioista.
- Sävyjen koodausta voidaan muuttaa käyttämällä hyväksi jakautuman ei-satunnaisuutta.
- Kuvataan kuvaa jollakin fraktaaliprosessilla.

Virheetön (tietoa hukkaamaton, kohinaton) menetelmä säilyttää kaiken alkuperäisen kuvan informaation, joten kuva on rekonstruoitavissa tarkasti alkuperäiseen muotoon.

Tietoa hukkaava (kohinainen) menetelmä säilyttää vain oleellimmat piirteet; yleensä menetetään pienimmät yksityiskohdat.

Jos harmaatason q_i todennäköisyys on p_i , kuvaan sisältyvän informaation määrä on

$$H = - \sum_{i=1}^K p_i \log_2 p_i.$$

Tämä on alaraja sille, kuinka monta bittiä pikseliä kohti tarvitaan, jotta alkuperäinen kuva pystytään säilyttämään.

Transformaatiotiivistys: Esitetään kuva kantafunktioiden kombinaationa ja säilytetään vain merkittävimmät termit. Järkevien kantafunktioiden valinta edellyttää kuitenkin tietoa itse kuvasta.

Karhusen ja Loèven muunnos (KL-muunnos): Jos kuva esitetään KL-funktioiden kombinaationa, kertoimet ovat riippumattomia. Kantafunktioiden etsiminen on periaatteessa raskas ominaisarvotekniikka. Jos autokorrelaatiofunktio oletetaan tunnetuksi, voidaan kantafunktiolle johtaa lausekkeet.

Fourier'n muunnos: Helppo laskea, mutta yleensä tarvitaan enemmän termejä kuin KL-muunnoksessa, jotta päästäisiin samaan kuvan laatuun.

Hadamardin muunnos: Hyvin tehokas muunnos, jonka kantamatriisit koostuvat pelkästään luvuista $+1$ ja -1 (ja normitusvakiosta). $N \times N$ kuvalle, missä N on kakkosen potenssi kantamatriisit ovat

$$\phi^{(u,v)}(m,n) = \frac{1}{N}(-1)^{b(u,v,m,n)},$$

missä

$$b(u,v,m,n) = \sum_{k=0}^{\log_2 N - 1} (b_k(u)b_k(v) + b_k(m)b_k(n)),$$

ja $b_k(x)$ on luvun x k :s bitti.

Jakson pituus -koodaus (Run length coding): Pikselit korvataan pareilla (v,n) , missä v on pikselin arvo ja n on niiden peräkkäisten pikselien määrä, joilla on sama arvo. Säästää paljon tilaa, jos kuvassa on suuria tasaisia pintoja.

Huffman-koodaus: Usein esiintyvät arvot korvataan lyhyillä koodailla ja harvinaiset pitemmällä.

Fraktaalikoodaus: Luonnossa esiintyvät kohteet ovat harvoin yksinkertaisia geometrisia objekteja, vaan niillä on itsesimilaarinen rakenne (yksityiskohta suurennettuna näyttää samantapaiselta kuin suurempi osa kohdetta). Fraktaalitiivistys perustuu iteroituihin funktiosysteemeihin (IFS, iterated function systems, ks. esim. Barnsley: *Fractals everywhere*, Academic Press 1988.). IFS on satunnaiskulku, jonka kussakin pisteessä valitaan satunnaisesti (mutta sopivilla painoilla) jokin mahdollisista muunnoksista seuraavan pisteen laskemiseksi. Laskennallisesti erittäin raskas, mutta voi tiivistää mutkikkaan metsää, kukkia tai pilviä esittävän kuvaan muutamaan tuhanteen tavuun.


```

program ifs
! Plot attractor of an iterative process
! | x | | a b || x | | e |
! | | = | | | | + | |
! | y | | c d || y | | f |
! There are maxn different sets for the coefficients
! p(i) is the probability that set i will be used
! sp is the cumulative probability distribution
use plot
implicit none
integer, parameter:: maxn=4, npoints=30000
real, dimension(maxn) :: a,b,c,d,e,f,p
real, dimension(0:maxn) :: sp
real, dimension(npoints) :: xx, yy
integer i

a = (/ 0.00, 0.85, 0.20, -0.15 /)
b = (/ 0.00, 0.04, -0.26, 0.28 /)
c = (/ 0.00, -0.04, 0.23, 0.26 /)
d = (/ 0.16, 0.85, 0.22, 0.24 /)
e = (/ 0.00, 0.00, 0.00, 0.00 /)
f = (/ 0.00, 1.60, 1.60, 0.44 /)
p = (/ 0.01, 0.85, 0.07, 0.07 /)
sp = 0.0
do i=1, maxn
    sp(i) = sp(i-1)+p(i)
end do
sp(maxn) = 1.01
call iterate_2(xx, yy, npoints)
call init("xx.ps", 5.0, 5.0, 40.0, 40.0)
call plotpoints(xx, yy, npoints)
call finish()
contains
subroutine iterate_2(xx, yy, npoints)
! random iteration algorithm
integer, intent(in) :: npoints
real, dimension(npoints) :: xx, yy
integer i, k
real x, y, xprev, yprev, r
xprev=0.0 ; yprev=0.0
do i=1,npoints
    call random_number(r)
    do k=1,maxn
        if (r < sp(k)) exit
    end do
    x=a(k)*xprev+b(k)*yprev+e(k)
    y=c(k)*xprev+d(k)*yprev+f(k)
    xx(i) = x ; yy(i) = y
    xprev=x ; yprev=y ;
end do
end subroutine
end program

