

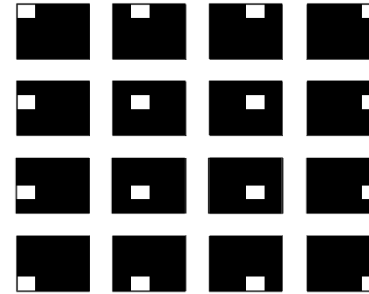
# INF2310 – 13. mars 2012

## FORELESNING 9: FOURIER-TRANFORM II

- Kort repetisjon av forrige tirsdagsforelesning
- Konvolusjonsteoremet og bruk av dette:
  - Design av lineære filtre med bestemte frekvensegenskaper
  - Analyse og rask implementasjon av konvolusjonsfiltre
- Hvordan unngå wraparound-feil
- Vindusfunksjoner

# Repetisjon: Standardbasis

### Eksempel: Standardbasis for 4x4



- Et gråtonebilde representeres vanligvis som et rutenett av gråtoneintensiteter
- Dette tilsvarer å bruke den såkalte *standardbasisen* for matriser
- Eksempel: 4x4-gråtonebilder
  - Standardbasisen er de 16 matrisene vist til venstre, der sort er 0 og hvit er 1
  - En vektet sum av disse matrisene kan unikt representere enhver 4x4-matrise/gråtonebilde

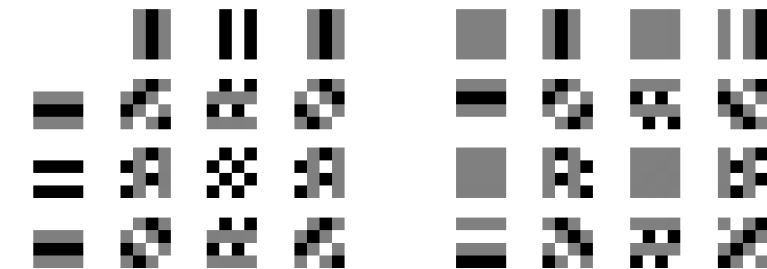
### Undereksempel:

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

$$= 1 * \begin{matrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{matrix} + 3 * \begin{matrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{matrix} + \dots + 6 * \begin{matrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{matrix}$$

# Repetisjon: Alternativ basis

- Det finnes mange andre basiser for matriser
  - Muligheten til å unikt representere enhver matrise ligger i *basis*
- 2D DFT bruker én slik basis som er basert på sinuser og cosinuser med forskjellige frekvenser
  - Disse sinusene og cosinusene er faste for en gitt bildestørrelsen (MxN) og kan representeres som hver sin mengde av bilder:

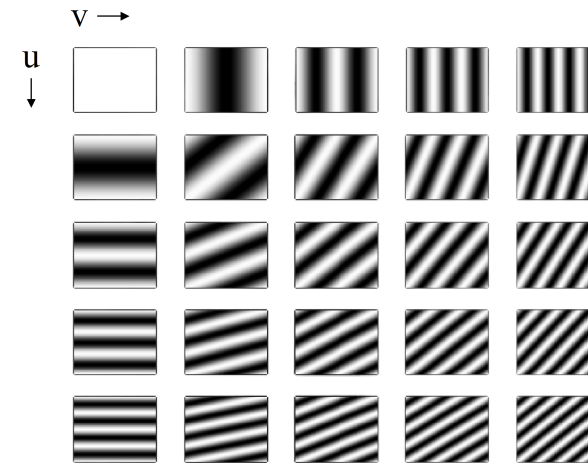


Cosinus-bildene for 4x4-bilder

Sinus-bildene for 4x4-bilder

(i bildene er sort -1, grått er 0 og hvitt er 1)

# Repetisjon: Cosinus-bilder for større bilder

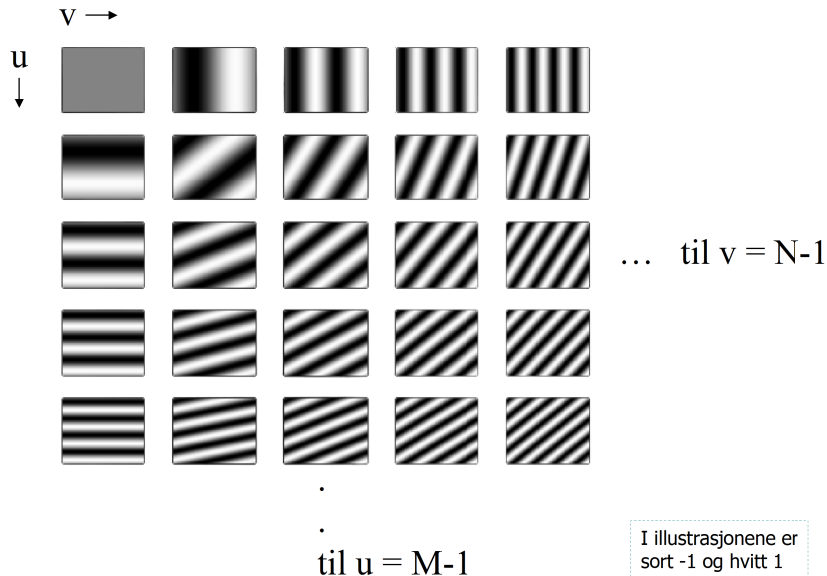


... til  $v = N-1$

til  $u = M-1$

I illustrasjonene er sort -1 og hvitt 1

## Repetisjon: Sinus-bilder for større bilder



## Repetisjon: 2D diskret Fourier-transform (DFT)

- 2D DFT av enhver  $M \times N$  matrise (til og med kompleks!) er definert som:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

