INF2310 – Digital bildebehandling

FORELESNING 9

FOURIER-TRANFORM – II

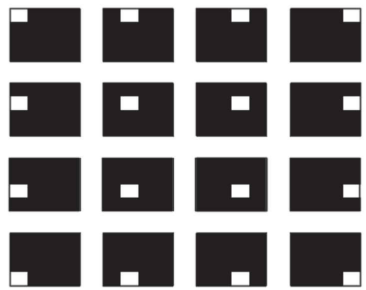
Andreas Kleppe

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

G&W: 4.6.6, 4.7.2-4.9.3, 4.10 og 5.4-5.4.3

Repetisjon: Standardbasis

Eksempel: Standardbasis for 4x4



- Et gråtonebilde representeres vanligvis som en matrise av gråtoneintensiteter.
- Dette tilsvarer å bruke den såkalte *standardbasisen* for matriser.
- Eksempel: 4x4-gråtonebilder:
  - Standardbasisen er de 16 4x4-matrisene vist til venstre.
  - En vektet sum av disse matrisene kan unikt representere enhver 4x4-matrise/-gråtonebilde.

Undereksempel:

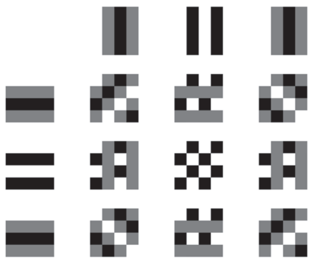
1	3	2	1
5	4	5	3
4	1	1	2
2	3	2	6

$= 1*$  $+ 3*$  $+ \ldots + 6*$ 

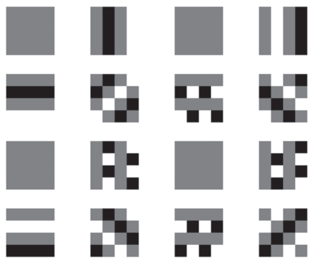
I bildene er sort 0 og hvitt 1.

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 bildestørrelse (MxN) og kan representeres som hver sin mengde av MN MxN-bilder;



Cosinus-bildene for 4x4-bilder



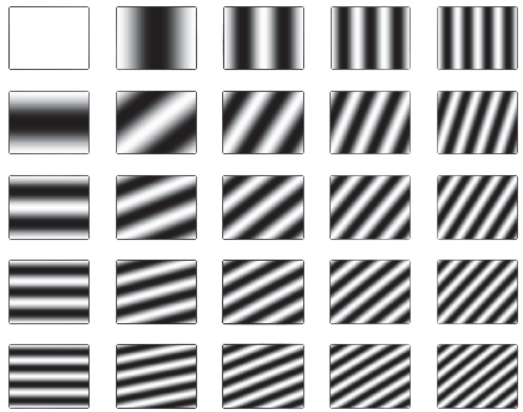
Sinus-bildene for 4x4-bilder

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

Repetisjon: Cosinus-bilder for større bilder

v →

u ↓



… til $v = N\text{-}1$

.

.

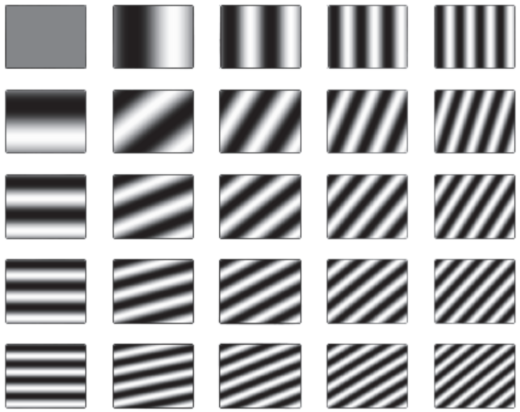
til $u = M\text{-}1$

I bildene er sort -1 og hvitt 1.

Repetisjon: Sinus-bilder for større bilder

v →

u ↓



… til $v = N\text{-}1$

.

.

til $u = M\text{-}1$

I bildene er sort -1 og hvitt 1.

Repetisjon: 2D diskret Fourier-transform (DFT)

- 2D DFT av et M x N- bilde er matrisen F som er gitt ved:

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

  for $u = 0, 1, 2, \ldots, M\text{-}1$ og $v = 0, 1, 2, \ldots, N\text{-}1$.

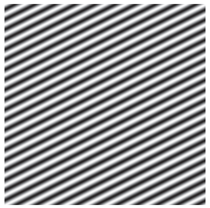
- Merk:
  - $F(u,v)$ er (generelt) et komplekst tall.
  - $F(u,v)$ er en vektet sum av **alle** gråtoneintensitetene i bildet.
- 2D invers diskret Fourier-transform (2D IDFT) er gitt ved:

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

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

- 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)

Repetisjon: Grunnleggende om 2D DFT

- real(F(u,v)) = sum( x )

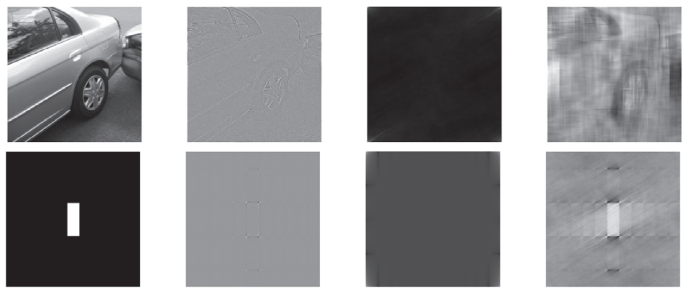
  realdelen til 2D DFT-en i frekvens (u,v) = sum(bildet punkt-multi-plisert cosinus-bildet for frekvens (u,v))

- Tilsvarende for imaginærdelen og sinus-bildet.
- **Hvert punkt** i 2D DFT-en beskriver altså noe ved **hele bildet**.

