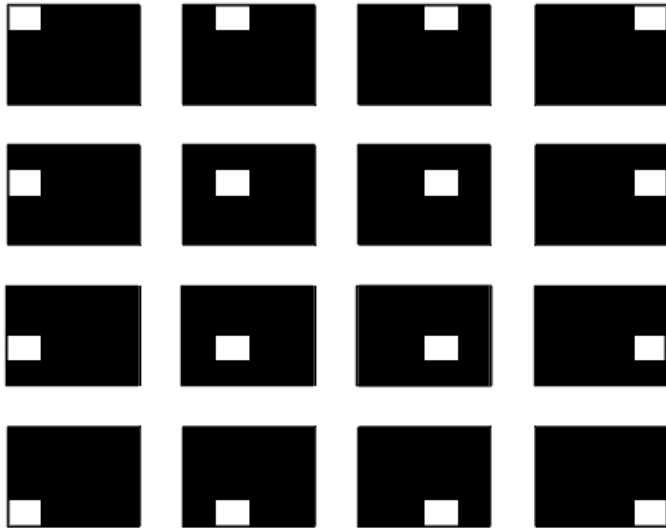

INF2310 – 29. mars 2017

Diskret Fouriertransform – del II

- Kjapp repetisjon
- Konvolusjonsteoremet
- Filtre og filtrering i frekvensdomenet
- Bruk av vinduer

Repetisjon

Basis-bilder



Sort er 0, hvit er 1.

← Ortogonal basis for alle
4x4 gråtonebilder.

Eksempel:

1	3	2	1
5	4	5	3
4	1	1	2
2	3	2	6

$$= 1 * \begin{array}{|c|c|c|c|} \hline \text{white} & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} + 3 * \begin{array}{|c|c|c|c|} \hline & \text{white} & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} + \dots + 6 * \begin{array}{|c|c|c|c|} \hline & & & \text{white} \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

En alternativ basis

- Bildene

$$\cos\left(\frac{2\pi(ux + vy)}{N}\right)$$

$$\sin\left(\frac{-2\pi(ux + vy)}{N}\right)$$

- med frekvensene

$$u = 0, 1, \dots, N-1$$

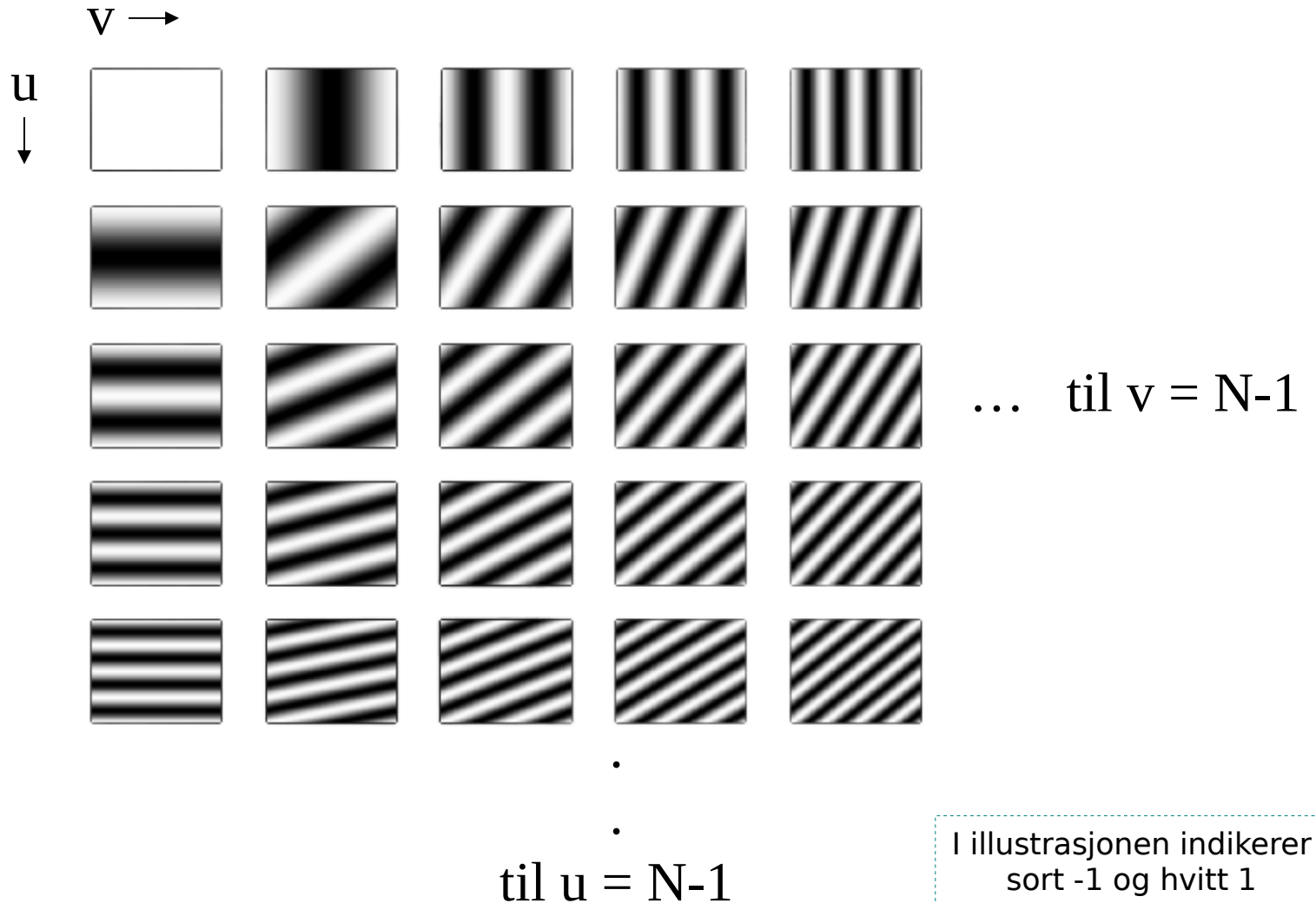
$$v = 0, 1, \dots, N-1$$

- Alle digitale gråtonebilder av størrelse $N \times N$ kan representeres ved en vektet summasjon av disse $N \times N$ sinus- og cosinus-bildene (basisbilder/basisvektorer)