for  $u = 0, 1, 2, \dots, M-1$  og  $v = 0, 1, 2, \dots, N-1$

- $x$  og  $y$  er hhv. vertikal og horisontal koordinat
- $u$  og  $v$  er hhv. vertikal og horisontal frekvens
- $j = \sqrt{-1}$  er den imaginære enheten (ofte betegnet  $i$  i matematikken)
- 2D invers diskret Fourier-transform (IDFT) er definert som:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

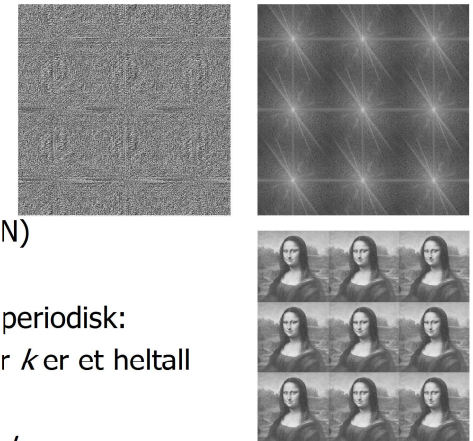
og vil *alltid perfekt reversere* basisskifte gjort av 2D DFT

## Repetisjon: Grunnleggende om 2D DFT

- For et gråtonebilde  $f(x, y)$  er realdelen av 2D DFT-ens  $(u, v)$ -frekvens gitt som summen av punktproduktet mellom  $f(x, y)$  og cosinus-bildet for frekvens  $(u, v)$ ; tilsvarende sinus-bilde vil gi imaginærdelen
  - $F(u, v)$  er (generelt) et komplekst tall
  - $F(u, v)$  er en vektet sum av *alle* gråtoneintensitetene i bildet
    - Hvert punkt av 2D DFT-en beskriver noe ved *hele* bildet og *ikke et bildepunkt*
- Magnituden av en 2D DFT kalles dets *Fourier-spekteret*
  - Beskriver hvilke frekvenser gråtonebildet inneholder
  - Sterkt knyttet til intuisjonen i Fourier-analyse

## Repetisjon: Utvalgte egenskaper ved 2D DFT

- $F(u, v)$  er periodisk:
  - $F(u, v) = F(u+kM, v+kN)$  der  $k$  er et heltall;  $k \in \mathbb{Z}$
  - $\Rightarrow F(u, v) = F(u+M, v) = F(u, v+N) = F(u+M, v+N)$
- Bildet antas indirekte å være periodisk:
  - $f(x, y) = f(x+kM, y+kN)$  der  $k$  er et heltall
- $F(u, v)$  er *konjugert symmetrisk* hvis (og bare hvis)  $f(x, y)$  er reell
  - Konjugert symmetri:  $F(u, v) = F^*(-u, -v)$
  - $\Rightarrow |F(u, v)| = |F(-u, -v)|$



# Konvolusjonsteoremet

- Konvolusjonsteoremet er en svært viktig egenskap ved 2D DFT
- Mesteparten av denne forelesningen kommer til å handle om dette
- Konvolusjonsteoremet består av to deler:

1) Når  $\Leftrightarrow$  betegner at høyresiden er 2D DFT-en til venstresiden, er:

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

Sirkelkonvolusjon i billedet  $\Leftrightarrow$  Punktvis multiplikasjon i frekvensdomenet

2) Det motsatte gjelder også:

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

Punktvis multiplikasjon i billedet  $\Leftrightarrow$  Sirkelkonvolusjon i frekvensdomenet

– Bildene  $f$  og  $h$  antatt å ha samme størrelse

- Nullutvid det minste bildet slik at de får samme størrelse

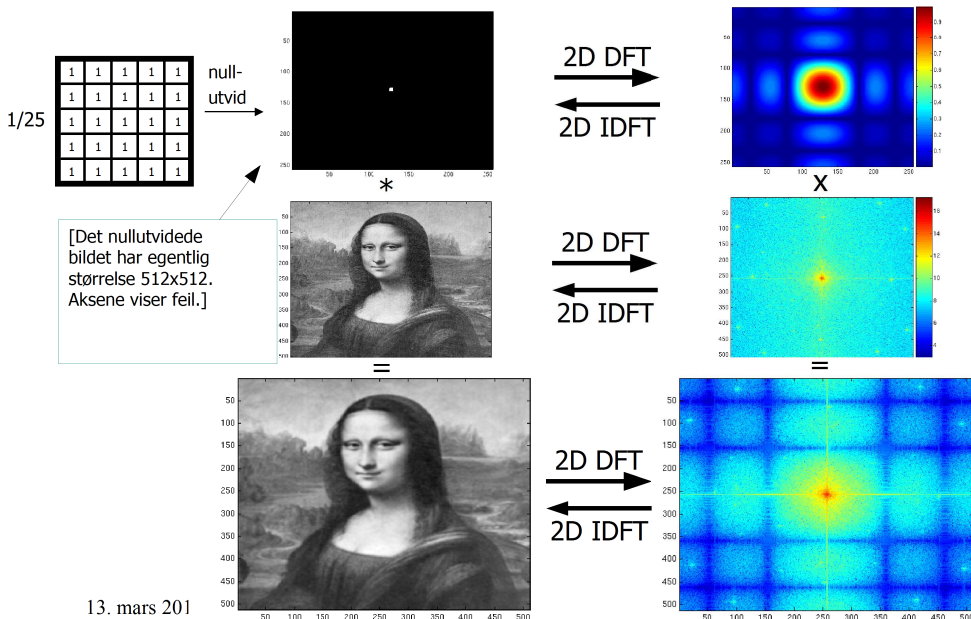
# Bevis for konvolusjonsteoremet i 1D (kursorisk)