Repetisjon: 2D DFT på polarform

- Magnituden av en 2D DFT kalles **Fourier-spekteret**.
  - Beskriver hvilke frekvenser gråtonebildet inneholder.
  - Sterkt knyttet til intuisjonen i Fourier-analyse.
- For å rekonstruere det opprinnelige bildet er derimot fasen viktigere:

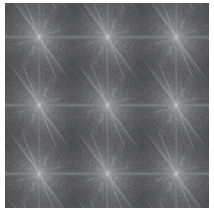
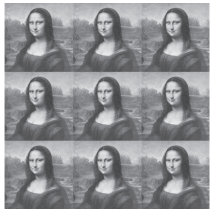
Alle bildene er uniformt rekvantifisert slik at de fyller gråtone-intervallet.



Rekonstruert ved bruk av: Original | Kun fasen | Kun spekteret | Den korrekte fasen og den andres spekter

Repetisjon: Utvalgte egenskaper ved 2D DFT

- F(u,v) er **periodisk**:
  - F(u,v) = F(u+kM,v+kN) der k er et heltall; $k \in \mathbb{Z}$
    - ⇒ 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)
    - ⇒ |F(u,v)|=|F(-u,-v)|

Konvolusjonsteoremet

- Konvolusjonsteoremet er en svært viktig egenskap ved 2D DFT.
  - Hovedtemaet i dag er anvendelser av dette teoremet.
- Konvolusjonsteoremet består av to deler:

1) Når ⇔ betegner at høyresiden er 2D DFT-en til venstresiden, og ⋆ betegner *sirkel-konvolusjon*, dvs. at sirkulær indeksering benyttes, er:

Vi bruker både * og ⋆ for å betegne konvolusjon.
Læreboka bruker ⋆

$$f \star h \quad \Leftrightarrow \quad F \times H$$

F og H er 2D DFT-en av bildene f og h, hhv.

Sirkelkonvolusjon i bildedomenet ⇔ **Punktvis multiplikasjon** i Fourier-domenet

2) Tilsvarende er også:

$$f \times h \quad \Leftrightarrow \quad F \star H$$

Punktvis multiplikasjon i bildedomenet ⇔ **Sirkelkonvolusjon** i Fourier-domenet

- Formlene antar at ***f, h, f ⋆ h* og *F ⋆ H* har samme størrelse**.

Konvolusjonsteoremet: Bevis i 1D (kursorisk)

Likhet etter definisjon, akkurat dette stedet etter definisjonen av 1D DFT

Sirkelkonvolusjon

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

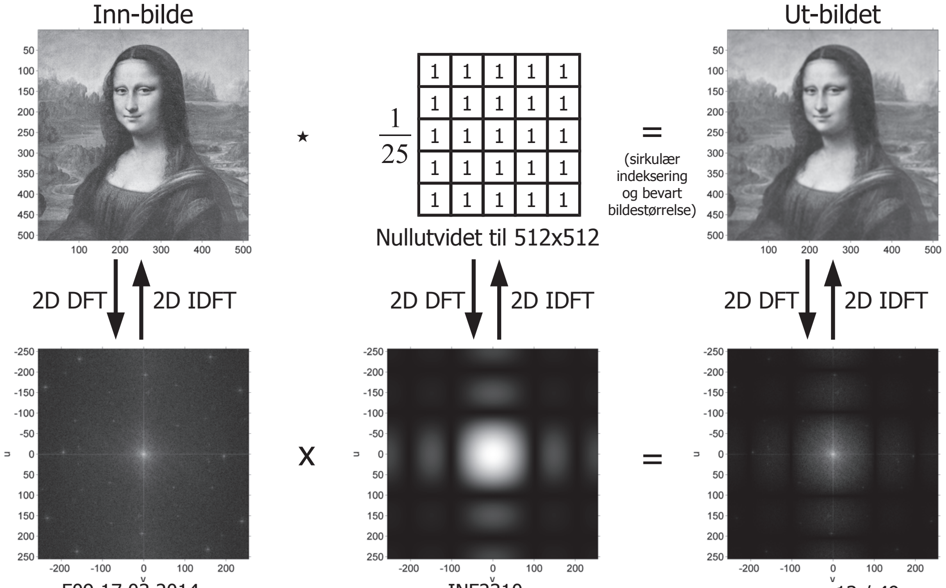
(Fra dsprelated.com)

