

Signaler, information & bilder, föreläsning 13

Michael Felsberg och Maria Magnusson
Computer Vision Laboratory (Datorseende)
Department of Electrical Engineering (ISY)
michael.felsberg@liu.se,
maria.magnusson@liu.se



Översikt

- 2D signalbehandling (bildbehandling)
 - Den digitala bilden, färgtabeller
 - 2D kontinuerlig fouriertransform och 2D DFT
 - 2D sampling
 - 2D diskret faltning, linjär och cirkulär
- Teori: Kap. 2, 3.1-3.8, 3.10
- Bygger på Maria Magnussons föreläsningar



En bild är en 2D signal

Fig. 2.1

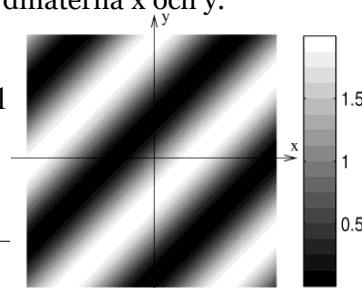
- 1D: $f(t)$ är en funktion f som beror av tiden t .
- 2D: $f(x,y)$ är en funktion f som beror av de spatiella (rums-) koordinaterna x och y .

Ex)

$$f(x, y) = \sin(x - y) + 1$$

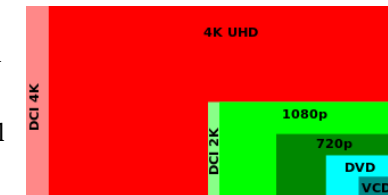
$$f(x, y) = 0 \Rightarrow \text{svart}$$

$$f(x, y) = 2 \Rightarrow \text{vitt}$$



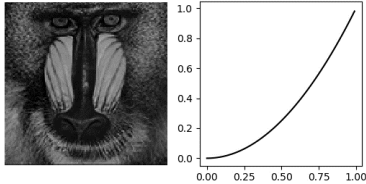
För en digital bild gäller

- En digital bild är en samplad 2D-funktion.
- Samplen kallas pixlar (pixels, picture elements).
- Antalet pixlar = bildens storlek.
- Vanliga storlek:
 - $512 \times 512 = 2^{18} = 0.25$ Mpixel
 - HD
 1920×1080
 $\approx 2^{21} = 2$ Mpixel
 - UHD ("4K")
 3840×2160
 $\approx 2^{23} = 8$ Mpixel

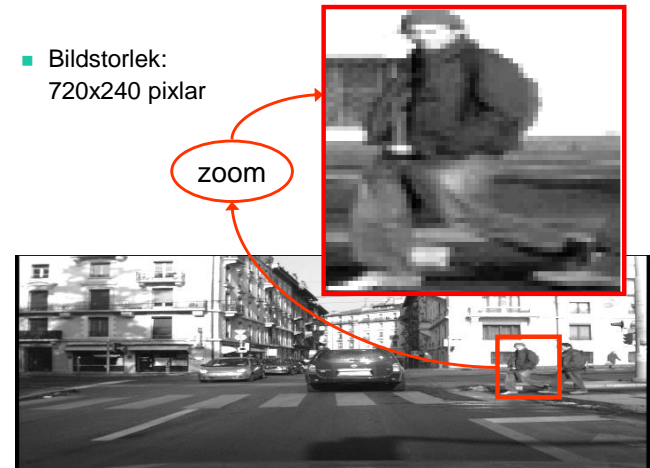


För en digital bild gäller

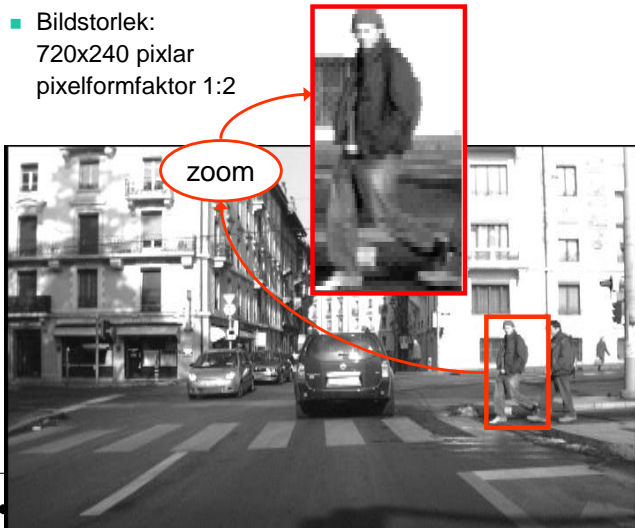
- Oftast är samplen kvantiserade i intervallet $[0,255]$. Dessa värden översätts via en färgtabell i datorn till
 - gråskalevärden, dvs 0->svart och 255->vitt eller
 - indexerade färger (pseudo-färg, färgtabell)
- HDR (high dynamic range) använder större intervall
- Ibland: samplen flyttatsvärden. Då de ska visas på skärm transformeras de till intervallet $[0,255]$ eller $[0,1]$ och vidare via färgtabell.
- Värden i intervallet $[0,1]$ kan gamma-korrigeras $f \rightarrow f^\gamma$