DFT<sub>k</sub>(x ⊛ y)

↙  
Sirkelkonvolusjon

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

# Eksempel: Middelverdifilteret



# Anvendelse av konvolusjonsteoremet

- Design av lineære filtre med bestemte frekvenssegenskaper
  - Designe filteret i frekvensdomenet slik at det får de ønskelige frekvenssegenskapene
- Analyse av konvolusjonsfiltre
  - 2D DFT-en til et konvolusjonsfilter gir oss innblikk i hvordan filteret vil påvirke de forskjellige frekvenskomponentene
- Rask implementasjon av større konvolusjonsfiltre

# Filterdesign i frekvensdomenet

## Lavpassfiltre

- Slipper bare gjennom lave frekvenser (mindre enn en grense  $D_0$  som kalles filterets *cut-off-frekvens*)
  - I signalbehandling (1D) oppgis ofte  $D_0$  oppgis ofte som et tall mellom 0 og 1, da menes at cut-off er  $D_0 N/2$ ; vi kommer da til å mene at cut-off er  $D_0 \min\{M,N\}/2$

- Ideelt lavpassfilter:

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

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

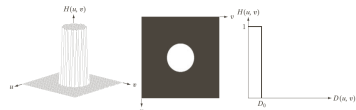


FIGURE 4.40 (a) Perspective plot of an ideal low-pass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

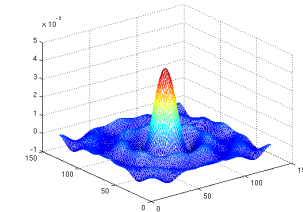
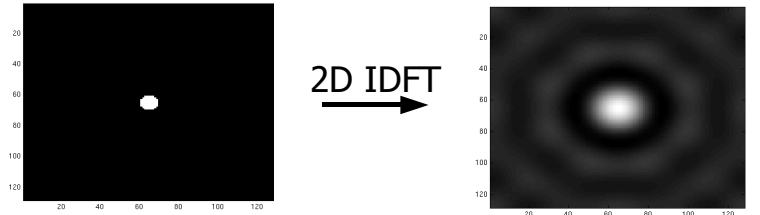
- Ordet *ideelt* kommer av at overgangene mellom 0 og 1 i  $H(u,v)$  er maksimalt raske
  - Det må ikke tolkes som *best!*

13. mars 2012

INF2310

13 / 45

## Romlig representasjon av ideelt lavpassfilter



(trunkert sinc-funksjon)

- Vi får en «ringing»-effekt i bildet
  - Husk også tommelfingerregel fra forrige forelesning:  
Smal/bred struktur i bildet  $\Leftrightarrow$  Bred/smal struktur i Fourier-spekteret

13. mars 2012

INF2310

14 / 45

## Eksempel: Ideelt lavpassfilter



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)

13. mars 2012

INF2310

15 / 45

## Matlab-eksempel: Ideelt lavpassfilter

```
f = double(imread('..'));
[M N] = size(f);
H = zeros(M,N);
D0 = 0.2 * min(M,N) / 2;

for i = 1:M
    for j = 1:N
        if sqrt( (i-floor(M/2+1))^2 + (j-floor(N/2+1))^2 ) <= D0 % +1 pga én-indeksing
            H(i,j) = 1; % og nedrunding for konjugert symmetri ved odde M og/eller N
        end
    end
end

F = fftshift( fft2(f) );
g = real( ifft2( ifftshift( F.*H ) ) );

imshow(g, []);
```

13. mars 2012

INF2310

16 / 45

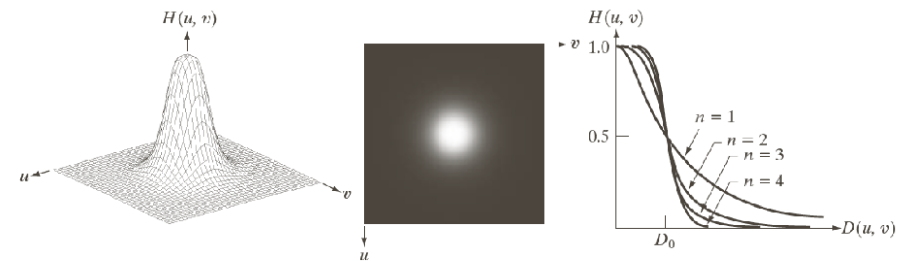
## Butterworth lavpassfilter I

- Glattere overganger mellom 0 og 1 vil redusere ringing-effekten
- Butterworth lavpassfilter av orden  $n$  er definert som:

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

- $D_0$  angir avstanden fra DC til punktet der  $H(u, v)$  har falt til halvparten av sin maksimumsverdi
- $n$  angir hvor rask overgangen er;
  - Lav filterorden ( $n$  liten):  $H(u, v)$  faller langsomt: Lite ringing
  - Høy filterorden ( $n$  stor):  $H(u, v)$  faller raskt: Mer ringing

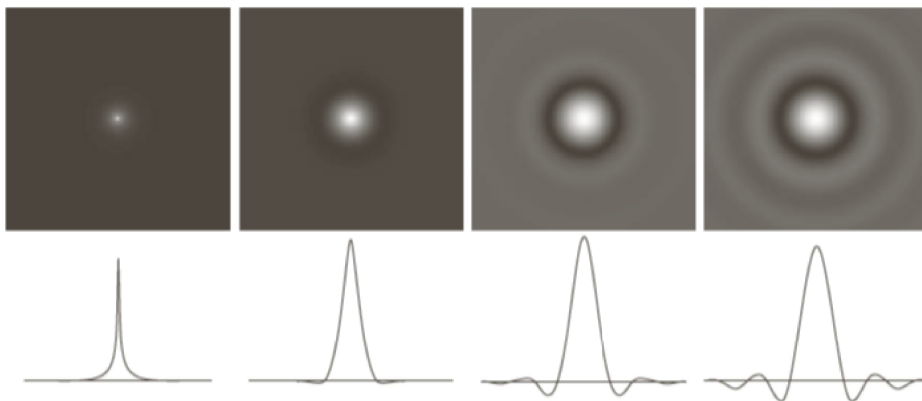
## Butterworth lavpassfilter II



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.

## Romlig representasjon av Butterworth lavpassfilter



a b c d

**FIGURE 4.46** (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is  $1000 \times 1000$  and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

## Eksempel: Butterworth lavpassfilter



$n=11$

$n=41$

$n=61$

## Gaussisk lavpassfilter I

- 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

- Standardavviket angir avstanden fra DC til punktet der  $H(u, v)$  har falt til omtrent 60 % av sin maksimumsverdi
- 2D IDFT-en av et Gaussisk lavpassfilter er også Gaussisk
  - Får ingen ringing-effekt i billedomenet!
  - Tommefingerregelen er nøyaktig tilfredsstillt her: Smal/bred struktur i bildet  $\Leftrightarrow$  Bred/smål struktur i Fourier-spekteret

## Gaussisk lavpassfilter II

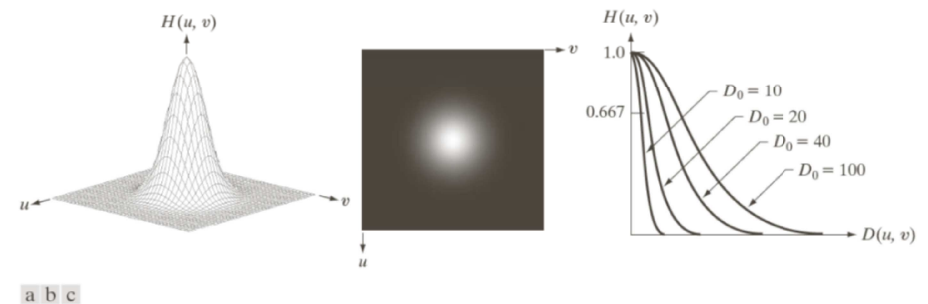


FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of  $D_0$ .

## 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:

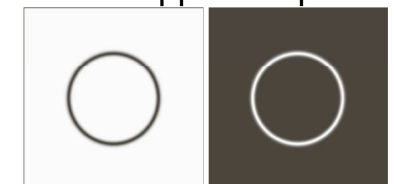
$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2D_0^2}}$$

## Båndpass- og båndstopppfiltere

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

- Kan bruke de samme overgangene som i lav-/høypassfiltre:
  - Ideelt, Butterworth, Gaussisk (finnes også mange andre)

Båndstopp Båndpass



I illustrasjonene er sort 0 og hvitt 1

# Notch-filtre

- Slipper igjennom (notch-passfiltre) eller stopper (notch-stoppfiltre) 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-stoppfiler