```

Aallokkeet

Aallokkeita (wavelets) voidaan käyttää lokaalina analyysimenetelmänä. Aallokkeiden voidaan ajatella hajoitettavan datan samanaikaisesti eri mittaisiin aaltopaketteihin. Aallokkeita on käytetty tehokkaasti esim. kuvaprosessoinnissa, jossa aallokkein esitetty kuva on paljon kompaktimpi kuin varsinaisin mittauspistein esitetty kuva. Yleisesti datan kompressoinnissa voidaan käyttää aallokkeita.

Aallokkeet ovat kaksi-indeksisiä yhden muuttujan funktioita

$$g_x = \frac{1}{\sqrt{|a|}} g\left(\frac{x-b}{a}\right), \quad a \neq 0,$$

missä g :lle pätee

$$C = 2\pi \int_{-\infty}^{\infty} |\xi|^{-1} |F(g)| d\xi < \infty,$$

missä $F(g)$ on g :n Fourier'n muunnos.

Määritelmästä johtuu että $F(g)$ on origossa arvoltaan nolla, ja aallokkeen keskiarvon on oltava

$$\int_{-\infty}^{\infty} g(x) dx = 0.$$

Tästä johtuu että g :n täytyy vaihtaa merkkiä jossakin kohtaa ja että $g(x)$ lähestyy nollaa x :n kasvaessa tai pienetessä rajatta.

Aallokkeiden äitifunktiosta $\psi(t)$ voidaan muodostaa perhe aallokkeita, joita voidaan skaalata ja siirtää:

$$\psi_{s,l}(t) = \frac{1}{2^{s/2}} \psi\left(\frac{t}{2^s} - l\right)$$

Tässä s on skaalausindeksi ja l paikkaindeksi. Aallokkeen leveys riippuu vain suureesta

$$S = 2^s.$$

Suuretta S voi tavallaan pitää analogisena Fourier-avaruuden taajuudelle.

Paikkasuurelle l sitä vastoin ei löydy vastinetta Fourier-avaruudesta. Juuri tämä suureen olemassaolo ja se, että $\psi_{s,l}(t)$ voidaan valita sopivasti ortonormaaliksi funktiojoukoksi tekee aallokkeista tehokkaita analyysityökaluja. Tarkastellaan seuraavaksi vain diskreettejä

aallokkeita, joissa analysoitavana on tasavälinen havaintopisteistö, ja oletetaan, että havaintopisteitä X_i on 2^N verran, missä N on kokonaisluku.

Nyt voidaan määritellä aallokemuunnos

$$W_{s,l}^x = \sum_{i=1}^N \psi_{s,l}(i) X_i,$$

$W_{s,l}^x$ mittaa muutoksien suuruutta skaalalla s kohdassa l . Indeksien $s = 0$ voidaan ajatella viittaavan suurimpaan tai pienimpään skaalaan. Jos valitsemme sen viittaavan pienimpään mitattavaan skaalaan, niin silloin kasvavia skaaloja merkitään suurenevin positiivisin indekseihin.

Jos valitut aallokkeet muodostavat ortonormaalin joukon, voimassa on myös käänteisoperaatio

$$X_i = \sum_{s,l} \psi_{s,l}(i) W_{s,l}^x.$$

Aallokkeiden paikallisuuden vuoksi ne ovat tehokkaita löytämään m.m. teräviä epäjatkuvuuskohtia ja singulariteettejä (Haarin aallokkeet) tai lyhytkestoisia oskillaatioita (Moreletin aallokkeet).