- Bildstorlek:
720x240 pixlar



- Bildstorlek:
720x240 pixlar
pixelformfaktor 1:2



För en digital bild gäller, forts

- En *äkta färgbild* har 3 st värden per pixel, RGB (R: röd, G:grön, B:blå)
 - De transformeras var för sig till intervallet $[0,255]$
 - Sedan ut på datorns röda, gröna respektive blåa kanal
 - Vilket möjliggör $256^3=16777216 \approx 16,8$ miljoner färger
- En fjärde kanal kan tillfogas för transparensen, en s.k. alfakanal. Används t ex inom datorgrafik.
- Man kan även använda andra färgmodeller, t ex
 - YCbCr (linjär; Y: luminans, Cb: blådifferens, Cr: röddifferens). Används t ex vid JPEG-kodning.
 - CIELAB (olinjär; L*: luminans, a*: magenta – grön balans, b*: gul – blått balans)
- Överföring sker som mest via DP, HDMI eller SDI

PET-bild av hjärna



Pseudo-färgbild

gråskalebild



Äkta färgbild




Foto: SOHNPIE

li.u LINKÖPINGS UNIVERSITET

Hur kan man uppleva olika färger av RGB?

- Datorskärm, tv: Ljuset lyser ut från skärmen, ger *additiv* färgblandning.
- Målarfärger: lampljus/solljus reflekteras från färgen, ger *subtraktiv* färgblandning.
- Ex) Datorpixel som ger vitt ljus:
- Ex) Datorpixel som ger gult ljus:
- Ex) Datorpixel som ger cyan ljus:
- Ex) Datorpixel som ger magenta ljus:



li.u LINKÖPINGS UNIVERSITET

Vanlig gråskalefärgtabell

Pixelvärde $f(x,y)$

Linjär transformation

gamma transformation

	R	G	B
0:	0	0	0
1:	1	1	1
2:	2	2	2
...
255:	255	255	255

Ut på skärmen

256 färger

I denna kursen jobbar vi mest med gråskalefärgtabellen.

li.u LINKÖPINGS UNIVERSITET

Pseudo-färgtabell

Pixelvärde $f(x,y)$

godtycklig transformation

	R	G	B
0:			
1:			
2:			
...	?	?	?
255:			

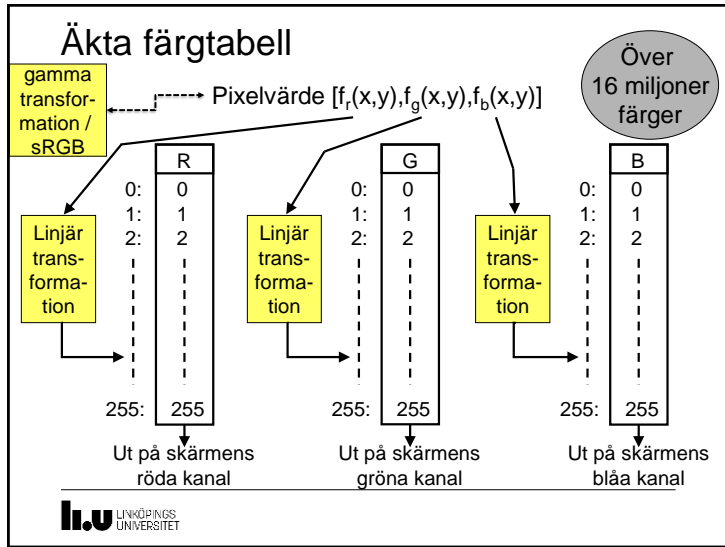
Ut på skärmen

256 färger


Ex 1) En PET-bild kan visa var det är aktivitet i hjärnan. Hög aktivitet kan visas röd och låg aktivitet kan visas blå.