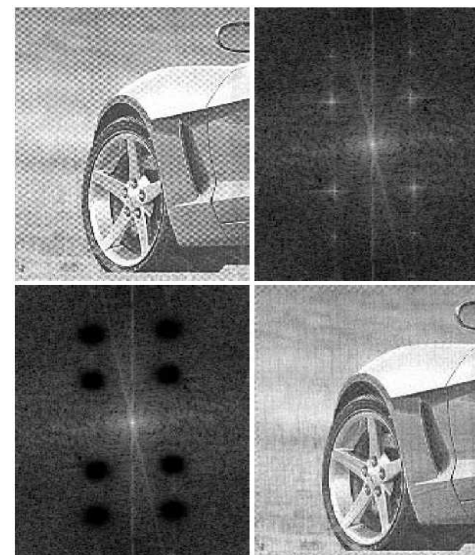
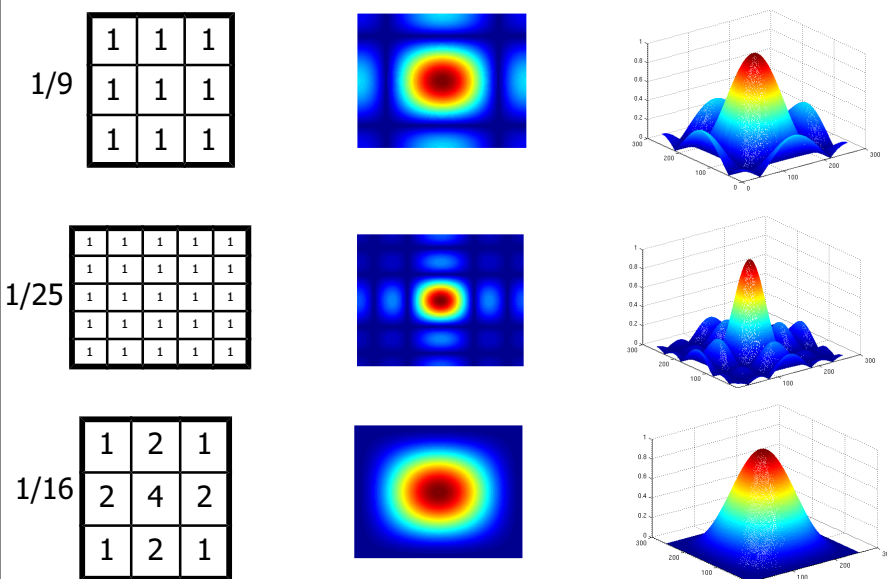


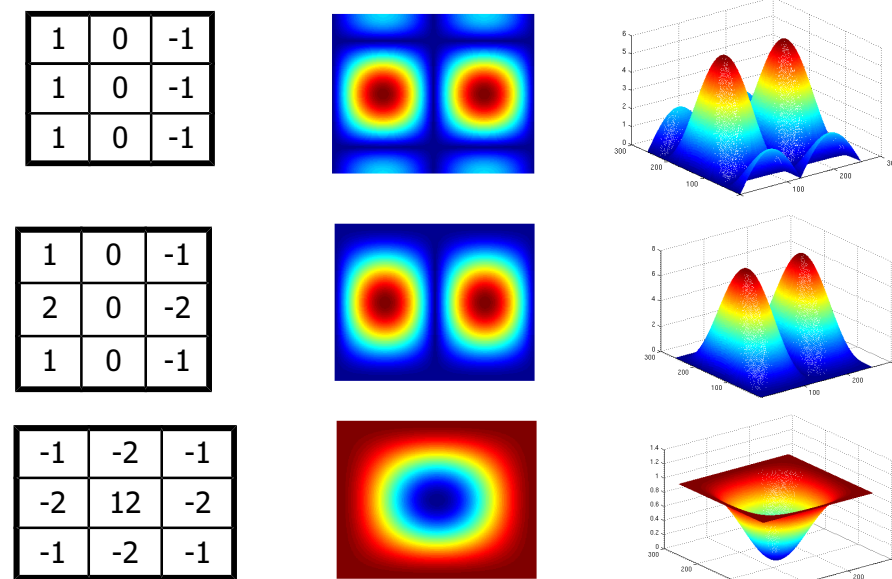
Fig. 4.64 i DIP

# Analyse av filtre

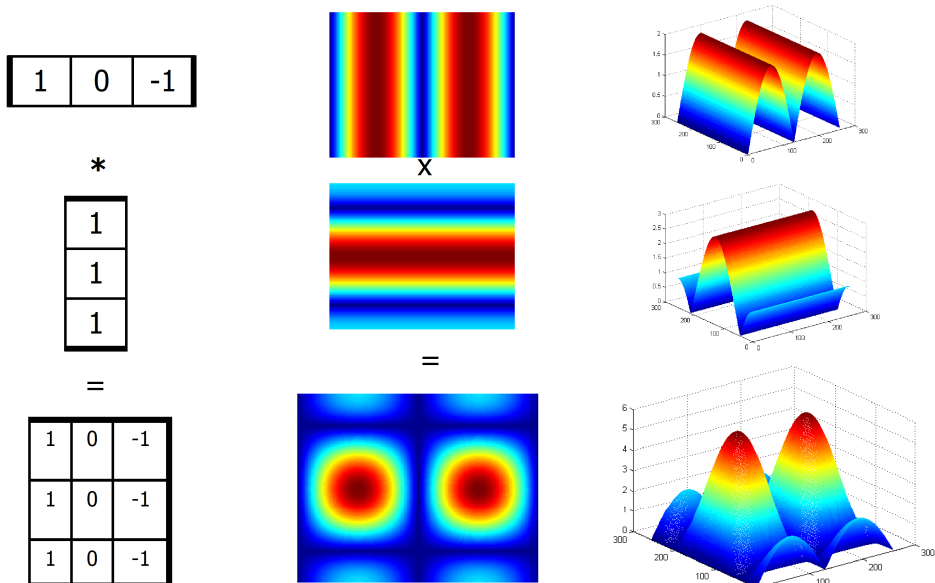
## Frekvensrespons til noen vanlige filtre I