Haarin aallocke

Haarin aallokkeen äiti on

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}t_0 \\ -1, & \frac{1}{2}t_0 \leq t < t_0 \\ 0, & \text{muualla.} \end{cases}$$

Itse aallocke on pienimmällä mittakaavalla muotoa

$$h_n = \begin{cases} h_0 & n = 1 \\ -h_0 & n = 2 \\ 0 & \text{muualla.} \end{cases}$$

Kerroin $h_0 = \frac{1}{\sqrt{2}}$ on normitusvakio. Seuraavan skaalan aallocke on kaksi kertaa pidempi ja amplitudiltaan $\frac{1}{\sqrt{2}}$ -kertainen. Näin kunkin aallokkeen "teho" on yhtäsuuri.

Haarin aallokkeet muodostavat ortonormaalin joukon.

Daubechies'n aallokeet

Daubechies'n aaloke D_4 on

$$\psi_{s,l}(x) = \frac{1}{2^{s/2}} \psi\left(\frac{x}{2^s} - l\right),$$

missä $\psi(x)$ on D_4 :n iti

$$\psi(x) = c_1\phi(2x) - c_0\phi(2x-1) - c_2\phi(2x+1) - c_3\phi(2x+2),$$

ja $\phi(x)$, D_4 aallokkeen isä, joka on määritelty

$$\phi(x) = c_0\phi(2x) + c_1\phi(2x-1) + c_2\phi(2x-2) + c_3\phi(2x-3).$$

Kertoimet c_i ovat:

$$c_0 = \frac{1}{4}(1 + \sqrt{3}), c_1 = \frac{1}{4}(3 + \sqrt{3}), c_2 = \frac{1}{4}(3 - \sqrt{3}), c_3 = \frac{1}{4}(1 - \sqrt{3}).$$

Moreletin aaloke

$$\psi_{s,l} = \exp\left(-ic\left(\frac{x-l}{s}\right) - \frac{1}{2}\left(\frac{x-l}{s}\right)^2\right).$$

Jos c on pieni, aalopaketti on leveä, mutta vähemmän herkkä kohinalle. Kompromissina voidaan pitää $c = 2\pi$. Jos merkitään $s = 1/\nu$, huomataan samankaltaisuus Fourier'n muunnoksen kanssa:

$$\psi_{s,l} = e^{-i2\pi\nu(t-t_0)} \cdot e^{-\frac{1}{2}\nu(t-t_0)^2}$$

Meksikolainen hattu

$$\psi(x) = (1 - x^2)e^{-\frac{1}{2}x^2}.$$

Meksikolaishatut eivät muodosta ortonormaalijoukkoa, eikä niillä näinollen ole varsinaista äitiaaloketta. Ne ovat kuitenkin matemaattisen yksinkertaisuutensa vuoksi olleet suosittuja.

Kuvaformaatteja

eps (Encapsulated PostScript) Tekstiä, oikeastaan ohjelmointikieli, helppo siirtää. Sopii hyvin vektorigrafikkaan, huonommin pikselikuviin. Kielen kehittyneiden rakenteiden avulla monimutkaisiakin kuvia voi toteuttaa hyvin lyhyesti. Esitystapa on riippumaton tulostuslaitteen resoluutiosta. Kuvan rasterointi on tulostimen tehtävä. Adoben lisenssi nostaa PS-tulostimien hintaa.

tif (Tagged Image format) Pikselikuva, ei pakattu, tiedostot isoja.

gif (CompuServe graphics interchange format) Pakattu kuva, ei hukkaa informaatiota.

jpeg (Joint Photographic Experts Group file interchange format) Informaatiota hukkaava pakattu kuva. Tarkkuus kärsii hieman, mutta tilansäästö huomattava.

mpeg (Motion Picture Experts Group file interchange format) Liikkuvan kuvan pakkausmenetelmä. Tietyn pikselin väri ei yleensä muutu kahden kuvan välillä, joten koko kuvan sijasta talletetaan vain niissä tapahtuvat muutokset.

pdf (Portable Document Format) Lähinnä julkaisualan uusi standardi. Tiedosto ASCII-tekstiä, joten helposti siirrettävissä.