Bruke konvolusjonsteoremet til andre konvolusjoner

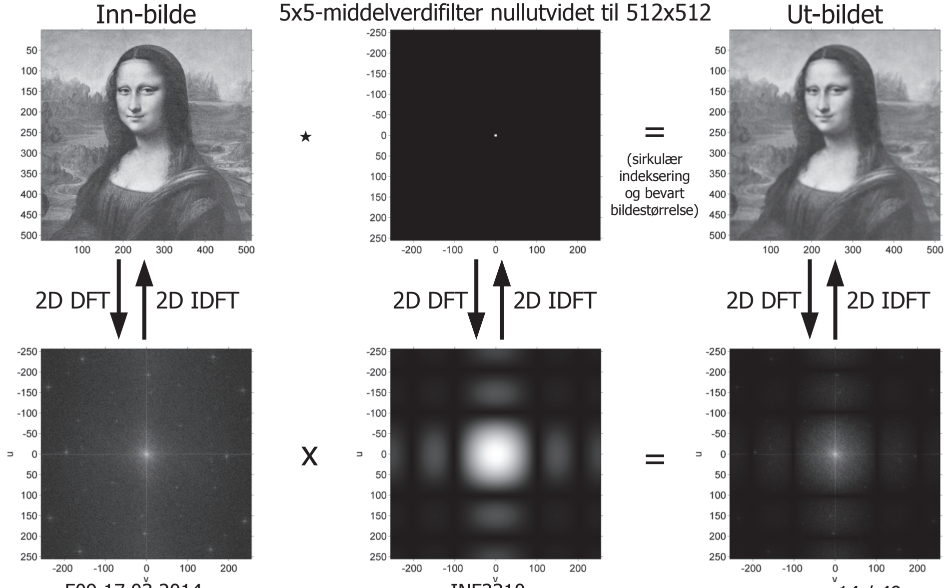
Kan konvolusjonsteoremet brukes for å konvolvere bildet f med filteret h når:

- ***h* er mindre enn *f***?
  - Ja, anvend teoremet på f og h_p der h_p er h nullutvidet til størrelsen av f.
    - Husk at det alltid er implisitt «antatt» at filteret er 0 utenfor randen når vi konvolverer, derfor er det opplagt at $f \star h = f \star h_p$.
- vi **ikke** ønsker å behandle bilderandproblemet med **sirkulær indeksering**?
  - Ja. 1) Utvid f på den ønskelig måten og nullutvid h med like mange piksler. 2) Anvend teoremet på disse utvidelsene. 3) Fjern den utvidede delen.
    - For et sentrert m xn- filter (m og n er odde) må vi utvide med minst m-1 piksler horisontalt og n-1 piksler vertikalt.
    - Hvis vi bruker teoremet uten å utvide f og h så får vi det vi kaller *wraparound-feil* ; vi implisitt antar sirkulær indeksering pga. filtrering i Fourier-domenet, men dette er ikke ønskelig behandling av bilderandproblemet.
- vi ønsker det **utvidede ut-bildet**? (Altså ønsker vi responsen overalt hvor det er overlapp mellom f og h .)
  - Ja. Som over, bare at vi nå må utvide f og h med $2(m$-$1)$ og $2(n$-$1)$ og i punkt 3 bare fjerne m-1 piksler horisontalt og n-1 piksler vertikalt.

Eksempel: 5x5-middelverdifiltrering og konvolusjonsteoremet



Eksempel: 5x5-middelverdifiltrering og konvolusjonsteoremet



Anvendelse av konvolusjonsteoremet

- Design av konvolusjonsfiltre med bestemte frekvensegenskaper.
  - Designe konvolusjonsfilteret i Fourier-domenet slik at vi har bedre kontroll på dets frekvensegenskaper.
- Analyse av konvolusjonsfiltre.
  - 2D DFT-en til et konvolusjonsfilter gir oss innblikk i hvordan filteret vil påvirke de forskjellig frekvenskomponentene.
- Rask implementasjon av større konvolusjonsfiltre.

Filterdesign i Fourier-domenet
Generelt

- Vi ønsker bare å påvirke Fourier-spekteret
  => filteret er reelt og ikke-negativt i Fourier-domenet.
- Ofte er alle **verdiene til filteret mellom 0 og 1**;
  0 fjerner og 1 bevarer den aktuelle frekvensen.
- Hvis DC i filteret er 1 så bevares bildets middelverdi.
  - Vi viste forrige mandag 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 bevares.

Ideelt lavpassfilter

- Slipper bare gjennom lave frekvenser, alle andre fjernes.
- Vi definerer frekvensene som slippes gjennom ved $D_0 \in [0, \sqrt{2}]$
  - D_0 kalles filterets *cut-off*.
  - For NxN-bilder kan vi kalle $D_0 N/2$ for filterets *cut-off-frekvens*.