## Frekvensrespons til noen vanlige filtre II



# Prewitt-operatoren (gradientfilter)



INF2310

# Rask implementering

## Design i bildedomenet og filtrering i frekvensdomenet

Situasjon: Skal filtrere gråtonebildet med et konvolusjonsfilter  
 Fremgangsmåte: Utføre filtreringen i frekvensdomenet

Initielt forslag til fremgangsmåte (antar at bildet er størst):

1. Beregn 2D DFT-en av bildet
2. Beregn 2D DFT-en av filteret (etter eventuelt nullutvidelse)
3. Punktmultipliser de to transformerte matrisene
4. Beregn 2D IDFT av produktet

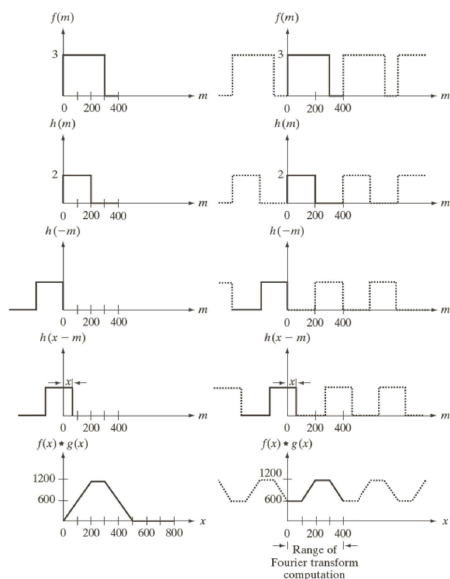
Resultatet av punkt 4 vil være det filtrerte bildet

13. mars 2012

INF2310

30 / 45

# Eksempel: Wraparound-feil i 1D



INF2310

31 / 45

# Wraparound-feil

- Når vi filtrerer bildet i bildedomenet kan vi velge hvordan vi skal takle bildekantproblemet («randproblemet»)
  - Ofte nullutvider vi bildet
- Periodisitetsegenskapen til 2D DFT gjør at sirkulær indksering er implisitt antatt når vi filtrerer i frekvensdomenet
  - Derfor betegnelsen «sirkelkonvolusjon» i konvolusjonsteoremet
- For å benytte et annet valg må bildet utvides på ønskelig måte før 2D DFT-en av det beregnes
  - Filteret må da nullutvides til det utvidede bildet
- Dersom bildet har størrelse  $M \times N$  og filteret  $m \times n$ , må det utvidede bildet minst ha størrelse  $(M+m-1) \times (N+n-1)$ 
  - Kan utvide i vilkårlig retning, ofte velges det å legge til på slutten
  - Dersom bildet er størst er  $(2M-1) \times (2N-1)$  alltid en trygg utvidelse
- Bildet forminskes (til sin opprinnelige størrelse) etter 2D IDFT-beregningen i henhold til retningen det ble utvidet

13. mars 2012

INF2310

32 / 45



## Design i billedomenet og filtrering i frekvensdomenet

Fremgangsmåte (antar at bildet er størst):

1. Utvid bildet på ønskelig måte til minst  $(M+m-1) \times (N+n-1)$
2. Beregn 2D DFT-en av det utvidede bildet
3. Beregn 2D DFT-en av filteret (etter nullutvidelse)
4. Punktmultipliser de to transformerte matrisene
5. Beregn 2D IDFT av produktet
6. Forminsk resultatet til opprinnelig størrelse (i henhold til retningen det ble utvidet i punkt 1)

Resultatet av punkt 6 vil være det filtrerte bildet med ønsket behandling av randproblemet

## Når er filtrering raskest i frekvensdomenet?

- Anta at bildet har størrelse  $N \times N$  og filteret  $n \times n$
- Filtrering i billedomenet:  $O(N^2 n)$  ved separabilitet eller oppdatering ( $O(N^2 n^2)$  ved direkte implementasjon)
- Filtrering i frekvensdomenet er  $O(N^2 \log_2 N)$ 
  - 2D FFT av bildet og filteret:  $O(N^2 \log_2 N)$
  - Punktmultiplikasjon i frekvensdomenet:  $N^2$  multiplikasjoner
  - Inverstransform av resultatet:  $O(N^2 \log_2 N)$
  - $N$  vil kunne være opptil det dobbelt for å unngå wraparound-feil
- Konklusjon: Filtrering i frekvensdomenet er raskere når filteret er stort;  $n \gg \log_2 N$  der  $\gg$  brukes (istedetfor  $>$ ) fordi frekvensimplementeringen har høyere konstantfaktor

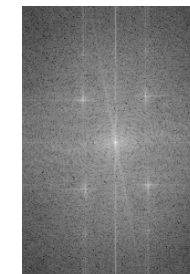
## Wraparound-feil for filtre designet i frekvensdomenet

- Wraparound-feil er ikke et isolert problem for frekvensimplementering av romlige filtre
- Vil også forekomme ved filtrering ved bruk av filtre designet i frekvensdomenet
- Dette er naturlig med tanke på dualiteten som konvolusjonsteoremet beskriver

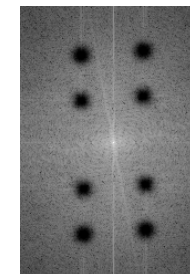
## Eksempel: Wraparound-feil i 2D I



Original



In(Fourier-spekter)

