10. Suodattimet

Konvoluutio

Yli- ja alipäästösuodattimet

Reunan havaitseminen

Epälineaarisia suodattimia

Naapureiden vaikutus

- Edellä pikselikorjaukset riippumattomia naapureista
- Jos halutaan tehdä lokaalisia korjauksia, pitää naapuripikselien sävyt ottaa huomioon.
- Operaatioiden perustyypit:
  - Pikselin uusi arvo on sen ja naapurien painotettu summa (myös negatiiviset painot sallitaan).
  - Pikselin uusi arvo valitaan sen vanhan arvon ja ympäristön arvojen joukosta.

Konvoluutio

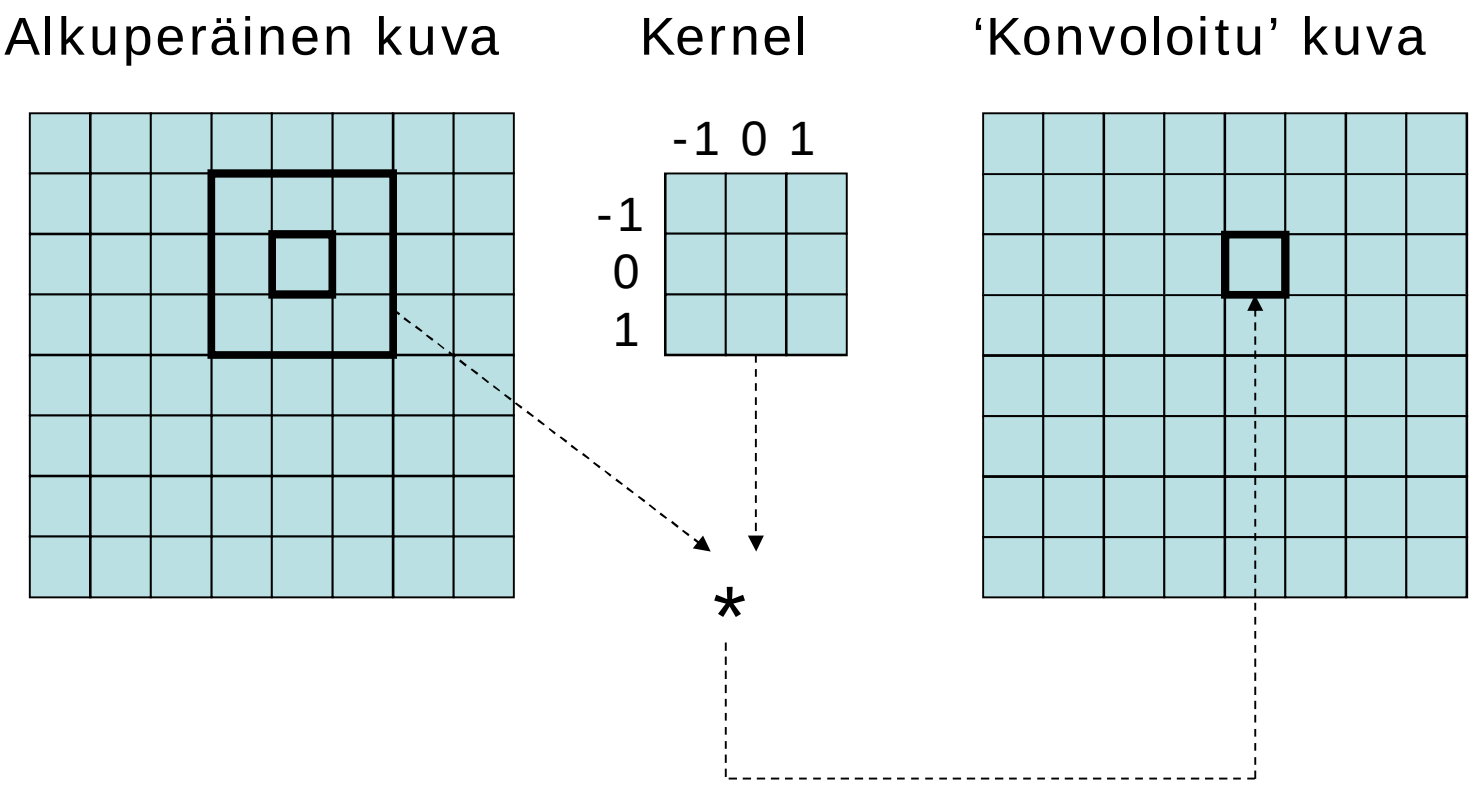
- Suodatuksen perusoperaatio
- Käyttötarkoituksia:
  - Kohinan vähennys
  - Kohteiden reunojen korostus
- Kyseessä *lineaarinen* muunnos:
  - $C(k \cdot f(x,y)) = k \cdot C(f(x,y))$
  - $C(f_1(x,y) + f_2(x,y)) = C(f_1(x,y)) + C(f_2(x,y))$