Ex2) Användbart ex när vi vill visa negativa värden blå och positiva värden röda.

li.u LINKÖPINGS UNIVERSITET



Varför fouriertransformen?



- Följning i video (av t ex den gula fisken) kräver realtidsprestanda
- Bästa metoden (vi på CVL ☺) använder fouriertransformen

li.u LINKÖPINGS UNIVERSITET

2D kontinuerlig fouriertransform

$$\mathfrak{F}_2[f(x, y)] = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(xu+yv)} dx dy \quad (3.3)$$

2D fouriertransform

$$\mathfrak{F}_2^{-1}[F(u, v)] = f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(xu+yv)} du dv \quad (3.4)$$

2D invers fouriertransform

li.u LINKÖPINGS UNIVERSITET

2D fouriertransformen kan beräknas med 2 st 1D fouriertransformer

- Den kan beräknas först i ena lednen och sen i andra lednen:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(xu+yv)} dx dy$$

$$= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) e^{-j2\pi yv} dy \right] e^{-j2\pi xu} dx \quad (3.3)$$

$F_1(x, v)$ (fouriertransform i y - led)

li.u LINKÖPINGS UNIVERSITET

Fouriertransformen av en reell funktion är hermitisk:

$$F(u, v) = F^*(-u, -v)$$

Realdelen är jämn och imaginärdelen är udda.

$$\operatorname{Re}F(u, v) = \operatorname{Re}F(-u, -v)$$

$$\operatorname{Im}F(u, v) = -\operatorname{Im}F(-u, -v)$$

Amplitudspektrum är symmetriskt i origo.

$$F(u, v) = F^*(-u, -v) \Rightarrow |F(u, v)| = |F(-u, -v)|$$

se Fig. 3.2

Bevis: 2D fouriertransformen av en reell funktion är hermitisk

$$\begin{aligned} \underbrace{F^*(-u, -v)} &= \left[\iint \underbrace{f(x, y)}_{\text{real}} e^{-j2\pi(-ux - vy)} dx dy \right]^* = \\ &= \left\{ (e^{j\phi})^* = e^{-j\phi} \right\} = \\ &= \iint f(x, y) e^{-j2\pi(ux + vy)} dx dy = \underbrace{F(u, v)} \\ \therefore F^*(-u, -v) &= F(u, v) \end{aligned}$$

En bild med amplitudspektrum

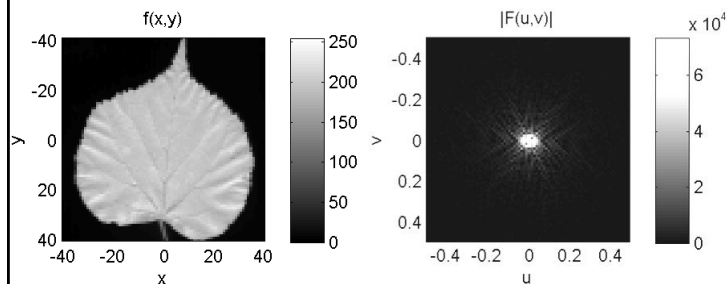
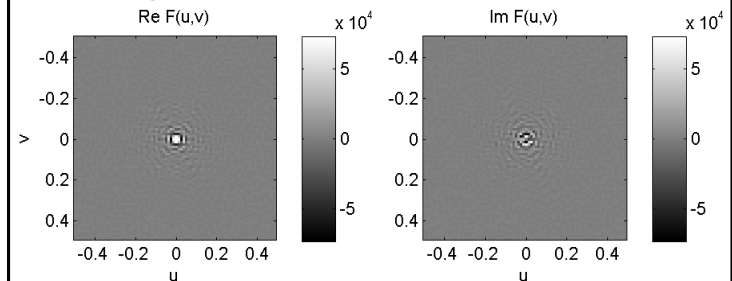


Fig. 3.2

Realdel och Imaginärdel av Fouriertransformen

- Realdelen är jämn
- Imaginärdelen är udda

Fig. 3.2



Teorem och samband, se formelsamlingen!

- Teoremen för 2D-fouriertransform, bl a skalnings-, faltnings-, translations- och derivata-teoremet gäller även i 2D.