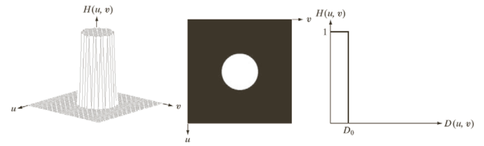


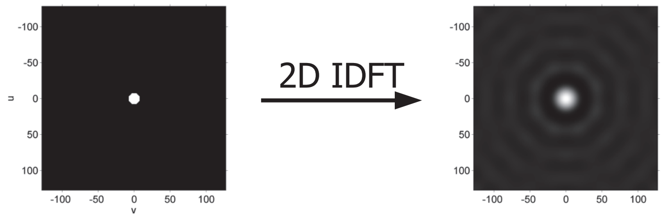
FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

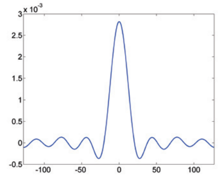
$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)>D_0\end{cases}$$

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

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

Romlig representasjon av ideelt lavpassfilter



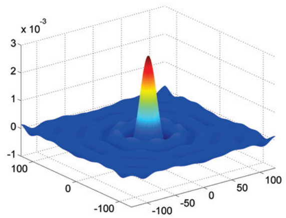


Horisontal intensitetsprofil gjennom origo

Den romlige representasjonen er en trunkert sinc-funksjon. Sinc-funksjonen er definert som:

$$\text{sinc}(x)=\frac{\sin(\pi x)}{\pi x}$$

og sinc(0) = 1.



- Vi får *ringing* i bildet.
  - Husk også tommelfingerregel fra forrige forelesning:
    Smal/bred struktur i bildet ⇔ Bred/smal struktur i Fourier-spekteret

Eksempel: Ideelt lavpassfilter

Original

Filtrert med D_0 = 0,2

Filtrert med D_0 = 0,3

I god nok oppløsning kan striper/ringinger sees ut fra markante kanter i de to filtrerte bildene. Det er dette vi kaller *ringing*.

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 floor(M/2+1)?

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

Hvis M er like: Senterpunktet i filteret er nå midt mellom piksler, men DC skal ligge i et 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-indeksering skal vi da skifte med M/2+1 = floor(M/2+1).

Analog forklaring for floor(N/2+1).

Butterworth lavpassfilter

- Glattere overganger mellom 0 og 1 vil redusere ringingen.
- Butterworth lavpassfilter av orden *n* er definert som:

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

- H(0,0) er 1 og H er strengt avtagende i alle retninger ut fra DC.
- D_0 angir avstanden fra DC til punktet der H har sunket til 0,5.
- *n* angir hvor rask senkningen er;
  - Lav filterorden (*n* liten): H faller langsomt: Lite ringing (ingenting hvis n=1)
  - Høy filterorden (*n* stor): H faller raskt: Mer ringing

Butterworth lavpassfilter

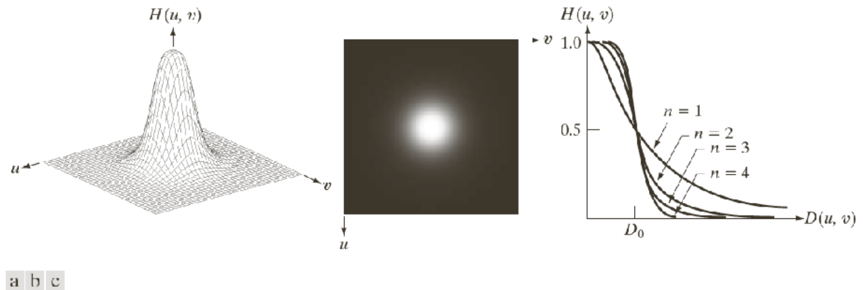


FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Romlig representasjon av Butterworth lavpassfilter

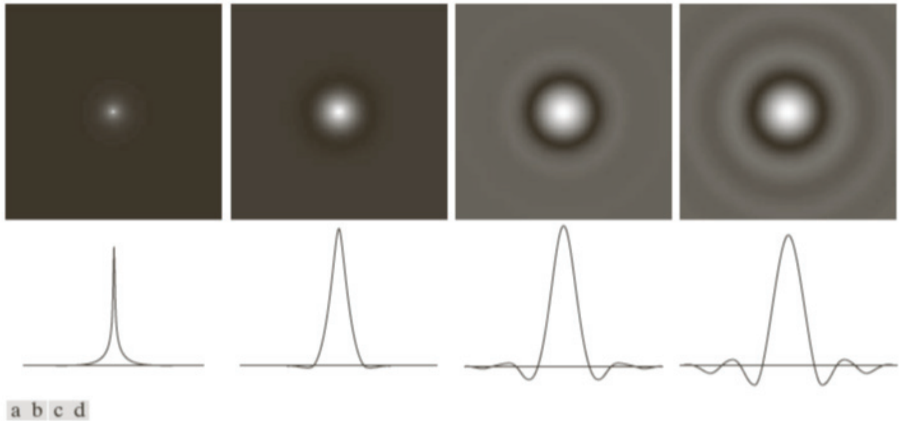


FIGURE 4.46 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 × 1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

Eksempel: Butterworth lavpassfilter

n=11 n=41 n=61

D_0 = 0,2 i alle filtreringene.