  (f, f_1, f_2=kuvafunktio, C=konvoluutio)
  Esim. kontrastin säätö voidaan tehdä ennen tai jälkeen konvoluution.

Konvoluution laskenta

- Konvoluutio = pikselin ja sen ympäristön sävyarvojen painotettu summa; se lasketaan erikseen kuvan kaikille pikseleille.
- Ympäristö on suorakaide (yleensä neliö), jonka keskellä tarkasteltava pikseli on
- Summan painokertoimet:
  matriisi, ns. konvoluution *kernel* (*ydin*; maski)
- Tyypillinen ytimen koko: 3 x 3, myös suurempia käytetään
- Leveys & korkeus yleensä pariton, jotta matriisi on symmetrinen keskipisteen suhteen.

Konvoluution idea



Konvoluution kaava

- Merk. $h(j, k)$ = ydinmatriisi
- Matriisin leveys = m ja korkeus = n
- Pikselin $f(x, y)$ *konvoloitu* arvo:

$$g(x,y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j,k)\, f(x-j, y-k)$$

missä $m_2 = \lfloor m/2 \rfloor$ ja $n_2 = \lfloor n/2 \rfloor$

- Merk. $g(x, y) = h * f(x, y)$,
  ja koko kuvalle $g = h * f$

Korrelaatio

- Lähes identtinen konvoluution kanssa (huomaa ’+’ eikä ’-’):

$$g(x,y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j,k)\, f(x+j, y+k)$$

- Käyttö samankaltaisuusvertailuissa:
  - Etsittävä malli (template) *h* vastaa ydintä.
  - Malli yleensä pienempi kuin kohdekuva mutta suurempi kuin konvoluution ydin.
  - Ongelma: suosii vaaleita alueita (korjaantuu normalisoinnilla)

Konvoluution laskennasta

- Ydinmatriisin kerroin on aina vastaavan pikselin *vastakkaisella* puolella.
- Jos ydintä käännetään 180^{o}, niin kyseessä on korrelaatio-operaatio.
- Jos ydin on symmetrinen 180^{o}:n rotaatiolle (kuten usein on), niin konvoluutio = korrelaatio
- Konvoluutio vastaa kertolaskua frekvenssi-alueella (→ *Fourier-muunnos*):
  Fourier(h * f) = Fourier(h) · Fourier(f)

Konvoluutiotuloksen arvoalue

- Tulos ei ole välttämättä oikealla välillä (harmaasävykuville 0..255), vaan tarvitaan lisämuunnos
- Jos ytimen alkiot ≥ 0, niin *normalisointi* riittää (siten että kertoimien summa = 1), esim.

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

(kyseessä aritmeettinen keskiarvo)

Kuvan reunojen käsittely

- Ydin osittain ulkopuolella; ratkaisuja:
  - Reunoille oletusarvo (’kehys’)
  - Reunat jätetään pois (kuva pienenee)
  - Reunat kopioidaan käsittelemättä
  - Ytimen kutistus (ei sovi kaikille ytimille)
  - Reunojen peilaus ulospäin
  - Syklinen indeksointi:
    - Vasen ja oikea reuna vierekkäin, samoin ylä- ja alareuna.
    - Kuva toistaa itseään äärettömyyksiin.
    - Jaksollisuus matemaattisesti luonteva oletus (sovelletaan mm. Fourier-muunnoksen yhteydessä)

Lisää konvoluution laskennasta

- Raskas operaatio
- *n* x *n* –ydin: O($n^2 \cdot$ pikselimäärä) kerto- ja yhteenlaskua
- Esimerkki nopeasta konvoluutioytimestä:

1	2	1
2	4	2
1	2	1

Ei kerto- eikä jakolaskuja!