Linuxin `convert` muuntaa ainakin noin 75 eri kuvaformaattia toisikseen.

```
66146 Apr 15 13:40 forest.bmp ==>
524554 Apr 15 13:42 forest.tif
218378 Apr 15 13:41 forest.eps
75115 Apr 15 13:40 forest.gif
24881 Apr 15 13:43 forest.jpg
```

```
826 Dec 17 1997 color.ps ==>
2908547 Apr 15 13:59 color.tif
1454166 Apr 15 14:00 color.bmp
551223 Apr 15 14:01 color.gif
137581 Apr 15 14:17 color.eps
47422 Apr 15 13:59 color.jpg
```

Geometriset muunnokset

Geometrinen muunnos kuvaa pisteen (x, y) toiselle pisteelle (x', y') . Tietokonegrafikassa on käytännöllistä esittää pisteitä sarakevektoreilla, joissa on yksi ylimääräinen alkio. Tason piste (x, y) esitetään muodossa

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Muunnokset voidaan silloin esittää 3×3 -matriiseina:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

1) translaatio

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}.$$

2) skaalaus

$$\mathbf{M} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

3) kierto

$$\mathbf{M} = \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Näitä voi käyttää viivapiirroksille, mutta EI pikselikuville.

Pikselikuvien geometriset muunnokset

1) Suora muunnos:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \rightarrow \mathbf{M} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}.$$

Ei käy, sillä

- jotkin pikselit eivät ehkä saa lainkaan arvoa
- useat pikselit voivat kuvautua samalle pikselille

2) Käänteinen muunnos: kullekin (x', y') etsitään pikseli (x, y) , joka kuvautuu pisteeseen (x', y') ja annetaan sen arvo pikselille (x', y') . Parempi, ei jätä tyhjiä pikseleitä. Edelleen jotkut alkuperäisen kuvan pikselit voivat kuvautua useammalle pikselille, jotkin eivät millekään.

3) Käänteinen interpolaatio: haetaan (x, y) kuten edellä; x eivät y yleensä ole kokonaislukuja, joten interpoloidaan pistettä (x, y) vastaava pikselin arvo. Toimii yleensä hyvin, varsinkin jos kuva on kohtuullisen jatkuva. Terävät reunat voivat muuttua epätarkoiksi.

4) Uudelleen konstruointi: esitetään alkuperäinen kuva funktiona, jonka avulla lasketaan uudet pikselien arvot. Työlästä, mutta antaa parhaan tuloksen.

Suotimet (filtterit)

Suodin on tavallisesti lokaali muunnos

$$f(x, y) \rightarrow f'(x, y) = \int_S f(x - x', y - y')g(x', y') dx' dy',$$

missä S is pieni origin ympäristö. Monet suotimet ovat konvoluutiomuotoa, jolloin voidaan käsitellä joko alkuperäistä kuvaa tai sen Fourier'n muunnosta.

Ei-rekursiivinen suodin: kaikki uudet arvot lasketaan pelkästään alkuperäisen kuvan avulla.

Rekursiivinen suodin: korvataan kukin pikseli uudella arvolla heti kun se on laskettu; muunnettu kuva voidaan tallettaa alkuperäisen päälle.

Alipäästösuodin: poistetaan tiettyä rajaa korkeammat taajuudet. Helpoin toteuttaa Fourier'n muunnoksen avulla. Suotimella on tasoitava vaikutus. Koska rect-funktion Fourier'n muunnos on sinc-funktio, alkuperäisen kuvan tasoitus (konvoluutio rect-funktion kanssa) ei poista korkeita taajuuksia kokonaan.

Alipäästösuotimia:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{5} \begin{bmatrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{bmatrix} \quad \frac{1}{6} \begin{bmatrix} & 1 & \\ 1 & 2 & 1 \\ & 1 & \end{bmatrix}$$

Ylipäästösuodin: poistetaan matalat taajuudet. Kuvaa terävöittävä vaikutus. Vastaa derivointia.

Esim. Laplacen operaattori on ylipäästösuodin

$$\begin{aligned} \nabla^2 f(i, j) &= \Delta_x^2 f(i, j) + \Delta_y^2 f(i, j) \\ &= (f(i + 1, j) + f(i - 1, j) + f(i, j + 1) + f(i, j - 1)) - 4f(i, j). \end{aligned}$$

eli

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}$$

Ajatellaan, että kuvaa on sumennettu lisäämällä kuhunkin pikseliin osa ϵ naapuripikselien summasta:

$$p(i, j) = f(i, j) + \epsilon n(i, j),$$

missä

$$n(i, j) = f(i + 1, j) + f(i - 1, j) + f(i, j + 1) + f(i, j - 1).$$

Vähennetään Laplace-operaattoriin verrannollinen lauseke:

$$\begin{aligned} p(i, j) - k\nabla^2 p(i, j) &= (1 + 4k)p - k(p(i + 1, j) + p(i - 1, j) + p(i, j + 1) + p(i, j - 1)) \\ &= (1 + 4k)f(i, j) + (1 + 4k)\epsilon n(i, j) - kn(i, j) \\ &\quad - k\epsilon(n(i + 1, j) + n(i - 1, j) + n(i, j + 1) + n(i, j - 1)). \end{aligned}$$