Gaussisk lavpassfilter

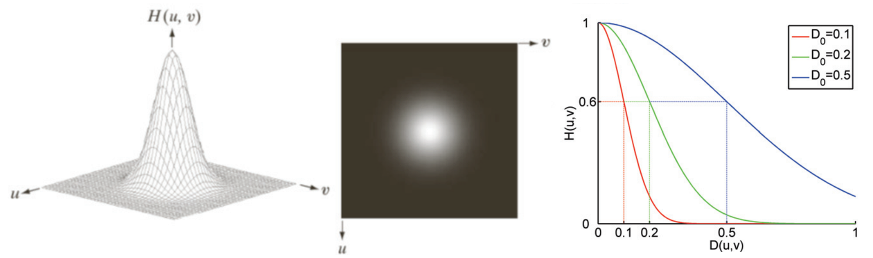
- Gaussisk lavpassfilter med spredning D_0 er definert som:

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

altså en 2D normalfordeling (uten konstantfaktoren) med DC som forventning og D_0 som standardavvik (i hver retning, 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 ≈0,6.
- 2D IDFT-en av et Gaussisk lavpassfilter er også Gaussisk.
  - Får ingen ringing i bildedomenet!

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 ⇔ Bred/smal struktur i Fourier-spekteret

Høypassfiltrering

- Et høypassfilter kan defineres ut fra et lavpassfilter:

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

- Ideelt høypassfilter:

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

- Butterworth høypassfilter:

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

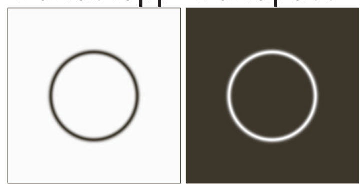
- Gaussisk høypassfilter:

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

Båndpass- og båndstoppfiltere

- Båndpassfilter: Slipper kun gjennom energien i et bestemt frekvensbånd [D_{low}, D_{high}] (eller [D_0-Ω, D_0+Ω]).
- Båndstoppfilter: Fjerner energien i et bestemt frekvensbånd [D_{low}, D_{high}].
- Kan bruke de samme overgangene som for 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) 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-stoppfilter

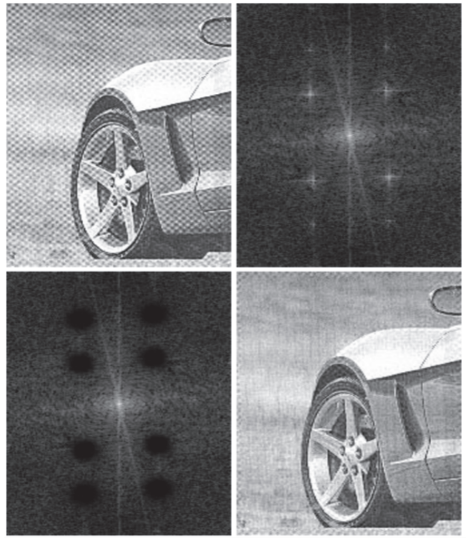
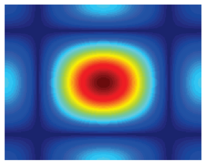


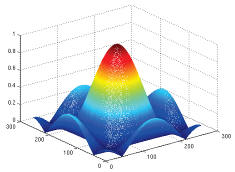
Fig. 4.64 i DIP

Analyse av filtre

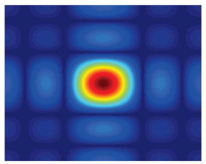
Fourier-spekteret til noen lavpassfiltre

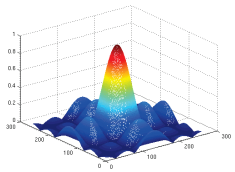
$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$



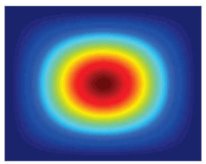


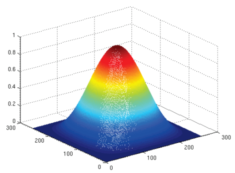
$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$





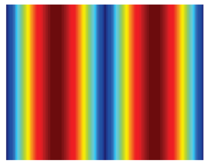
$$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

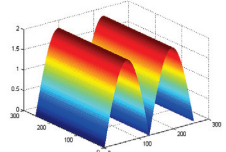




h_y i Prewitt-operatoren (en gradientoperator)

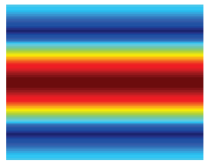
$$\begin{bmatrix}1&0&-1\end{bmatrix}$$

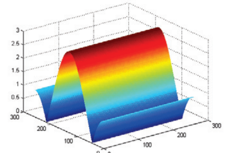




$\star$ X

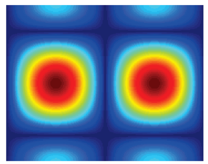
$$\begin{bmatrix}1\\1\\1\end{bmatrix}$$

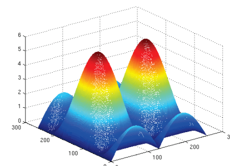




= =

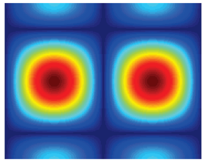
$$\begin{bmatrix}1&0&-1\\1&0&-1\\1&0&-1\end{bmatrix}$$

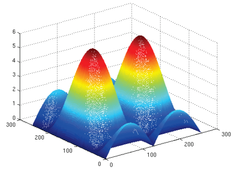




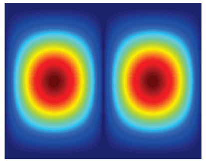
Fourier-spekteret til noen høypassfiltre

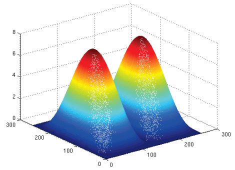
$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$



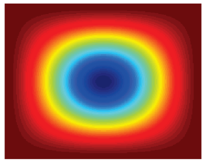


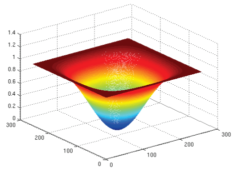
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$





$$\frac{1}{16}\begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$





Rask implementering

Design i bildedomenet og filtrering i Fourier-domenet

Situasjon: Skal filtrere et bilde med et konvolusjonsfilter.
Fremgangsmåte: Utføre filtreringen i Fourier-domenet.