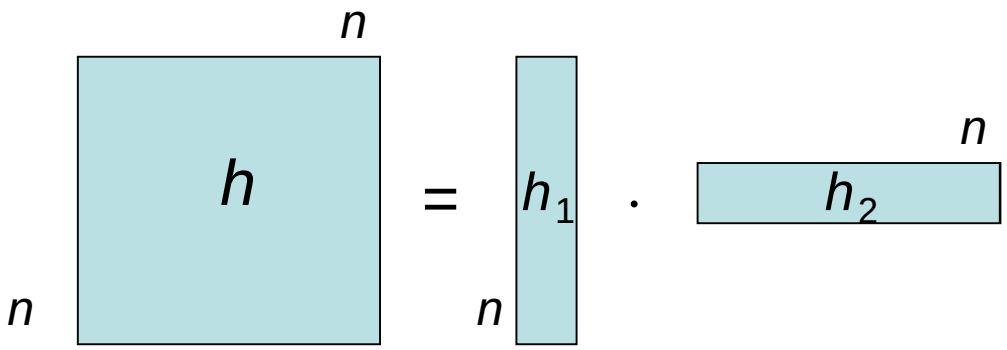
$2 \cdot f(x,y) = f(x,y)<<1$

$4 \cdot f(x,y) = f(x,y)<<2$

$[\Sigma(h(x,y) \cdot f(x,y))]/16 = [\Sigma(h(x,y) \cdot f(x,y))]>>4$

Separoituva ydin

- n x n –matriisi h on separoituva, jos se on n-vektorien h_1 ja h_2 tulo



- h_1 ja h_2 ovat 1-dimensioisia ytimiä.
- Sovelletaan ensin sarakkeittain, sitten riveittäin (tai päinvastoin).
- Kompleksisuus O(n) pikseliä kohti.
- Edullinen suurikokoisille ytimille

Konvoluution käytännön nopeutus

- Laitteistotason toteutus (esim. reaaliaika-sovelluksia varten)
- Moniprosessorikoneet:
  - Kuvan lohkominen; lohkojen jako eri prosessoreille.
  - Kerto- ja yhteenlaskut rinnakkain:
    Tulo lisätään summaan samalla kun lasketaan seuraavaa tuloa.
- Java2D: ConvolveOp-luokka: Kriittiset osat tehokkaasti toteutettu.

Konvoluutio Javalla

- Java2D:n luokat:
  - Kernel (luonnissa tarvitaan dimensiot ja ytimen kertoimet sisältävä float-vektori)
  - ConvolveOp (implementoi BufferedImageOp-rajapinnan)
- Konvoluution toteutus:

```
Kernel ker = new Kernel(w, h, kernVec);
ConvolveOp op = new ConvolveOp(ker);
BufferedImage tulos = op.filter(kuva, null);
```

- Reunan käsittely: nollaus tai ei-operaatiota

```
new ConvolveOp(ker, edgecond, renderinghints)
```

Lineaarinen suodatus

- Spatiaalisten frekvenssien säätö
  (= miten tiheästi pikselien kirkkaus voi vaihdella)
- Vastaa signaalinkäsittelyn suodatuksia:
  osa taajuuksista eliminoidaan
- Suodattimien perustyypit:
  ali- ja *ylipäästösuodatin*
- Toteutus erilaisilla konvoluutioytimillä

Alipäästösuodatus

- Eliminoi korkeat spatiaaliset frekvenssit
- Tavoitteet: kohinan vähennys; kuvan pehmennys (myös sumentaa)
- Toteutus: Konvoluution ydin, jonka kertoimet > 0.
- Esim. 9 pikselin keskiarvosuodatin:
- Ei tarvita kertolaskuja
- Voidaan toistaa

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Gaussin suodatin

- Alipäästösuodatin; pehmentää
- Ytimen kertoimet erisuuret; otoksia Gaussin funktiosta (2-ulotteinen):

$$h(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

  Lisäksi tarvitaan normalisointi.
- Parametri σ = hajonta
  - Pieni σ: Riittää pieni ydinmatriisi
  - Iso σ: Tarvitaan isompi ydinmatriisi

Esimerkki Gaussin suodatuksesta (säde = 2)

Photo Credit: US Fish and Wildlife Service
Archive: Gimp-Savvy

Esimerkki Gaussin ytimestä (7 x 7)

Normalisoitu ydin, $\sigma = 1.0$:

0.000	0.000	0.001	0.002	0.001	0.000	0.000
0.000	0.003	0.013	0.022	0.013	0.003	0.000
0.001	0.013	0.059	0.097	0.059	0.013	0.001
0.002	0.022	0.097	0.159	0.097	0.022	0.002
0.001	0.013	0.059	0.097	0.059	0.013	0.001
0.000	0.003	0.013	0.022	0.013	0.003	0.000
0.000	0.000	0.001	0.002	0.001	0.000	0.000

Gaussin suodattimen ominaisuuksia

- Ydin symmetrinen keskipisteen suhteen: Kaikki suunnat suodattuvat samanveroisesti.
- Reunoilla arvot lähellä nollaa; vaikutus vähäisempi (kuten pitääkin).
- Separoituva: nopea laskea.
  Ed. sivun esimerkkiydin:

$h = h_1 h_1^T$, missä
$h_1 \approx (0.000, 0.054, 0.242, 0.399, 0.242, 0.054, 0.000)^T$

Ylipäästösuodatin

- Eliminoi matalat spatiaaliset taajuudet
- Tavoite: terävöinti, ääriviivojen korostus
- Konvoluution ydin sisältää positiivisia ja negatiivisia arvoja
- Suunnasta riippumaton ylipäästö: negatiiviset arvot reunoilla, positiiviset keskellä, ks. oik.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Skaalaus: 0 siirretään kohtaan 128: tasaiset alueet keskiharmaita, ääriviivat erottuvat

Korkeiden frekvenssien korostus

- Ääriviivojen (yleisemmin muutosten) korostus
- Alkuperäinen kuva + ylipäästön tulos
- Saadaan aikaan yhdellä konvoluutiolla, esim.
  - (*c*-8) edustaa alkuperäisen kuvan painoa
  - Lähellä 8:aa oleva arvo antaa voimakkaamman korostuksen.

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

’Epäterävä maski’

- ’Unsharp masking’; yleinen operaatio kuvankäsittelyohjelmissa
- Vähennetään kuvasta sen pehmennetty versio; korkeat frekvenssit jäävät jäljelle
- Huom! Kyseessä **terävöinti**.
- Voidaan valita:
  - terävöinnin määrä
  - terävöinnin säde
  - Kynnys: vaadittava ero ympäristöstä
- Liian voimakas terävöinti ja iso säde luovat ääriviivojen ympärille ei-toivotun haloilmiön.

Esimerkki: Ylikorostetut reunat

Photo Credit: US Fish and Wildlife Service
Archive: Gimp-Savvy

Reunojen (ääriviivojen) havaitseminen

- Yksi konvoluution pääsovellusalueista
- Reuna = harmaa-/värisävyjen äkillinen muutos
- Tavoitteena kiinnostavien kohteiden automaattinen rajaus
- Ongelmia: kohina, epäolennaiset rajat
- Reunojen havaitsemisen vaiheet:
  - Kohinan vähentäminen
  - Reunojen korostus
  - Relevanttien reunojen lokalisointi

Gradienttien käyttö

- Diskreetit approksimaatiot derivaatoille x- ja y-suunnissa:

$$g_x(x,y) = f(x+1, y) - f(x-1, y)$$
$$g_y(x,y) = f(x, y+1) - f(x, y-1)$$

- Gradientin pehmennys (kohinan poisto): lasketaan g_x:n ja g_y:n (painotetut) keskiarvot 3x3-kohdealueessa.
- Saadaan kaksi ydintä, joita sovelletaan erikseen.

Esimerkkejä ydinpareista

- Prewittin ytimet:

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sobelin ytimet:

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gradienttien yhteistarkastelu

- Edelliset ytimet riippuvat suunnasta.
- Gradienttivektori $g = [g_x, g_y]^T$
  - Suuruus:

$$g = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

  - Suunta:

$$\theta = \arctan(g_y / g_x)$$

- Gradientin suuruus $|g|$ mittaa reunan jyrkkyyttä.