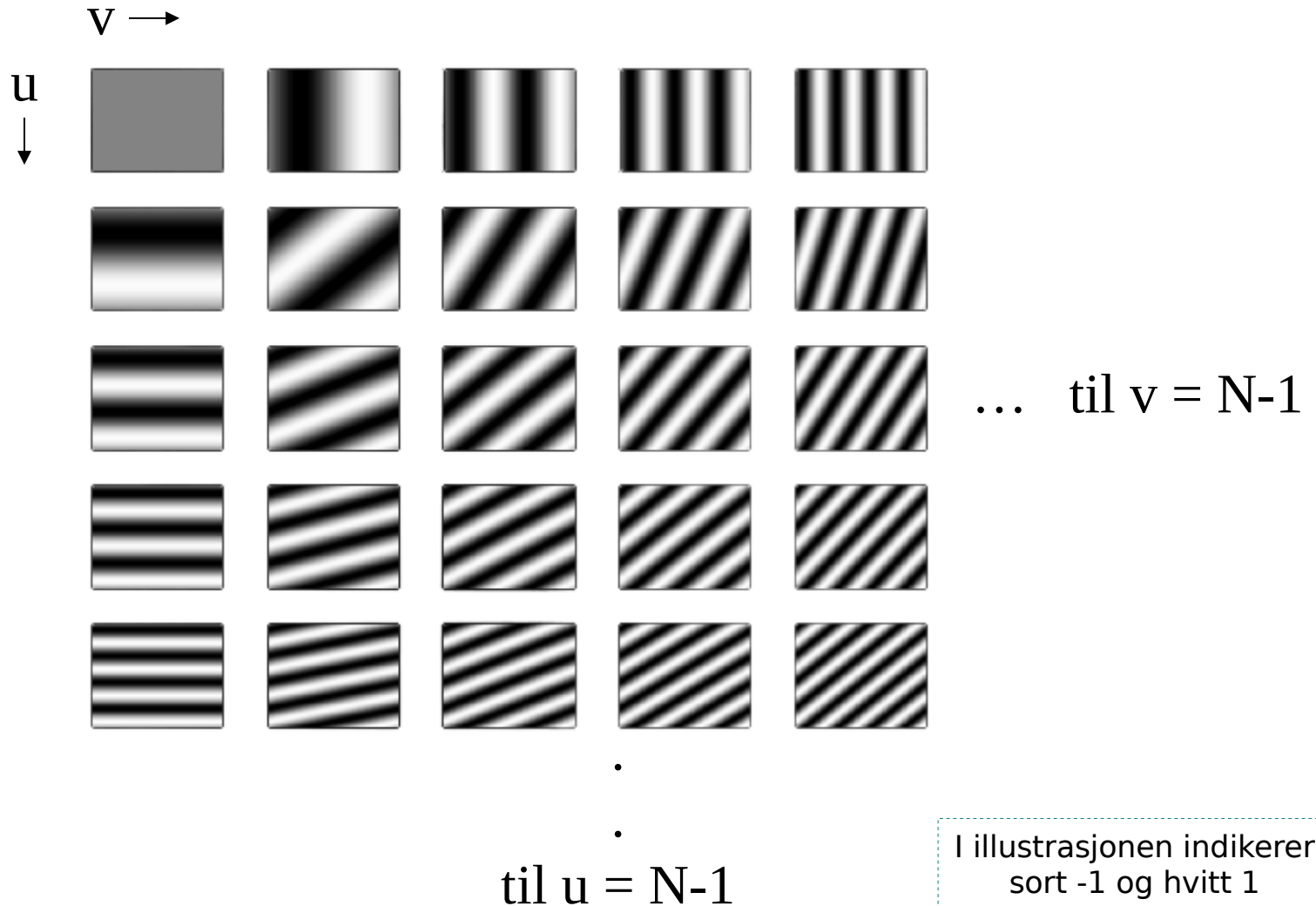
Denne basisen er også ortogonal, sett bort i fra duplikat-komponentene grunnet symmetriene og antisymmetriene til cos og sin (noe vi kommer til om litt)

Ved ikke-kvadratiske bilder:
 $\cos(2\pi(ux/M + vy/N))$
 $\sin(-2\pi(ux/M + vy/N))$

Basisbilder - cosinus



Basisbilder - sinus



2D diskret Fouriertransform (DFT)

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-2j\pi(ux/N + vy/M)}$$

Husk at $e^{j\theta} = \cos(\theta) + j \sin(\theta)$, slik at vi ender opp sin/cos-basisen vi er vant med:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \left[\cos(2\pi(ux/N + vy/M)) + j \sin(-2\pi(ux/N + vy/M)) \right]$$

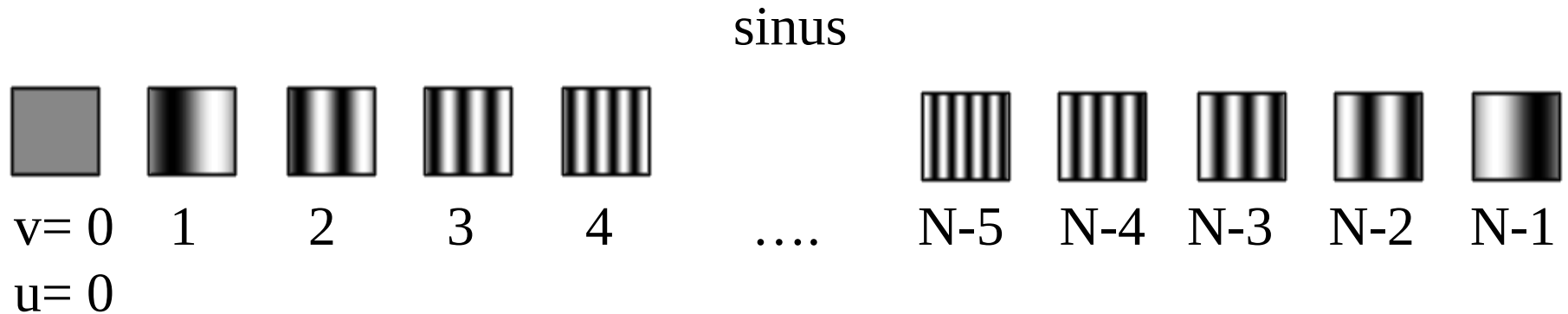
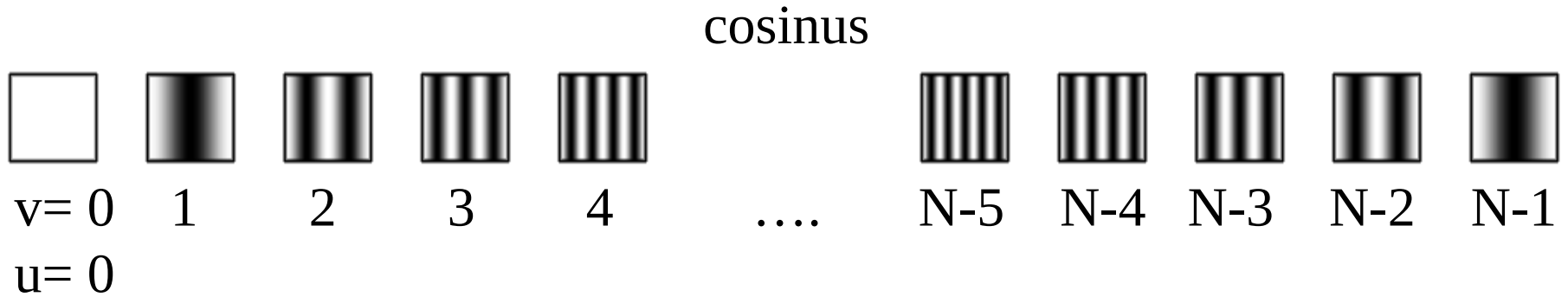
Den inverse transformen:

$$f(x, y) = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{2j\pi(ux/N + vy/M)}$$

Litt repetisjon om DFT

- Fouriertransformen $F(u,v)$ er periodisk:
 $F(u,v) = F(u+kN, v+kN)$, k heltall
- Bildet $f(x,y)$ implisitt periodisk: $f(x,y) = f(x+kN, y+kN)$
- Amplitudespekteret er gitt ved $|F(u,v)|$
- Konjugert symmetri: Hvis $f(x,y)$ er reell, er $F(u,v) = F^*(-u,-v)$
og altså $|F(u,v)| = |F(-u,-v)|$
- Ofte forskyver spekteret med $N/2$ for å få origo ($u=v=0$) midt i bildet
- 2D DFT er separabelt i to 1D DFT
- Shift-teoremet: $f(x-x_0, y-y_0) \Leftrightarrow F(u,v) e^{-j2\pi(ux_0+vy_0)/N}$

Symmetri i basisbildene



(antisymmetri i sinus-bildene)

Konvolusjonsteoremet

$$f(x, y) \star h(x, y) \iff F(u, v) \cdot H(u, v)$$

Konvolusjon i billedomenet \Leftrightarrow Punktvis multiplikasjon i frekvensdomenet

Det motsatte gjelder også:

$$f(x, y) \cdot h(x, y) \iff F(u, v) \star H(u, v)$$

Punktvis multiplikasjon i billedomenet \Leftrightarrow Konvolusjon i frekvensdomenet

Egentlig snakk om en «sirkelkonvolusjon»

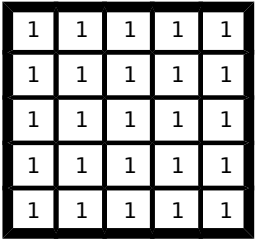
→ Diskrete tilfellet:
Elementvis produkt av de
komplekse matrisene F og H