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

1. Beregn 2D DFT-en av bildet.
2. Beregn 2D DFT-en av filteret etter nullutvidelse.
3. Punktmultipliser de to 2D DFT-ene.
4. Beregn 2D IDFT av produktet.

Resultatet av punkt 4 vil være det filtrerte bildet.

Eksempel: Wraparound-feil i 1D

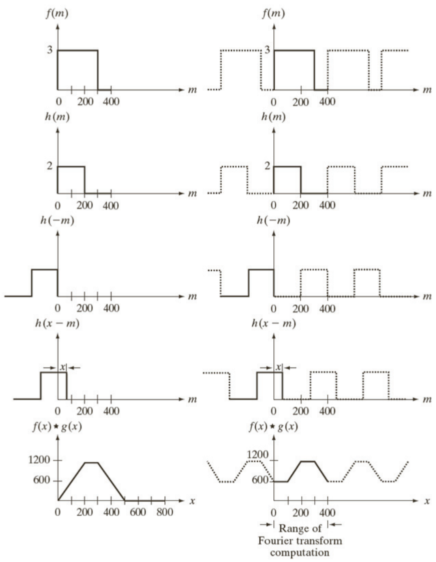


Fig. 4.28 i DIP

Wraparound-feil

- Når vi filtrerer i **bildedomenet** velger vi hvordan vi skal behandle bilderandproblemet, **ingenting er implisitt antatt**.
  - Ofte nullutvider vi inn-bildet.
- Pga. periodisitets-egenskapen til **2D DFT** er **sirkulær indeksering** implisitt antatt når vi filtrerer i Fourier-domenet.
- For å benytte en annen utvidelse av inn-bildet utvides på ønskelig måte *før* 2D DFT-en av det beregnes.
  - Filteret må da nullutvides til størrelsen av det utvidede inn-bildet.
- Dersom inn-bildet har størrelse MxN og filteret mxn, må det utvidede inn-bildet minst ha størrelse (M+m-1)x(N+n-1).
  - Kan utvide i vilkårlig retning, ofte velges det å legge til på slutten.
  - Dersom bildet er størst er (2M-1)x(2N-1) alltid tilstrekkelig.
- 2D DFT-en av det utvidede inn-bildet og det utvidede filteret beregnes.
- 2D IDFT-en av punktproduktet av 2D DFT-ene beregnes.
- Pikslene som inn-bildet med utvidet med fjernes, resultatet er ut-bildet.

Design i bildedomenet og filtrering i Fourier-domenet

Fremgangsmåte (antar at bildet er størst):

1. Utvid bildet på ønskelig måte til minst (M+m-1)x(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 2D DFT-ene.
5. Beregn 2D IDFT av produktet.
6. Fjern pikslene bildet ble utvidet med i punkt 1.

Resultatet av punkt 6 vil være det filtrerte MxN-bildet med ønsket behandling av bilderandproblemet.

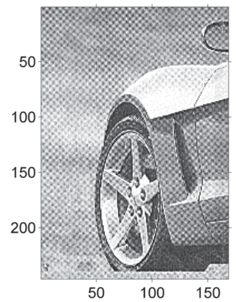
Når er filtrering raskest i Fourier-domenet?

- Anta at bildet har størrelse NxN og filteret nxn.
- Filtrering i bildedomenet:
  - $O(N^2n^2)$ ved direkte implementasjon.
  - $O(N^2n)$ ved bruk av separabilitet eller oppdatering.
- Filtrering i Fourier-domenet er $O(N^2 \log_2 N)$
  - 2D FFT av bildet og filteret: $O(N^2 \log_2 N)$.
  - Punktmultiplikasjon i Fourier-domenet: N^2 multiplikasjoner.
  - 2D IFFT av resultatet: $O(N^2 \log_2 N)$.
  - (For å unngå wraparound-feil vil N måtte være opptil det dobbelt, men dette bidrar «bare» til ytterligere økt konstantfaktor.)
- Raskere å filtrere i Fourier-domenet når **filteret er stort**.
  - n >> $\log_2 N$ der >> brukes (istedenfor >) fordi Fourier-implementeringen har **høyere konstantfaktor**.