- 2D-unikt teorem för generell linjär transform:

$$f(\mathbf{Ax}), \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{blir} \quad \frac{1}{|\det \mathbf{A}|} F((\mathbf{A}^{-1})^T \mathbf{u}), \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$$

- 2D rotation som specialfall:

$$f(\mathbf{Rx}), \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{blir} \quad F(\mathbf{Ru}), \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$$



LINKÖPINGS
UNIVERSITET

Teorem och samband

- Separabla funktioner ger separabel fouriertransform:

$$f(x, y) = g(x) \cdot h(y) \Rightarrow F(u, v) = G(u) \cdot H(v)$$

Separabel funktion:	$f(x, y) = g(x) \cdot h(y)$	$F(u, v) = G(u) \cdot H(v)$
Dirac-puls:	$\delta(x, y) = \delta(x) \cdot \delta(y)$	1
Spikmatta:	$\frac{1}{\Delta^2} \text{III}\left(\frac{x}{\Delta}, \frac{y}{\Delta}\right) = \frac{1}{\Delta} \text{III}\left(\frac{x}{\Delta}\right) \cdot \frac{1}{\Delta} \text{III}\left(\frac{y}{\Delta}\right)$	$\text{III}(\Delta u, \Delta v) = \text{III}(\Delta u) \cdot \text{III}(\Delta v)$
Box:	$\Pi(x, y) = \Pi(x) \cdot \Pi(y)$	$\text{sinc}(u) \cdot \text{sinc}(v)$
Böjd pyramid:	$\Lambda(x, y) = \Lambda(x) \cdot \Lambda(y)$	$\text{sinc}^2(u) \cdot \text{sinc}^2(v)$
Gauss:	$e^{-\pi(x^2+y^2)} = e^{-\pi x^2} \cdot e^{-\pi y^2}$	$e^{-\pi(u^2+v^2)} = e^{-\pi u^2} \cdot e^{-\pi v^2}$



LINKÖPINGS
UNIVERSITET

2D DFT

- Pythonkommando: `F = np.fft.fft2(f)`

$$F_D(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_D(n, m) e^{-j2\pi(nk/N + ml/M)}$$

$$f_D(n, m) = \frac{1}{MN} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F_D(k, l) e^{j2\pi(nk/N + ml/M)}$$

- Obs: i Python $F_D(k, l) \rightarrow F[l, k]$ och $f_D(n, m) \rightarrow f[m, n]$
- Sparas i row-major format
- Notera: vanligtvis väljs symmetriska varianten med summor från $-N/2$ till $N/2-1$ och $-M/2$ till $M/2-1$,
`F = np.fft.fftshift(np.fft.fft2(np.fft.ifftshift(f)))`



LINKÖPINGS
UNIVERSITET

Teorem och samband

- Teoremen för 2D DFT motsvarar kontinuerliga fallet. Notera att multiplikation i DFT-domänen motsvarar **cirkulär faltning** i spatialdomänen.

Teorem:	Spatialdomän	Fourierdomän
Notation:	$f(x, y)$	$F(u, v)$
Skalningsteoremet:	$f(ax, by)$	$(1/ ab) \cdot F(u/a, v/b)$
Cirkulär faltning:	$(f *_N g)(x, y)$	$F(u, v) \cdot G(u, v)$
Cirkulär korrelation:	$(f \square_N g)(x, y)$	$F^*(u, v) \cdot G(u, v)$
Translationsteoremet:	$f(x-a, y-b)$	$e^{-j\frac{2\pi}{N}(au+bv)} F(u, v)$
Parsevals formel:	$\sum_{x,y=0}^{N-1} f(x, y)g(x, y) =$	$= \frac{1}{N^2} \sum_{u,v=0}^{N-1} F(u, v)G^*(u, v)$