Tärkeiden reunojen valinta

- Helppo ratkaisu:
  - Kynnysarvo |*g*|:lle
  - Saadaan reunakartta (binäärikuva)
- Mutta:
  - Reunat eivät ole aina teräviä; korkea kynnysarvo hävittää pehmeät reunat
  - Matalalla kynnysarvolla loiva reuna tuottaa leveän nauhan
  - Kohinakin voi synnyttää korkeita gradienttiarvoja

Esimerkki: gradienttiarvojen kynnystys

Alkuperäinen Gradientit Kynnystys

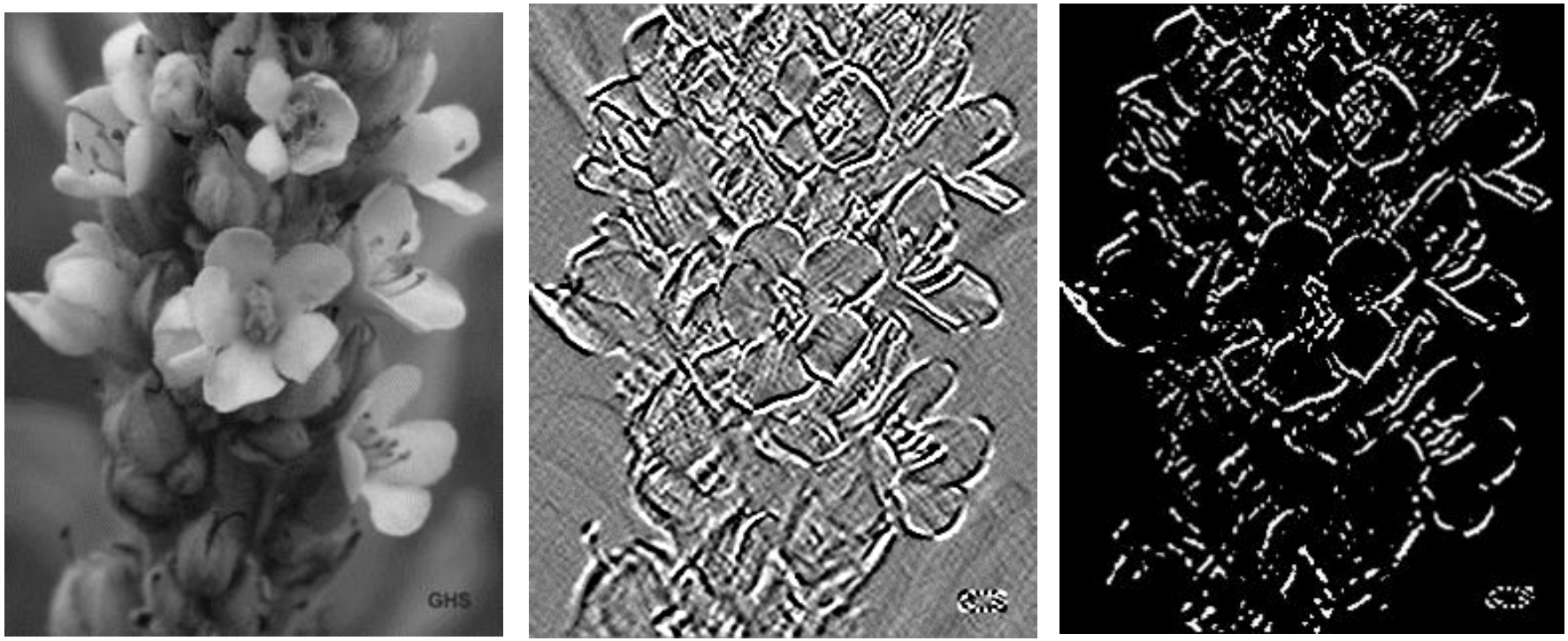


Photo Credit: US Fish and Wildlife Service
Archive: Gimp-Savvy

Laplace-suodatin

- Engl. ’Laplacian’
- Perustuu kuvafunktion toisen derivaatan approksimointiin (Prewitt- ja Sobel-ytimet approksimoivat 1. derivaattaa).
- Mittaa kuvafunktion jyrkkyyden muutos-nopeutta
- 2. derivaatta vaihtaa merkkiä reunan jyrkimmässä kohdassa:
  neg → pos tai pos → neg.

Laplace-suodatin (jatk.)