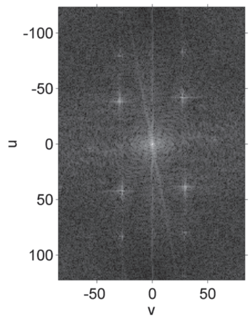
Wraparound-feil for filtre designet i Fourier-domenet

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

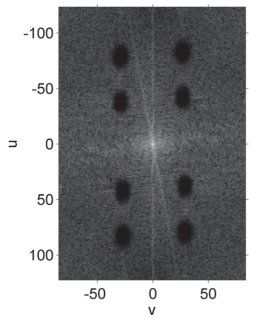
Eksempel: Wraparound-feil i 2D



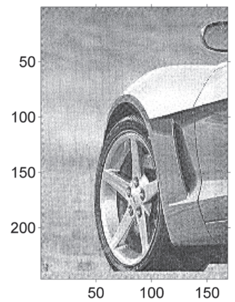
Original



ln(Fourier-spekter)

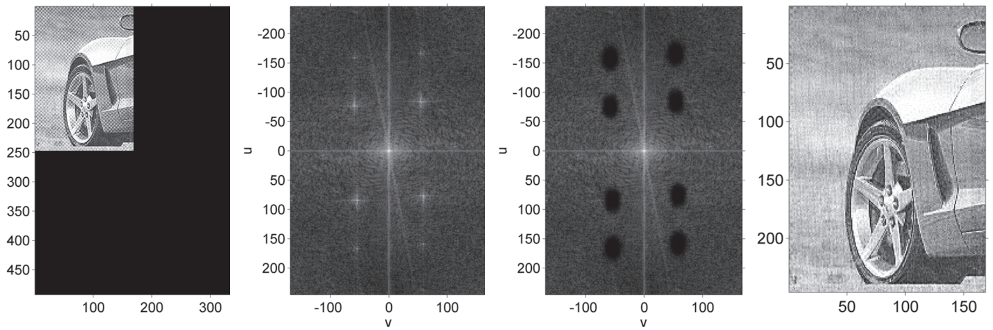


Notch-stoppfiltrert med Butterworth-overganger og D_0 = *0,1* og *n = 4*.
Notch-sentre er bestemt ved brukerinteraksjon.



Filtrerings-resultat

Eksempel: Wraparound-feil i 2D



Nullutvidet original

ln(Fourier-spekter) til nullutvidet org.

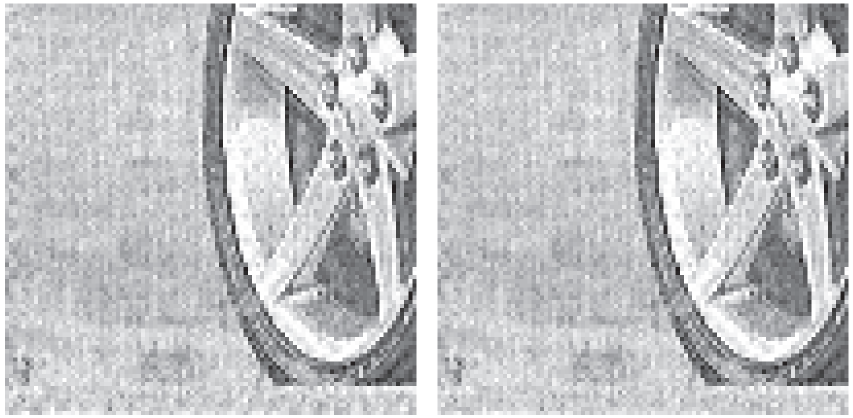
Merk at dette Fourier-spekteret har mye tydeligere akser enn spekteret til original-bildet.

Notch-stoppfiltrert med Butterworth-overganger og $D_0 = 0{,}1$ og $n = 4$.

Notch-sentre er bestemt ved brukerinteraksjon.