LINKÖPINGS
UNIVERSITET

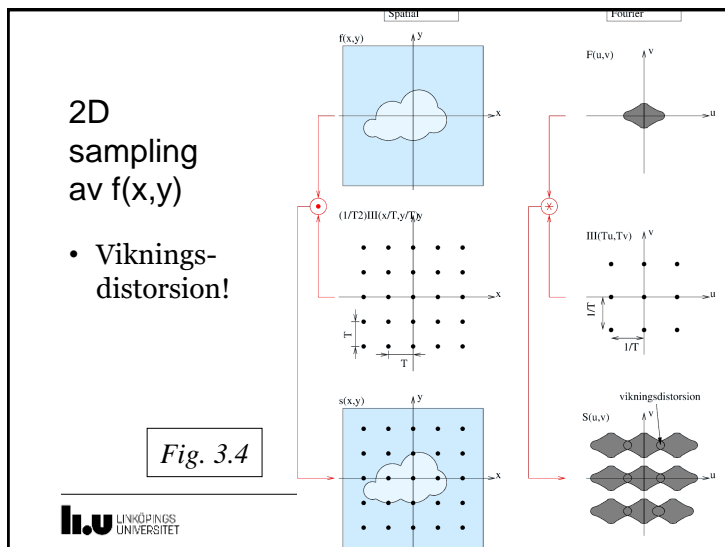
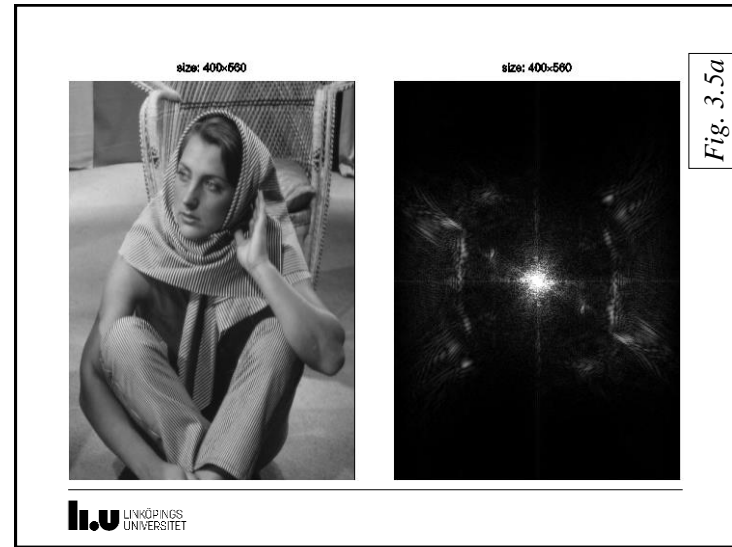
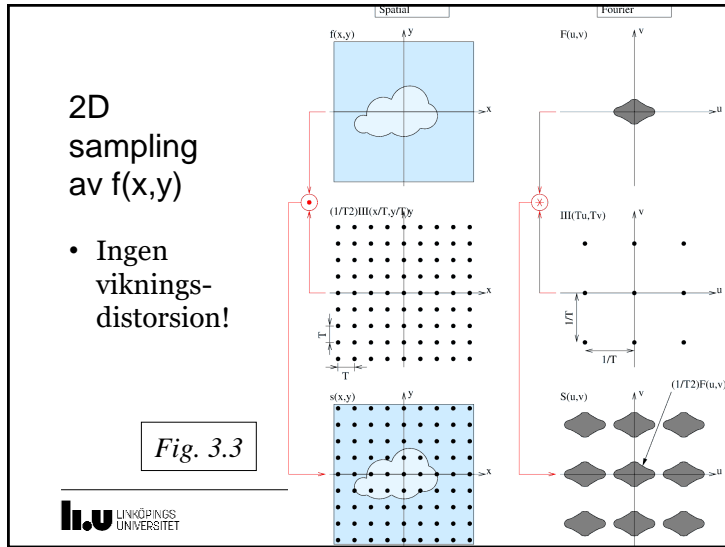


Illustration av
vignings-
distortion.
Original
sammel-
avstand.
Original
pixelstorlek.



Illustration av
vignings-
distortion.
Original
sammel-
avstand / 2
Original
pixelstorlek · 2



Illustration av
vignings-
distortion.
Original
sammel-
avstand / 4
Original
pixelstorlek · 4



Illustration av
vignings-
distortion.
Original
sammel-
avstand / 8
Original
pixelstorlek · 8