Anvendelser

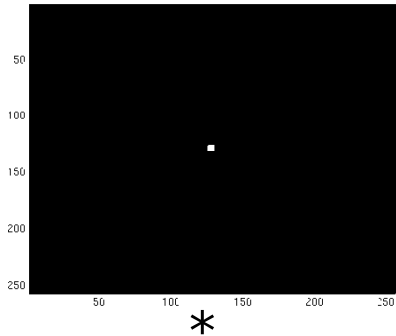
- Analyse av konvolusjonsfiltre
 - Fourier-transformen til et filter h gir oss innblikk i *frekvensresponsen* til filteret
- Filterdesign
 - Kan designe filter i både frekvensdomenet og billedomet
 - Begge kan implementeres som konvolusjon i billedomet, eller som multiplikasjon i frekvensdomenet
 - (Husk: F og H må ha samme størrelse: Nullutvide)
- Implementasjon
 - Store filtre kan implementeres raskere i frekvensdomenet

Eksempel: Middelvefilteret

1/25

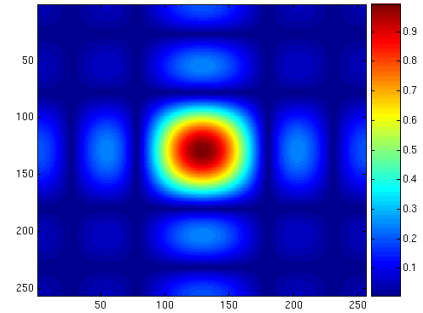


null-
utvide

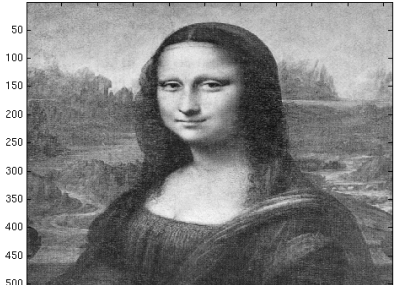


*

(I)DFT

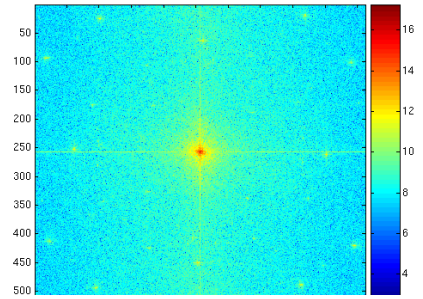


[Det nullutvidede bildet har egentlig størrelse 512x512. Aksene viser feil.]

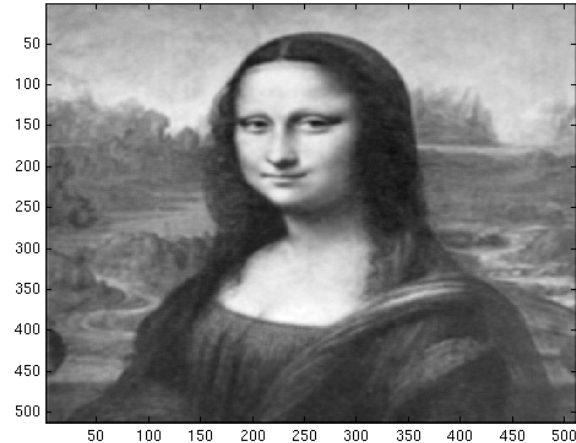


=

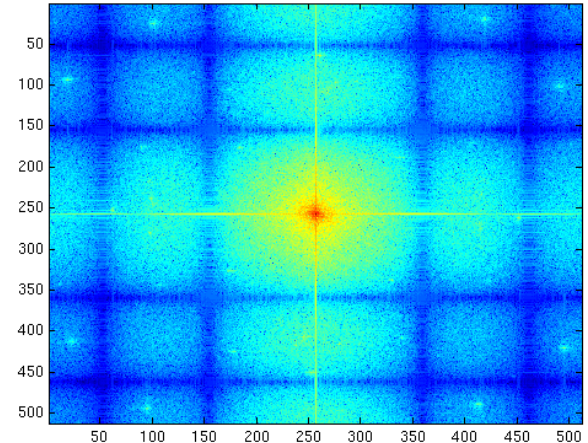
(I)DFT



=



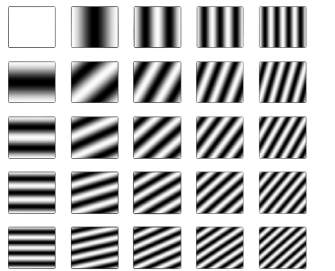
(I)DFT



Konvolusjonsteoremet: Tommelfingerforklaring



Sirkelkonvolvere

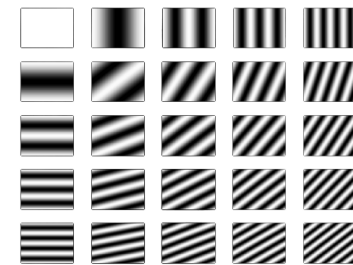


etc.

Sirkelkonvolvere
alle kombinasjoner
og så summere

MEN:

Ved ulik (u,v) får vi 0.
Ved like (u,v) endres kun
amplitude og fase.



etc.

Å (sirkel)konvolvere et bilde med en av basisbildene gir som resultat det samme basisbildet dog med mulig endret amplitude og fase

Konvolusjonsteoremet mer formelt (1D)

Sirkelkonvolusjon

$$\begin{aligned} \text{DFT}_k(x \circledast y) &\triangleq \sum_{n=0}^{N-1} (x \circledast y)_n e^{-j2\pi nk/N} \\ &\triangleq \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(m)y(n-m) e^{-j2\pi nk/N} \\ &= \sum_{m=0}^{N-1} x(m) \underbrace{\sum_{n=0}^{N-1} y(n-m) e^{-j2\pi nk/N}}_{e^{-j2\pi mk/N} Y(k)} \\ &= \left(\sum_{m=0}^{N-1} x(m) e^{-j2\pi mk/N} \right) Y(k) \quad (\text{by the Shift Theorem}) \\ &\triangleq X(k)Y(k) \end{aligned}$$

(Kopi fra dsprelated.com)

Design i romlige domenet og filtrering i frekvensdomenet

Har en filterkjerne og vil implementere filtreringen i frekvensdomenet:

1. Beregn DFT (fft) av bildet
 2. Beregn DFT (fft) av filterkjernen (med evt nullutvidelse)
 3. Multipliser de to transformerte matrisene elementvis
 4. Transformer resultatet tilbake til billedometet vha. invers DFT (IDFT, ifft)
- **Husk at filteret og bildet må ha samme størrelse (nullutvide filterkjernen)**
- Husk at vi snakker sirkelkonvolusjon (må nullutvide mer [også bildet] om vi ønsker alternative randhåndtering)

Filterdesign i Fourier-domenet

Generelt

- Vi ønsker reelle konvolusjonskjerne
=> (konjugert) symmetrisk i Fourierdomenet
- Ofte er alle **verdiene til filteret mellom 0 og 1**;
0 fjerner og 1 bevarer den aktuelle frekvensen
- Hvis nullfrekvensen ($u=0, v=0$), «DC», i filteret er 1 så bevarer bildets middelværdi
 - Vi viste forrige uke at DC er summen av gråtoneverdiene
 - Hvis DC i filteret er 1 så vil DC i ut-bildet bli lik DC i inn-bildet, altså vil summen av gråtoneverdiene bevarer

Filterdesign i frekvensdomenet

Lavpassfiltre

- Slipper bare gjennom lave frekvenser (mindre enn en grense D_0 som kalles filterets **cut-off-frekvens**)
 - D_0 oppgis ofte som et tall mellom 0 og 1; da menes en cut-off = $D_0 N/2$
- Enkelt (også kalt ideelt) lavpassfilter:

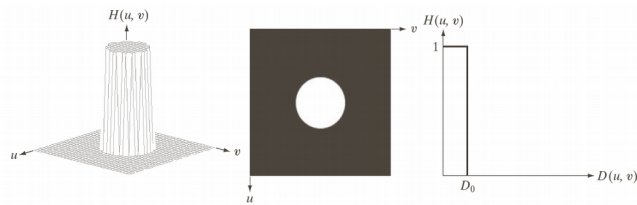


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

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

$$D(u, v) = \frac{\sqrt{(u - N/2)^2 + (v - N/2)^2}}{(N/2)}$$

- (Ordet "ideelt" kommer fra om $H(u,v)$ var enten 0 eller 1 for alle mulige frekvenser u og v , ikke kun $0, 1, \dots, N-1$. Dette er et urealiserbart filter, da filterkjernestørrelsen da vil gå mot uendelig)

MATLAB-eksempel: Ideelt lavpassfilter

```
f = double(imread('..'));
[M,N] = size(f);
H = zeros(M,N);
D0 = 0.2;

for i = 1:M
  for j = 1:N
    if sqrt( ((i-floor(M/2+1))/(M/2))^2 + ...
             ((j-floor(N/2+1))/(N/2))^2 ) <= D0
      H(i,j) = 1;
    end
  end
end

F = fftshift( fft2(f) );
g = real( ifft2( ifftshift( F.*H ) ) );
imshow(g, []);
```

i og j er array-indeksene til H og er relatert til frekvensene ved et skift:

$$u = i - \lfloor M/2 + 1 \rfloor$$

$$v = j - \lfloor N/2 + 1 \rfloor$$

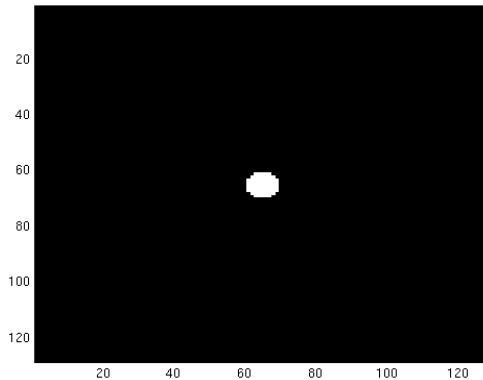
Hvorfor $\text{floor}(M/2+1)$?

Hvis M er odde: DC skal være pikselen midt i filteret. Da vil filterposisjonene representere frekvensintervallet $[-\text{floor}(M/2), \text{floor}(M/2)]$. Array-indeksene vil derfor angi frekvensene hvis vi skifter med $\text{floor}(M/2)$ (null-indeksert array) eller $\text{floor}(M/2+1) = \text{ceil}(M/2)$ (en-indeksert array).

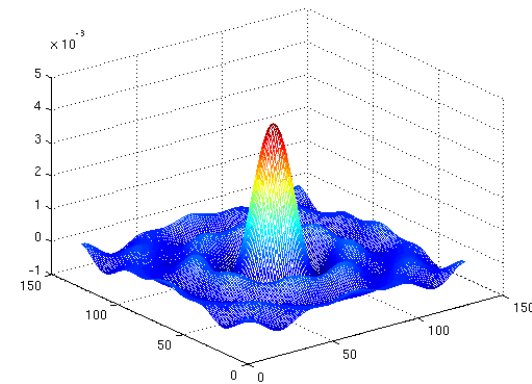
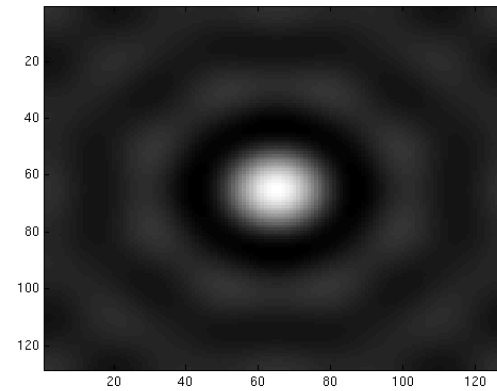
Hvis M er like: Senterpunktet i filteret er nå midt mellom piksler, men DC skal ligge i en piksel. Generelt kan vi velge om DC skal være den $M/2$ -te eller $(M/2+1)$ -te posisjonen i filteret. Det er sistnevnte som er vanlig og som brukes i FFTSHIFT og IFFTSHIFT. Med dette valget vil filterposisjonene representere frekvensintervallet $[-M/2, M/2-1]$. For én-indeksing skal vi da skifte med $M/2+1 = \text{floor}(M/2+1)$.

Analog forklaring for $\text{floor}(N/2+1)$.

Romlig representasjon av “ideelt” lavpassfilter



IDFT
→



(trunkert sinc-funksjon)

- Vi får en «ringing»-effekt i bildet
 - Og husk tommelfingerregel om utstrekning i fourier- og bilde-
domenet

Eksempler - ideell lavpass



Original



$D_0=0.2$



$D_0=0.3$

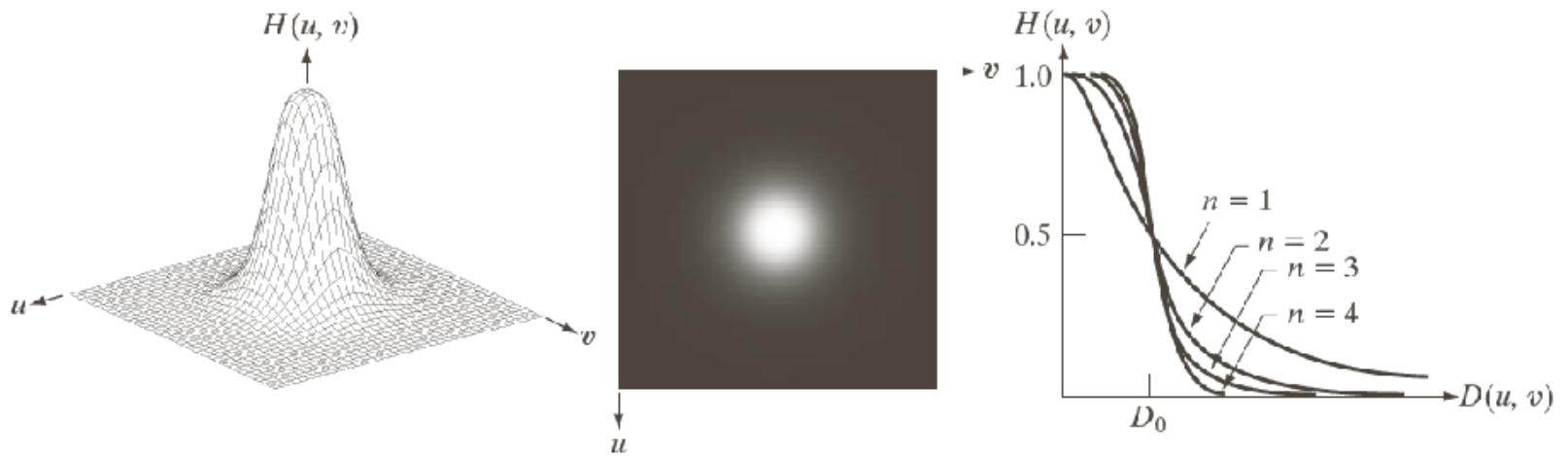
Se på bildene i god nok oppløsning (du skal se stripe/ringing-effekter i de to til høyre).