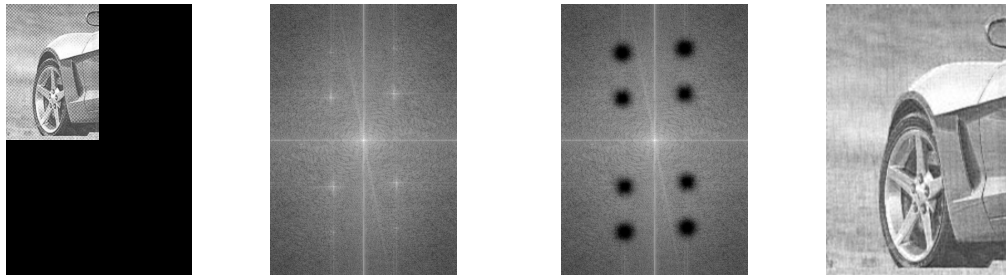


Notch-stopplert med Butterworth-overganger og  $D_0 = 12$  og  $n = 4$ .  
Notch-sentre er bestemt vha. brukerinteraksjon



Filtreringsresultat

## Eksempel: Wraparound-feil i 2D II



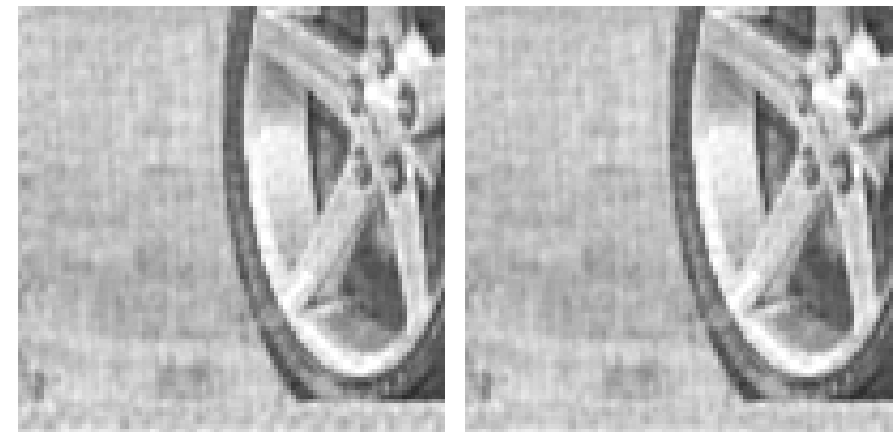
Nullutvidet original

In(Fourier-spekter) til nullutvidet org. Det ser likt ut, men er dobbelt så stort i begge retninger

Notch-stopfiltrert med Butterworth-overganger og  $D_0 = 2*12$  og  $n = 4$ . Notch-sentre er bestemt vha. brukerinteraksjon

Filtreringsresultat

## Eksempel: Wraparound-feil i 2D III



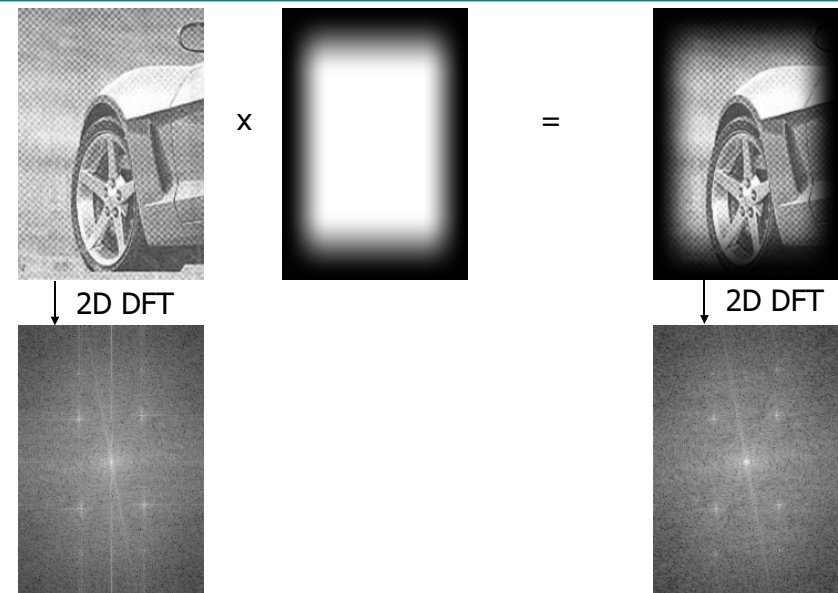
Uten nullutvidelse

Med nullutvidelse

## Wraparound-feil, diskontinuitet og vindusfunksjoner

- Dersom man *kun* skal analysere et bilde/konvolusjonsfilter trenger man ikke utvide (wraparound-feil er kun knyttet til frekvensfiltrering)
- Faktisk er det bedre å la være fordi utvidelsen kan forårsake diskontinuitet langs randen
  - Naturligvis kan det uansett være diskontinuitet langs randen når antagelsen om periodisk bilde er usann
  - Effekten av begge diskontinuitetene er tilsvarende; utsmørning i Fourier-spekteret, ofte mest synlig ved de økte høyfrekvente bidragene langs u- og v-aksen
- At bildeutvidelse kan forårsake diskontinuitet langs randen er helt generelt (og ikke spesifikt for analyse av bilder/konvolusjonsfiltre)
- En *vindusfunksjon* kan brukes for å redusere diskontinuiteten
  - En vindusfunksjon modifierer intensitetsverdiene til bildet (før eventuell utvidelse) slik at de går mot null når man går *mot* randen
  - Utføres ved å punktmultiplisere bildet med vindusfunksjonen *før* 2D DFT beregnes: 1)  $f_w(x,y)=f(x,y)w(x,y)$  deretter 2) 2D DFT av  $f_w(x,y)$