Jos k ja ϵ ovat pieniä, viimeinen termi voidaan unohtaa. Jos $k = \epsilon/(1 - 4\epsilon)$, kaksi muuta termiä katoaa, jolloin

$$p(i, j) - k\nabla^2 p(i, j) = (1 + 4k)f(i, j),$$

joten alkuperäinen kuva on rekonstruoitu vakiokerrointa vaille.

Mediaanisuodin: Keskiarvo pienentää piikkejä (kuten kosmisia säteitä), mutta ei poista niitä, vaan levittää niiden energian laajemmalle alueelle. Parempi ratkaisu on korvata pikseli pienen ympäristön mediaanilla. Tämä ei sumenna reunoja eikä vaikuta monotonisesti muuttuviin alueisiin.

Dekonvoluutio

Oletetaan, että psf h on translaatioinvariantti. Havaitun kuvan p ja alkuperäisen kuvan f pätee silloin

$$p(x, y) = \int \int h(x - x', y - y') f(x', y') dx' dy' + \nu(x, y),$$

missä ν on kohina. Laskemalla kummankin puolen Fourier'n muunnos saadaan

$$P(u, v) = H(u, v)F(u, v) + N(u, v),$$

missä H on kuvanmuodostusjärjestelmän siirtofunktio (transfer function).

Periaatteessa tästä voidaan ratkaista alkuperäinen kuva:

$$F(u, v) = \frac{P(u, v)}{H(u, v)} - \frac{N(u, v)}{H(u, v)}.$$

Tavallisesti h on suunnilleen Gaussin profiili, jonka Fourier'n muunnos on myös Gaussin profiili, siis lähes nolla pientä aluetta lukuunottamatta.

Ellei kohinaa ole, $P = 0$ kun $H = 0$, joten suhde P/H on epämääräinen. Kun mukana on kohinaa, H voi paikoitellen olla paljon pienempi kuin N , mikä aiheuttaa piikkejä. Kuvan korjaaminen suotimella $M(u, v) = 1/H(u, v)$ lisää sen kohinaa.

Parempi suodin on

$$M(u, v) = \begin{cases} 1/H(u, v), & \text{if } u^2 + v^2 \leq r^2, \\ 1 & \text{muuten,} \end{cases}$$

missä r on riittävän pieni, jotta H pysyy varmasti nolasta poikkeavana.

Pienimmän neliösumman mielessä optimaalinen suodin on Wienerin suodin:

$$M(u, v) = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_{\nu\nu}(u, v)/S_{ff}(u, v)}.$$

Jos kohina on valkoista kohinaa, $S_{\nu\nu}$ on vakio ja likimain

$$S_{\nu\nu}(0, 0) = \int_{-\infty}^{\infty} R_{\nu\nu}(x, y) dx dy.$$

Käytännössä voidaan kokeilla, millä arvolla saadaan paras tulos.

Toinen menetelmä perustuu entropian käsitteeseen

$$H_f = - \sum_m \sum_n f(m, n) \ln f(m, n).$$

Jos kokonaisenergia

$$f_{\text{tot}} = \sum_m \sum_n f(m, n)$$

pysyy vakiona, entropia on sitä pienempi, mitä enemmän arvot $f(m, n)$ vaihtelevat.

Kohina voi saada myös negatiivisia arvoja. Lisätään kohinaan vakio C siten, että $\nu(m, n) + C$ on aina positiivinen:

$$H_\nu = - \sum_m \sum_n (\nu(m, n) + C) \ln(\nu(m, n) + C),$$

Maksimientropiakorjaus tehdään maksimoimalla lauseketta

$$H = H_f + \alpha H_\nu,$$

missä α on tasoituksen määrää kuvaava vakio. Lisäksi tarvitaan kaksi rajoitusta:

- 1) Kokonaisenergian f_{tot} täytyy säilyä vakiona.
- 2) Havaitun kuvan p täytyy olla alkuperäisen kuvan f ja psf:n h konvoluutio + kohina:

$$p(q, r) = \sum_m \sum_n h(q - m, r - n) f(m, n) + \nu(m, n).$$

Ratkaisuna saadaan f ja ν .