Butterworth lavpassfilter

- «Glattere» funksjoner brukes til å redusere ringing-effekten
- F.eks. Butterworth lavpassfilter av orden n :

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

- Her vil D_0 beskrive punktet der $H(u, v)$ har falt til halvparten av sin maksimumsverdi
 - Lav filterorden (n liten): $H(u, v)$ faller langsomt: Lite ringing
 - Høy filterorden (n stor): $H(u, v)$ faller raskt: Mer ringing
- Andre funksjoner kan også brukes, f.eks. Gaussisk, Bartlett, Blackman, Hamming, Hanning



a b c

FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Eksempler Butterworth-lavpass

$D_0=0.2$



n=11



n=41



n=61

Gaussisk lavpassfilter

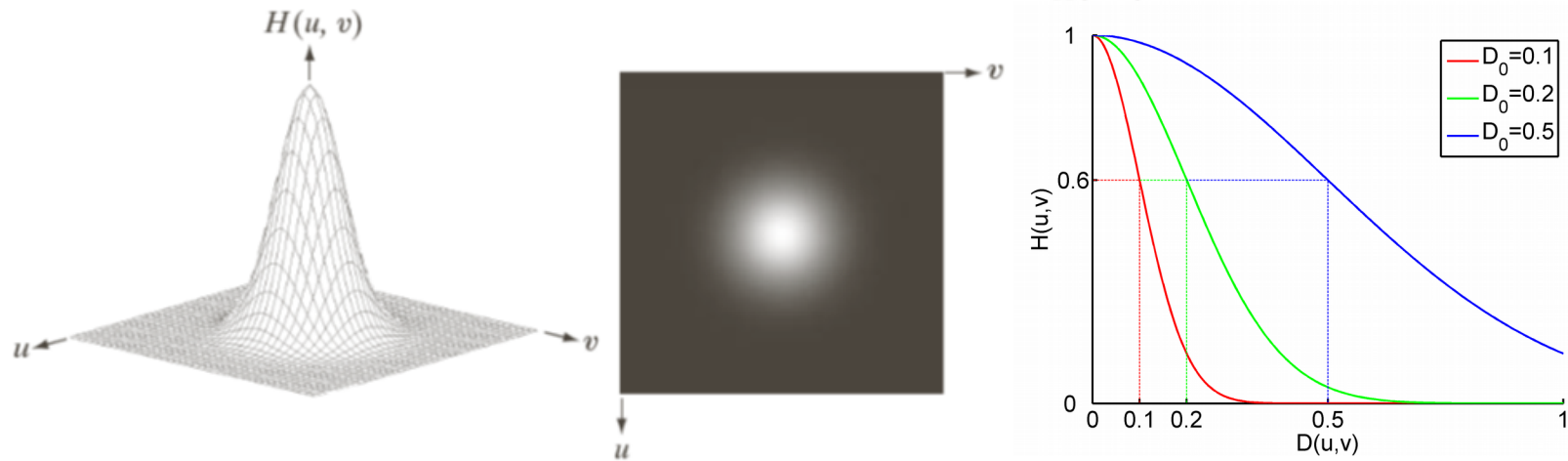
- Gaussisk lavpassfilter med spredning D_0 er definert som:

$$H(u, v) = e^{\frac{-D^2(u, v)}{2D_0^2}}$$

altså en 2D normalfordeling (uten konstantfaktoren) med DC som forventning og D_0 som standardavvik (i alle retninger, ingen kovarians).

- $H(0,0)$ er 1 og H er strengt avtagende i alle retninger ut fra DC.
- Standardavviket angir avstanden fra DC til punktet der H er $\approx 0,6$.
- 2D IDFT-en av et Gaussisk lavpassfilter er også Gaussisk.
 - Får ingen ringing i billedomenet!

Gaussisk lavpassfilter



a b c

FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0 .

Husk tommelfingerregelen:
Smal/bred struktur i bildet \Leftrightarrow Bred/smalt struktur i Fourier-
spekteret

Høypassfiltrering

- Et høypassfilter kan defineres ut fra et lavpassfilter:

$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$

- Ideelt høypassfilter:

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

- Butterworth høypassfilter:

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

- Gaussisk høypassfilter:

$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2D_0^2}}$$

Båndpass- og båndstoppfiltere

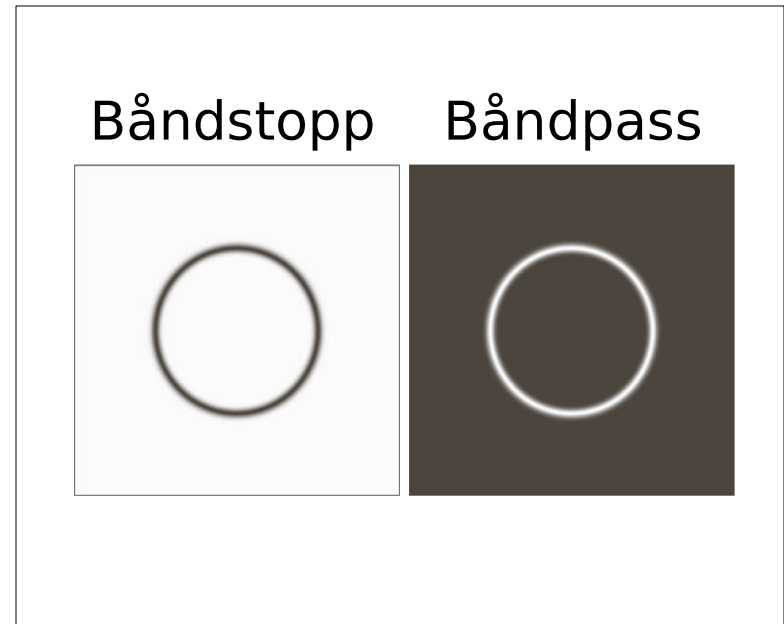
- Båndpassfilter: Slipper gjennom kun energien i et bestemt frekvensbånd $\langle D_{\text{low}}, D_{\text{high}} \rangle$ (eller $\langle D_0 - \Omega, D_0 + \Omega \rangle$)
- Båndstoppfilter: Fjerner energi i et bestemt frekvensbånd $\langle D_{\text{low}}, D_{\text{high}} \rangle$

- Butterworth båndstoppfilter:

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

- Butterworth båndpassfilter:

$$H_p(u, v) = 1 - H_s(u, v)$$



Notch-filtre

- Slipper igjennom (notch-passfiltre) eller stopper (notch-stoppfiltre) energien i mindre predefinerte området i Fourier-spekteret.
- Også disse kan bruke de samme overgangene:
 - Ideelt, Butterworth, Gaussisk (eller én av mange andre typer).
- + Kan være svært nyttige.
- - Ofte trengs interaktivitet for å definere de aktuelle områdene.