- Laplace-funktio: $$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Yksinkertainen approksimaatio 3 x 3 –konvoluutioytimellä:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Esimerkki: Laplace-approksimaatio

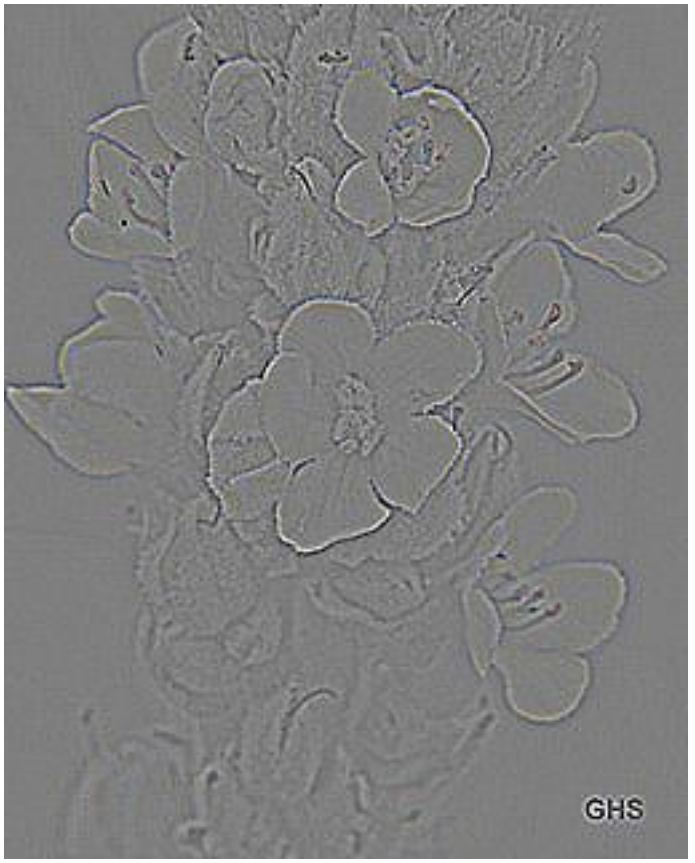


Photo Credit: US Fish and Wildlife Service

Archive: Gimp-Savvy

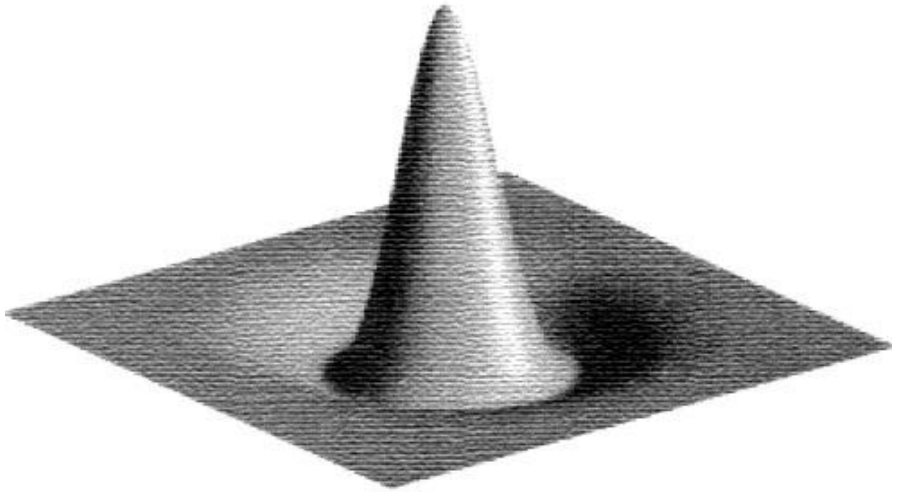
Laplace-suodatuksen yleistys

- Laplace-ongelma: herkkyys kohinalle
- Ratkaisu: yhdistetään Gaussin pehmentävä suodatus ja Laplace →
  LoG = ’Laplacian-of-Gaussian’

$$\nabla^2 h = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}}$$

  missä $r^2 = x^2+y^2$.
- LoG:n on ns. ’sombrero-funktio’. Sitä sovelletaan usein vastalukumuodossa (hattu ylösalaisin).

Sombrero-funktio ja sen 5x5-approksimaatio



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Huom! Summan oltava nolla

LoG-suodatuksen laskennasta

- σ:n arvo määrää suodattimen vaikutusalueen ja pehmennyksen määrän.
- Konvoluutioytimen leveyden tulisi olla > 6σ.
- LoG laskennallisesti kallis; voidaan approksimoida kahden ’erikokoisen’ (eri σ’t, suhde $\approx$ 1.6) Gaussin suodatuksen erotuksella.