Samband mellan samplad kontinuerlig fouriertransform och DFT

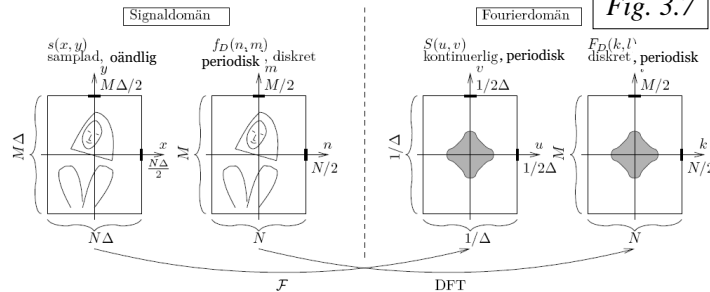


Fig. 3.7

- Relationen mellan kontinuerlig frekvens u, v och diskret frekvens k, l står i (3.13). N, M är antalet sampelpunkter och Δ är sampelvståndet.

$$\begin{cases} u = k/(N\Delta) \\ v = l/(M\Delta) \end{cases} \quad (3.13)$$

Varning! Lite svårare beräkningar här!

Fouriertransform av diskret 2D signal

- Fouriertransform av samplad signal ($\Delta=1$)

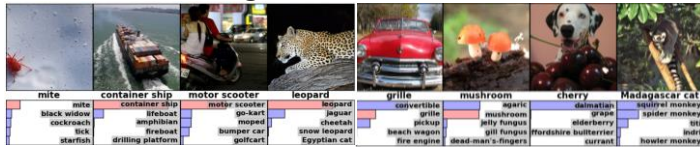
$$\mathfrak{T}_2[f_D(n, m)] = F_D(u, v)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta(x-n, y-m) f_D(n, m) e^{-j2\pi(xu+yv)} dx dy$$

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f_D(n, m) e^{-j2\pi(nu+mv)}$$

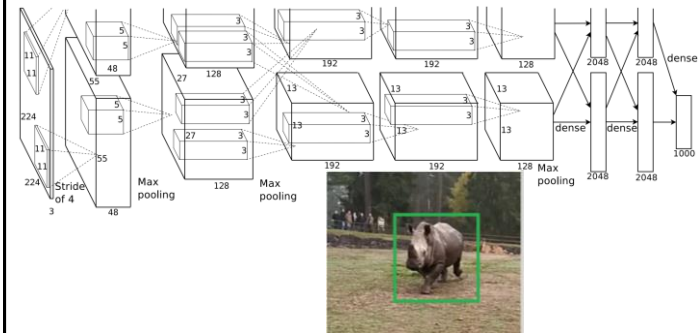
$$\mathfrak{T}_2^{-1}[F_D(u, v)] = f_D(n, m) = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} F_D(u, v) e^{j2\pi(nu+mv)} du dv$$

Varför faltning?



Method	Top-5 Error	Method Description
SuperVision (Toronto)	0.16422	CNN
ISI (Tokyo)	0.26172	Hand-crafted features: SIFT, HOG and LBP
OXFORD	0.26979	DPM + Hand-crafted features
XRCE/INRIA	0.27058	Hand-crafted features

The Convolutional Neural Network (CNN): AlexNet [Krizhevsky et al. 2012]



2D faltning

$$g(x, y) = (h * f)(x, y) = \int \int_{-\infty-\infty}^{\infty} h(x-\alpha, y-\beta) \cdot f(\alpha, \beta) d\alpha d\beta$$

Kontinuerlig (3.15)

$$g(x, y) = (h * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} h(x-\alpha, y-\beta) \cdot f(\alpha, \beta)$$

Linjär diskret (3.17)

$$g(x, y) = (h * f)(x, y) = \sum_{\alpha=0}^{N-1} \sum_{\beta=0}^{M-1} h(x-\alpha, y-\beta) \cdot f(\alpha, \beta)$$

Cirkulär diskret (3.18)

Ex) Linjär och Cirkulär faltning

Inbild

Medelvärdesbildande faltningskärna Storlek 15x15

Utbild efter linjär faltning

Utbild efter cirkulär faltning

Tänk att det är nollor utanför Inbilden då kärnan glider fram.