Eksempel: Notch-stopppfilter

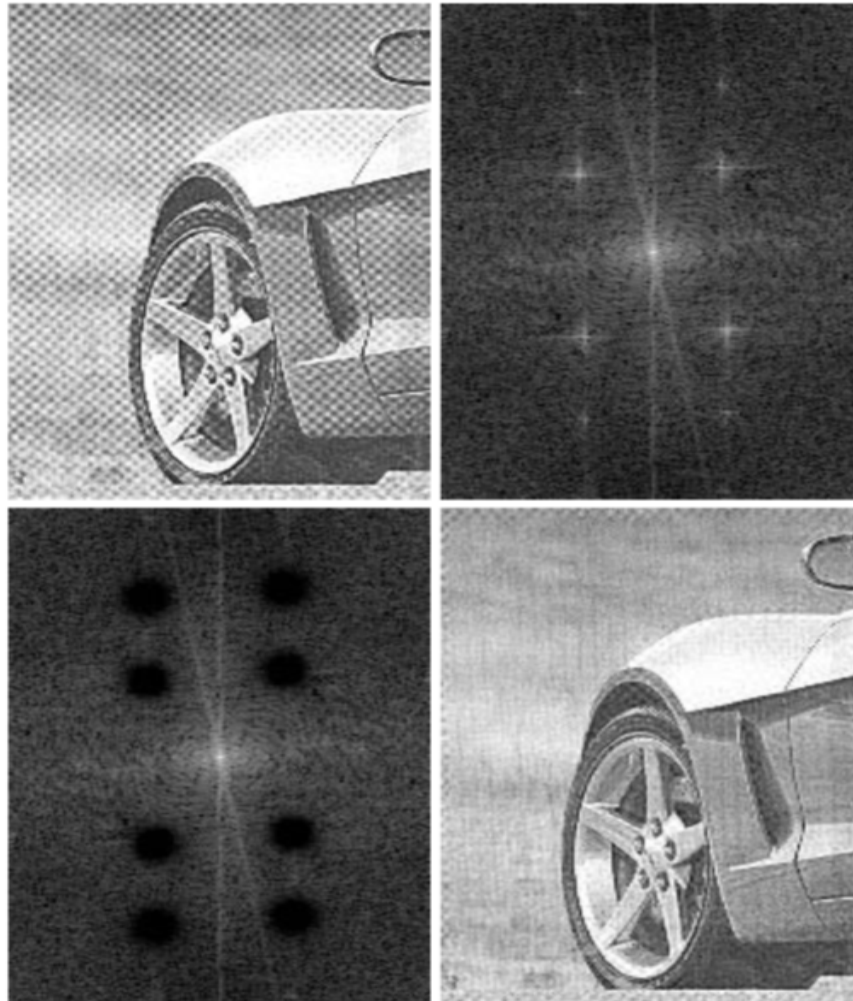
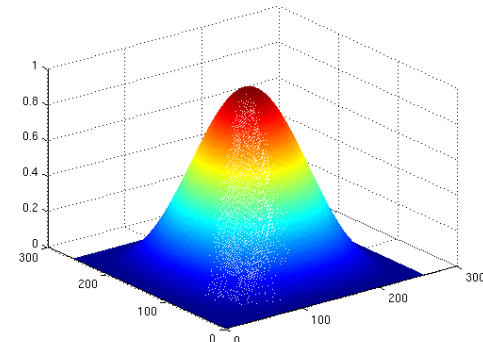
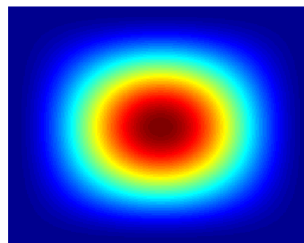
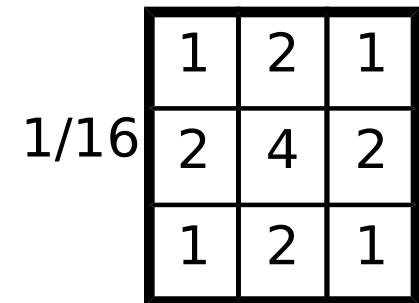
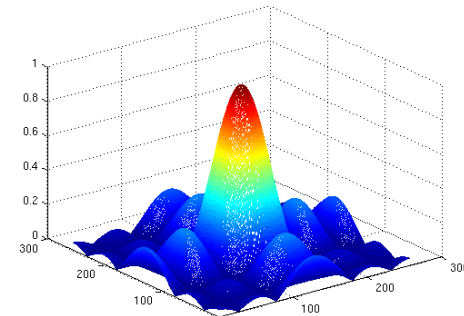
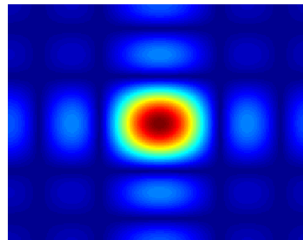
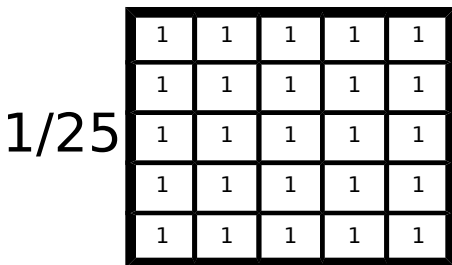
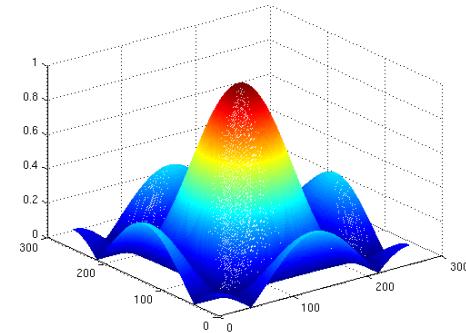
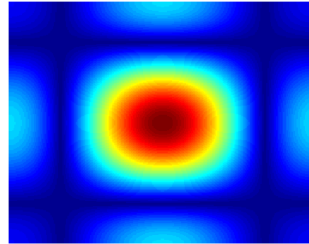
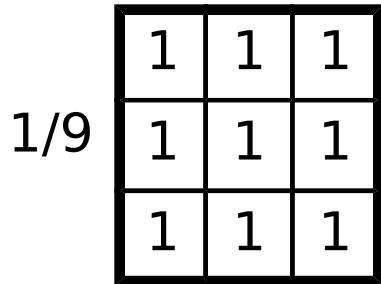


Fig. 4.64 i DIP

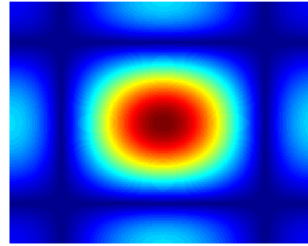
Analyse av filtre

Frekvensresponsen til noen vanlige filtre



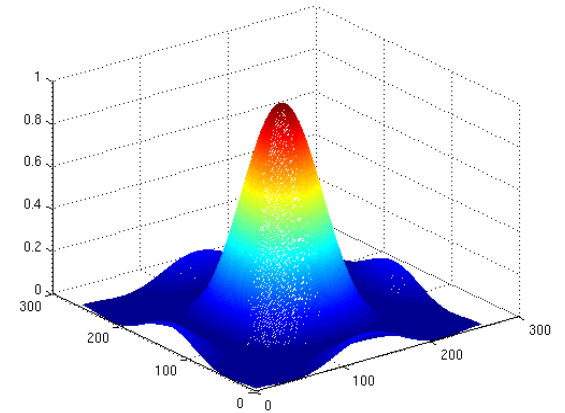
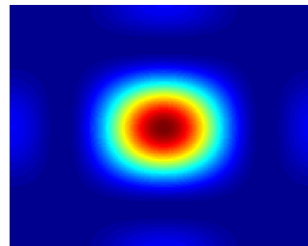
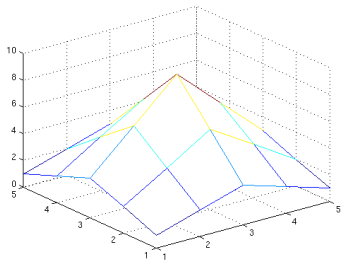
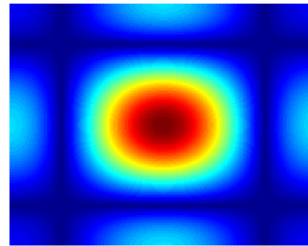
1	1	1
1	1	1
1	1	1

*



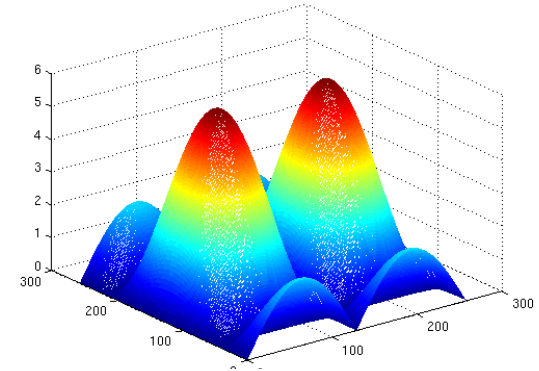
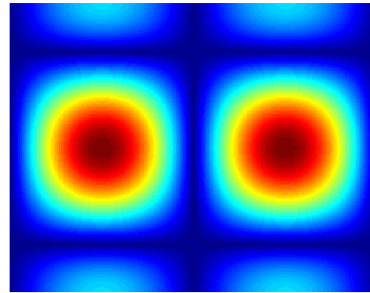
X