LoG-suodatuksen käyttö

- Reunojen etsintä: reuna ilmenee etumerkin vaihtumisena.
- Etsitään erimerkkisiä naapuripikseleitä.
- Reunapikseliksi valitaan se, jolla on pienempi itseisarvo.
- Naapurit voivat sijaita eri suunnissa (LoG suunnasta riippumaton).
- *Terävöinti*: Lisätään suodatuksen tulos alkuperäiseen kuvaan (kuten aik.)

Canny-reunanhavaitsija

- Yrittää ratkaista kohinavaimennuksen ja reunan lokalisoinnin välisen ristiriidan
- Vaiheet:
  1) Gaussin alipäästösuodatus
  2) Reunan lokalisointi:
     a) Ei-maksimaalisten gradientti-pisteiden eliminointi
     b) Hysteresis-kynnystys

Ei-maksimaalisten gradienttipisteiden eliminointi

- Etsitään gradientin suunta:
  $\theta = \arctan(g_y/g_x)$
- Approksimoidaan suuntaa jollakin arvoista 0^o, 45^o, 90^o, 135^o (= θ').
- Jos gradientti g(x,y) < naapuripikselin gradientti suunnassa θ' tai suunnassa θ'+180^o, niin f(x,y)$\leftarrow$0.
- Kyseessä on reunojen ohennus; jäljelle jää jyrkin kohta.

Hysteresis-kynnystys

- Käytetään gradienteille kahta kynnysarvoa: T_{low} ja T_{high}.
- Etsitään ensin pikselit, joiden gradientin suuruus on > T_{high}.
- Laajennetaan saatujen pikselien joukkoa *naapureilla*, joiden gradientti > T_{low}.
- Edesauttaa ääriviivojen pysymistä yhtenäisinä.

Rank-suodatus

- *Epälineaarinen* menetelmä
- Perustuu naapuruston pikselien sävyjärjestykseen (’ranking’)
- Ei käytetä ydintä (konvoluutiota).
- Tarvitaan algoritmi, joka etsii k:nneksi suurimman arvon n:n joukosta.
- Kuvan reunojen käsittely tässä helppoa: käytetään vähemmän naapureita.

Mediaanisuodatin

- Yleisin rank-suodatin
- Pikselin arvoksi naapuruston keskimmäinen arvo
- Esim. 3 x 3 –alueella 5. suurin arvo
- Yleisesti n x n –alueella ($\lfloor n^2/2 \rfloor+1$)’s arvo
- Poistaa tehokkaasti *impulssikohinaa*, jossa pikselit saavat ääriarvoja 0 tai 255.

Mediaanisuodatin (jatk.)

- Ei sanottavasti sumenna kuvaa
- Kohinapikselien osuuden tulisi olla alle puolet naapurustosta; isompi suodatin usein takaa tämän
- Haitta: objektien reunat ja pienet yksityis-kohdat saattavat muuttua (jopa hävitä). Isolla suodattimella myös muutokset ovat isompia ja lopputulos maalauksenomainen.

Esimerkki mediaanisuodattimen vaikutuksesta

Alkuperäinen

Suodatettu (säde = 3)

Hybridisuodattimet

- ’α-trimmed mean filter’:
  - Lajitellaan (x,y):n naapuruston pikselit, merkitään tulos: $\{ f_1, \ldots, f_{n^2} \}$
  - Eliminoidaan ääripäät (2α kpl)
  - Lasketaan keskiarvo (= korjattu f(x,y)):

$$g(x,y) = \frac{1}{n^2 - 2\alpha} \sum_{i=\alpha+1}^{n^2-\alpha} f_i$$

- Erikoistapauksina saadaan:
  - α=0: keskiarvosuodatin
  - α=(n^2-1)/2: mediaanisuodatin

Minimal-Mean-Square-Error (MMSE) -suodatin

- Adaptiivinen: Toiminta muuttuu lokaalisten ominaisuuksien mukaan

$$g(x,y) = f(x,y) - \frac{\sigma_n^2}{\sigma_{x,y}^2}(f(x,y) - \overline{f}(x,y))$$

  - σ_n^2 = kohinan varianssin estimaatti
  - $\sigma_{x,y}^2$ = pisteen (x, y) todellisen ympäristön varianssi
  - $\overline{f}(x,y)$ = pisteen (x, y) ympäristön keskimäär. sävy

- Homog. alueella: $\sigma_{x,y}^2 \approx \sigma_n^2$, $g(x,y) \approx \overline{f}(x,y)$
- Reunoilla $\sigma_{x,y}^2$ suuri, $g(x,y) \approx f(x,y)$