Tänk att inbilden sitter ihop över-/underkant och vänster-/högerkant då kärnan glider fram.

li.u LINKÖPINGS UNIVERSITET

2D linjär diskret faltning

$$g(x, y) = (h * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} h(x-\alpha, y-\beta) \cdot f(\alpha, \beta)$$

- Spegla h i x- och y-axeln = rotera 180° skillnad mot korrelation
- Glid med den speglade h över f
- Multiplicera och summera överlappande värden
- Python: signal.convolve2d()

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

$h(x, y)$

*

1	2	1
---	---	---

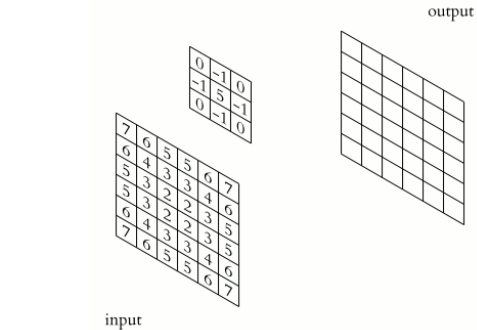
$f(x, y)$

=

1	2	0	-2	-1
2	4	0	-4	-2
1	2	0	-2	-1

$g(x, y)$

Animation 2D faltning/korrelation



Bildstorlek vid 2D linjär diskret faltning

- **'valid'**: Värden utanför inbilden anses odefinierade => Utbilden blir mindre än inbilden.
- **'full'**: Värden utanför inbilden anses vara 0 => Utbilden blir större än inbilden. Eller lika stor om de extra värdena slängs ('same')

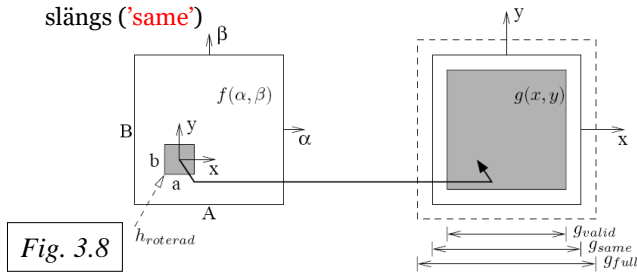
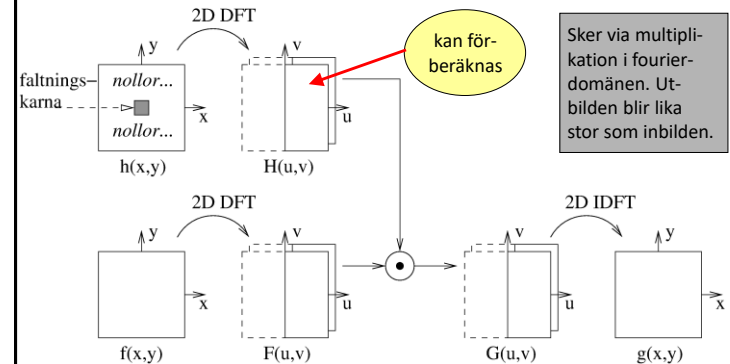


Fig. 3.8

LINKÖPINGS UNIVERSITET

Mer om cirkulär faltning: Filtrering via multiplikation i DFT-domänen ger cirkulär faltning



LINKÖPINGS UNIVERSITET

Fig. 3.9

1D cirkulär faltning

$$g_D *_{N} h_D(m) = \sum_{n=0}^{N-1} g_D(n) \cdot h_D((m-n)_N) \quad (2.59)$$

- $*_{N}$ noterar cirkulär faltning
- $(\)_N$ betecknar modulo N operation
- $h_D(n) = h_D(N+n)$ är periodiskt upprepad / cirkulär

Det gäller:

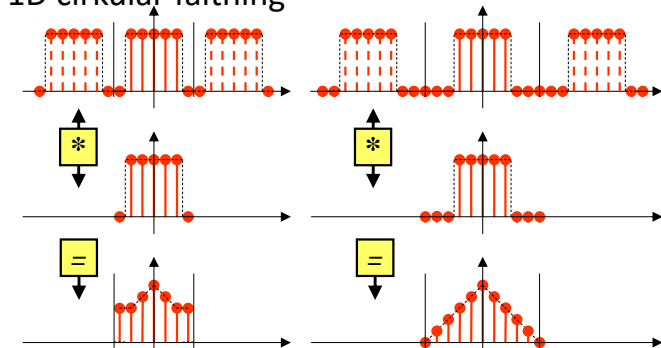
$$DFT(g_D *_{N} h_D(m)) = G_D(k) \cdot H_D(k)$$

Bevis finns i kompendiet

Python: `signal.fftconvolve()`

LINKÖPINGS UNIVERSITET

1D cirkulär faltning



Alltså: För att cirkulär faltning ska ge samma resultat som vanlig linjär faltning kan zero-padding behövas.

LINKÖPINGS UNIVERSITET