1	1	1
1	1	1
1	1	1

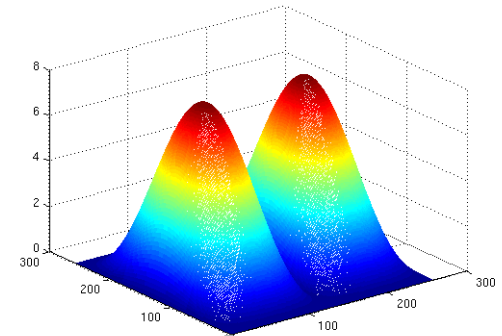
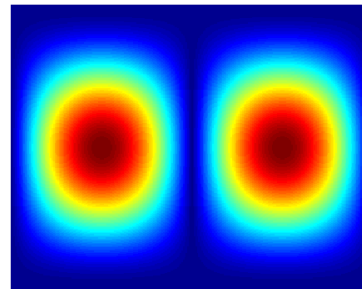


Høypassfiltre

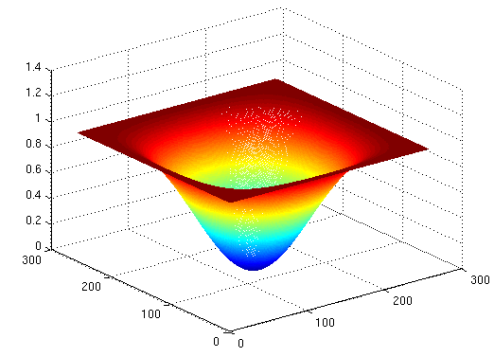
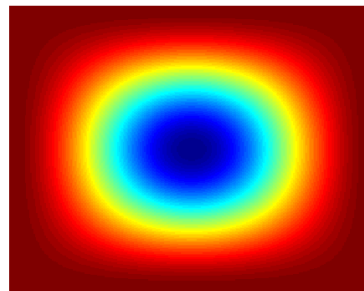
1	0	-1
1	0	-1
1	0	-1



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



-1	-2	-1
-2	12	-2
-1	-2	-1



Prewitt-filteret

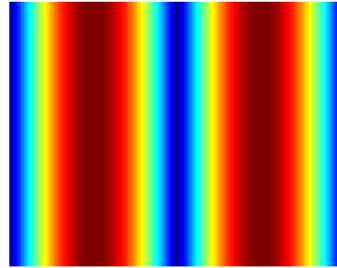
1	0	-1
---	---	----

*

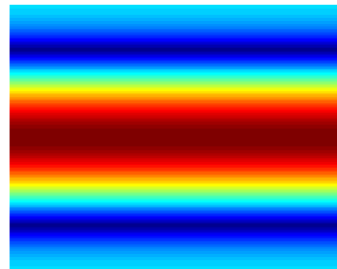
1
1
1

=

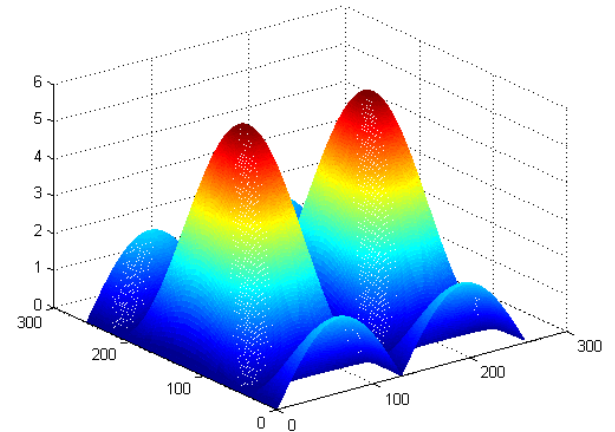
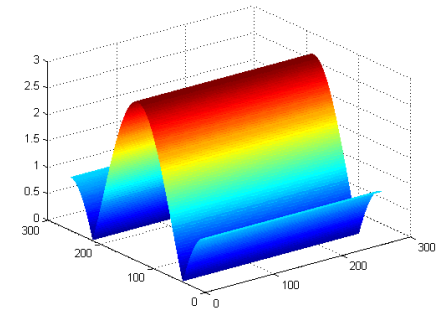
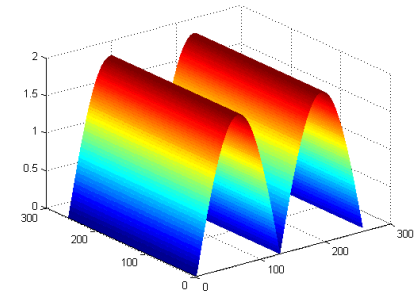
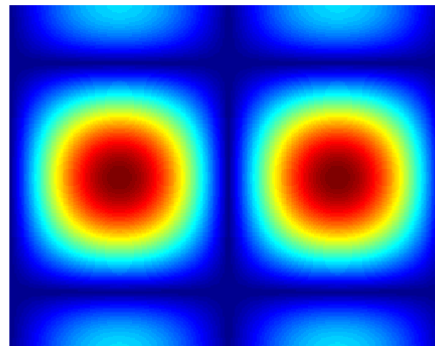
1	0	-1
1	0	-1
1	0	-1



X



=



Når er filtrering raskest i frekvensdomenet?

- Anta bildet har størrelse $N \times N$, filterkjernen $n \times n$
- Filtrering i billedometet krever $N^2 n^2$ multiplikasjoner og tilsvarende addisjoner
- Filtrering i frekvensdomenet:
 - FFT av bildet og filterkjernen: $2 * O(N^2 \log_2 N)$
 - Multiplikasjon i frekvensdomenet: N^2 multiplikasjoner
 - Inverstransform av resultatet : $O(N^2 \log_2 N)$
- Filtrering i frekvensdomenet raskere når filteret er stort ($n^2 \gg \log_2 N$)

«Korrelasjonsteoremet»

$$f(x, y) \circ h(x, y) \Leftrightarrow F(u, v)^* \cdot H(u, v)$$

- Korrelasjon i billedetomenet \Leftrightarrow Multiplikasjon (med $F^*(u, v)$) i frekvensdomenet
- Med $F(u, v)^*$ menes den kompleks-konjugerte til $F(u, v)$
- Det motsatte gjelder også:

$$f(x, y)^* \cdot h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$$

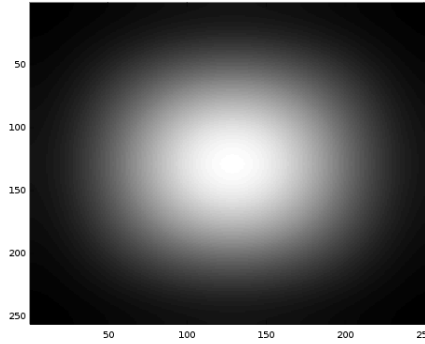
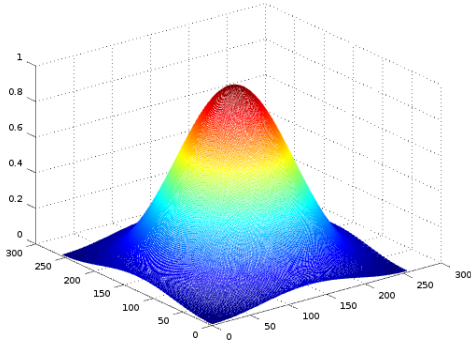
- Brukes f.eks. til templatmatching

Bortsett fra komplekskonjugeringen, $*$, er dette helt likt konvolusjonsteoremet!