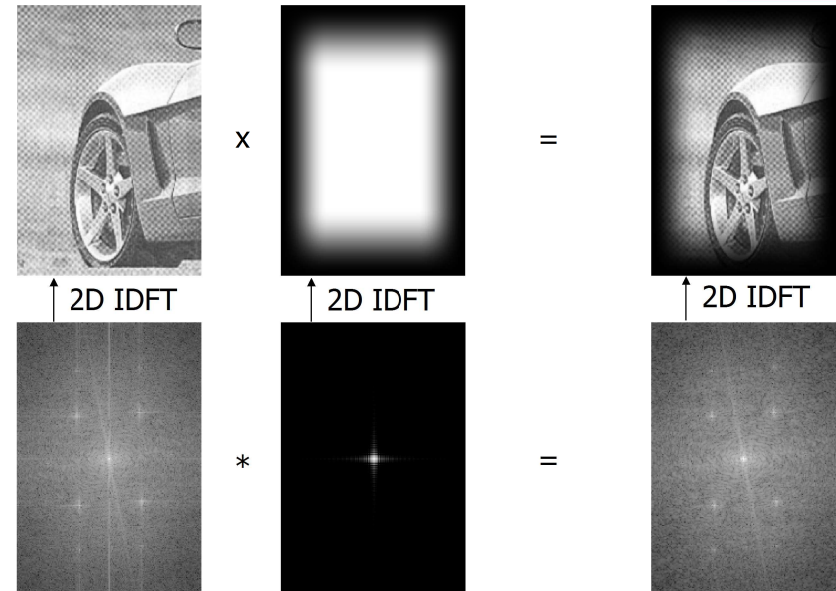
## Eksempel: Bruk av vindusfunksjon I



## Effekten av vindusfunksjoner

- Hvis vi bruker vindusfunksjon til å redusere diskontinuiteten langs bildekantene gjør vi  $f_w(x,y)=f(x,y)w(x,y)$  før 2D DFT
- Dette gjør at diskontinuitetens bidrag i Fourier-spekteret reduseres, ofte mest tydelig ved redusert aksebidrag, men vi påvirker også andre frekvenser i bildet
- Vindusfunksjonens totale effekt er lettest å tolke vha. konvolusjonsteoremet; Effekten av en punktmultiplikasjon i billededomenet er en sirkelkonvolusjon i frekvensdomenet
  - Punktmultiplikasjon med en *bred klokkefunksjon* i billededomenet er ekvivalent med en sirkelkonvolusjon av en *smal klokkefunksjon* i frekvensdomenet
  - **Bruk av vindusfunksjon gir en "blurring" av Fourier-spekteret**

## Eksempel: Bruk av vindusfunksjon II



## Vindusfunksjoner

- Det finnes mange vindusfunksjoner
- Ofte defineres de i 1D og utvides til 2D ved matrisemultiplikasjon
  - 1D samplet vindusfunksjon  $h$  (kolonnevektor) gir 2D-en  $hh^T$
- 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  $\alpha$  kontrollerer skarpheten til overgangen; 0 gir et rektangulært vindu, 1 gir et glatt vindu kalt *Hann vindu*
- Vindusfunksjoner kan også bli brukt i frekvensdomenet til å definere overgangene i et filter (lavpass, høypass osv.)
  - Butterworth og Gaussisk er vindusfunksjoner
  - Alle vindusfunksjoner kan brukes til begge deler

## Korrelasjonsteoremet

- Korrelasjon har tilsvarende 2D DFT-egenskap som konvolusjon:

$$1) \quad f(x, y) \circ h(x, y) \Leftrightarrow F(u, v)^* \cdot H(u, v)$$

Sirkelkorrelasjon i billededomenet  $\Leftrightarrow$  Punktmultiplikasjon (med den komplekskonjugerte av  $F$ ;  $F^*$ ) i frekvensdomenet

- 2) Det motsatte gjelder også:

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

- Bortsett fra komplekskonjugeringen av  $F$  eller  $f$  er dette teoremet helt likt konvolusjonsteoremet
- Kan f.eks. brukes til templatmatching

# Oppsummering

---

- Konvolusjonsteoremet:
  - Sirkelkonvolusjon i billedomenet er ekvivalent med punktmultiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
  - Design av filtre i frekvensdomenet (husk wraparound-feil!)
    - Lavpass, høypass, båndpass, båndstopp, notch
  - Analyse av konvolusjonsfiltre
    - Studere Fourier-spekteret og tolke vba. teoremet
  - Rask implementasjon av større konvolusjonsfiltre (husk wa!)
- Hindr wraparound-feil ved å utvide gråtonebildet
  - Dersom ikke ønsker sirkulær indeksering
- Punktvis multiplikasjon med en vindusfunksjon:
  - I billedomenet: Redusere bidraget til randdiskontinuiteten
  - I frekvensdomenet: Definere overgangene i et filter