Bruk av vindusfunksjoner

- Må se på bildet som periodisk
 - => Det oppstår diskontinuiteter i kantene av bildet
 - => «kunstige» bidrag på aksene i spekteret
- For å begrense slike høyfrekvente bidrag kan man bruke en vindusfunksjon og vekte dataene før DFT beregnes
 - Vindusfunksjonene modifierer pikselverdiene slik at de går mot null i enden av sekvensene
 - Lag $f_w(x,y)=f(x,y)w(x,y)$
 - Ta DFT av $f_w(x,y)$
- "Bildet" kan være en liten del av et større bilde, jfr egenskapsuttrekning

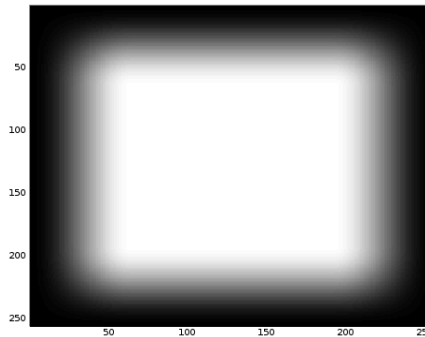
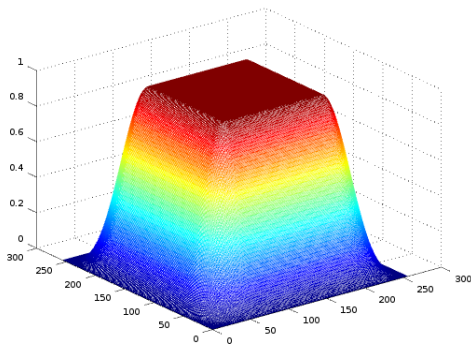
Eksempler på vindusfunksjoner



”window function”
”apodization”
”tapering”

← ”Hamming-vindu.”

```
h = hamming(N);  
w = h*h';  
fw = w .* f;
```



← ”Tukey-vindu.”

```
h = tukeywin(N);  
w = h*h';  
fw = w .* f;
```

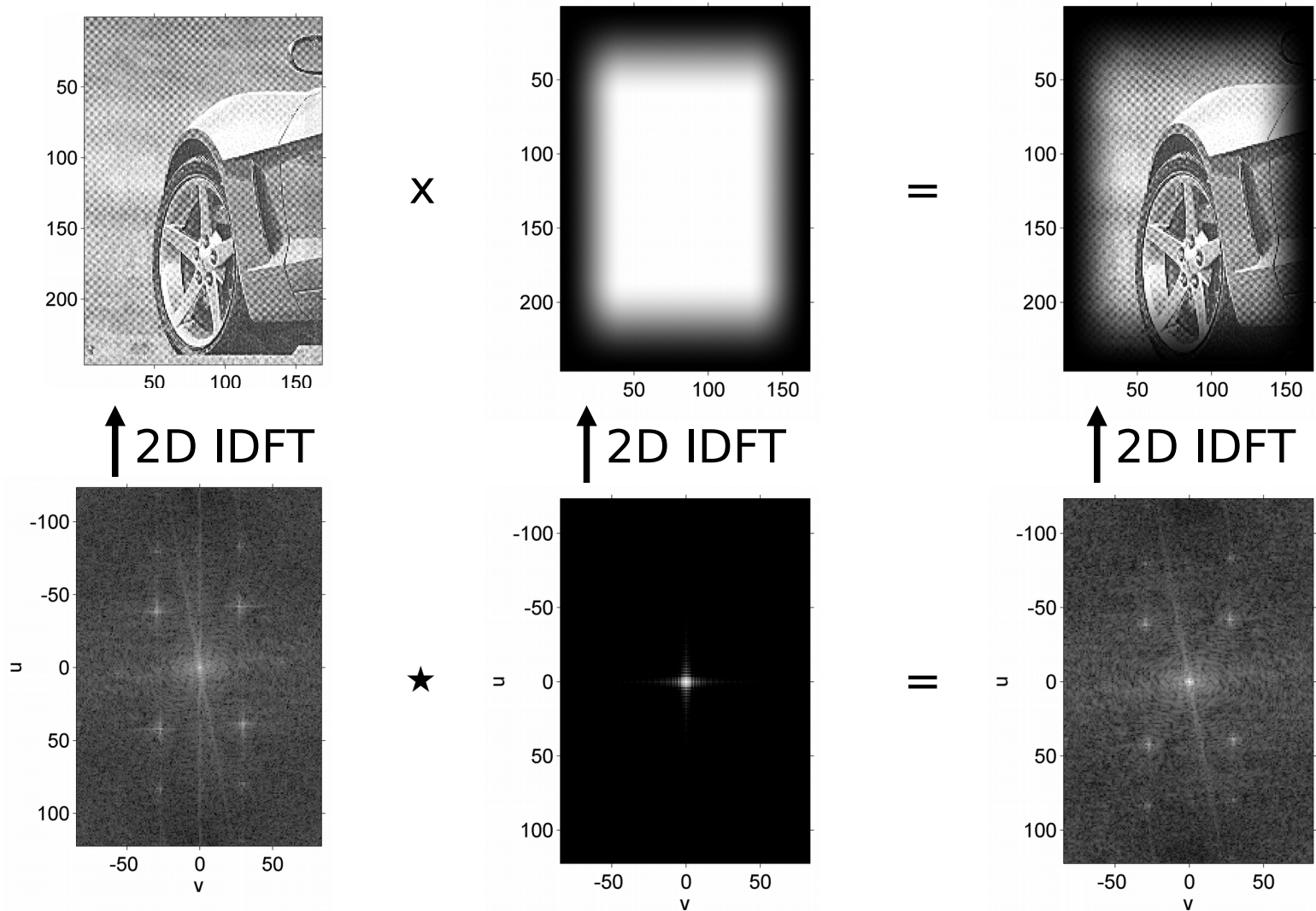
Effekten av vinduer

- Hvis vi bruker vindusfunksjon til å redusere effekten av bildekantene i spekteret gjør vi $f_w(x,y)=f(x,y)w(x,y)$ før FFT
- Dette gjør at bidragene langs aksene i Fourier-spekteret reduseres, men vi påvirker også andre frekvenser i bildet
- Effekten av en multiplikasjon i billedomenet er en konvolusjon i frekvensdomenet (konvolusjonsteoremet)
 - Multiplikasjon med en "bred klokkefunksjon" i billedomenet er ekvivalent med en konvolusjon av en "smal klokkefunksjon" i frekvensdomenet
 - Bruk av vindusfunksjon gir en "blurring" av spekteret



Jfr. konvolusjonsteoremet

Eksempel, bruk av vindusfunksjon



Vindusfunksjoner

- Det finnes **mange typer vindusfunksjoner**.
- Ofte defineres de i 1D og utvides til 2D ved matrisemultiplikasjon.
 - 1D samplet vindusfunksjon h (kolonnevektor) gir 2D-en ved hh^T
- Forrige eksempel benyttet *Tukey-vinduet*, som i 1D er definert som:

$$w(n) = \begin{cases} \frac{1}{2} \left[1 + \cos \left(\pi \left(\frac{2n}{\alpha(N-1)} - 1 \right) \right) \right] & \text{when } 0 \leq n \leq \frac{\alpha(N-1)}{2} \\ 1 & \text{when } \frac{\alpha(N-1)}{2} \leq n \leq (N-1) \left(1 - \frac{\alpha}{2} \right) \\ \frac{1}{2} \left[1 + \cos \left(\pi \left(\frac{2n}{\alpha(N-1)} - \frac{2}{\alpha} + 1 \right) \right) \right] & \text{when } (N-1) \left(1 - \frac{\alpha}{2} \right) \leq n \leq (N-1) \end{cases}$$

- Parameteren α kontrollerer skarpheten til overgangen; 0 gir et rektangulært vindu, 1 gir et glatt vindu kalt *Hann vindu*.
- Vindusfunksjoner kan også **brukes i Fourier-domenet**, da til å **definere overgangene i et filter**.
 - Butterworth og Gaussisk er vindusfunksjoner.
 - Alle vindusfunksjoner kan brukes i begge domener.

Oppsummering

- Konvolusjonsteoremet:
Konvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - «Myke» overganger -> redusere ringing
 - Analyse av konvolusjonsfiltre
 - Rask implementasjon av større konvolusjonsfiltre
- Vindusfunksjoner