Filtrerings-resultat

Eksempel: Wraparound-feil i 2D



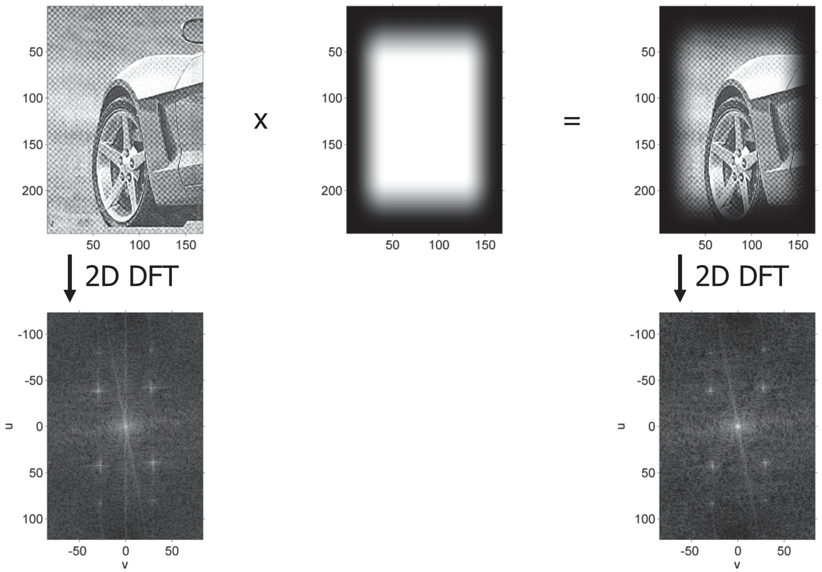
Uten nullutvidelse

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 Fourier-filtrering).
- Likevel vil 2D DFT-en og **analysen** av denne bli **påvirket av** eventuell **diskontinuitet** langs bilderanden.
  - Dersom vi ikke utvider så antas implisitt periodisk utvidelse pga. periodisitets-egenskapen til 2D DFT.
  - Dersom vi utvider kan dette også forårsake diskontinuitet.
  - Begge fremgangsmåtene vil ofte gi diskontinuitet i praksis, og resulterer i utsmørning av Fourier-spekteret langs u- og v-aksen.
- En **vindusfunksjon** kan brukes til å **redusere diskontinuiteten**.
  - En vindusfunksjon modifiserer gråtoneverdiene til bildet slik at de går mot null når man går mot randen.
  - Utføres ved å punktmultiplisere bildet med vindusfunksjonen *før* 2D DFT; 1) $f_w(x,y)=f(x,y)w(x,y)$ deretter 2) 2D DFT av f_w

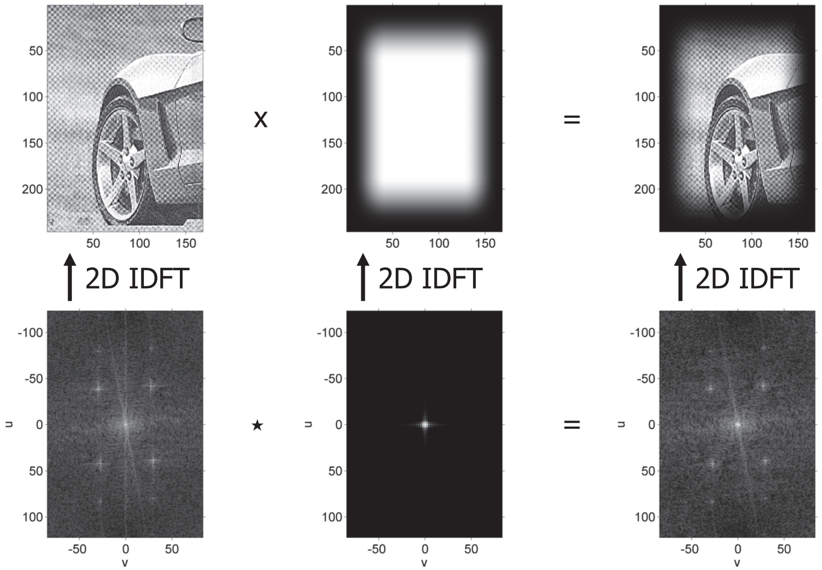
Eksempel: Bruk av vindusfunksjon



Effekten av vindusfunksjoner

- Hvis vi bruker vindusfunksjon til å redusere diskontinuiteten langs bilderanden gjør vi $f_w(x,y)=f(x,y)w(x,y)$ *før* 2D DFT.
- Dette gjør at bidrag fra bilderandens diskontinuitet reduseres i 2D DFT-en og dets Fourier-spekter.
  - Ofte mest tydelig ved redusert aksebidrag, men vi påvirker også andre frekvenser i bildet.
- Vindusfunksjonens totale effekt er lettest å forstå vba. konvolusjonsteoremet; Effekten av en punktmultiplikasjon i bildedomenet er en sirkelkonvolusjon i Fourier-domenet.
  - Punktmultiplikasjon med en *bred klokkefunksjon* i bildedomenet ⇔ sirkelkonvolusjon av en *smal klokkefunksjon* i Fourier-domenet.
  - **Bruk av vindusfunksjon glatter Fourier-spekteret.**

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:

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

  - 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.

Korrelasjonsteoremet

- Korrelasjon har tilsvarende 2D DFT-egenskap som konvolusjon:
  - Når ⇔ betegner at høyresiden er 2D DFT-en til venstresiden, og ☆ betegner *sirkel-korrelasjon* og * betegner kompleksonjugering, er:

$$f \star h \quad \Leftrightarrow \quad F^* \cdot H$$

F og H er 2D DFT-en av bildene f og h, hhv.

Sirkelkorrelasjon i bildedomenet ⇔ Punktmultiplikasjon med F^* i Fourier-domenet

  - Tilsvarende er også:

$$f^* \cdot h \quad \Leftrightarrow \quad F \star H$$

Punktmultiplikasjon med f^* i bildedomenet ⇔ Sirkelkorrelasjon i Fourier-domenet

- **Bortsett fra kompleksonjugeringen** av F eller f og at vi skal korrelere istedenfor å konvolvere er dette teoremet **helt likt konvolusjonsteoremet**.
- Kan f.eks. brukes til «template matching».

Oppsummering

- Konvolusjonsteoremet:
  - Sirkelkonvolusjon i bildedomenet er ekvivalent med punktmultiplikasjon i Fourier-domenet, og omvendt.
- Anvendelser:
  - Design av konvolusjonsfiltre i Fourier-domenet (husk wraparound-feil!).
    - Lavpass, høypass, båndpass, båndstopp, notch.
  - Analyse av konvolusjonsfiltre.
    - Studere Fourier-spekteret og tolke vba. teoremet.
  - Rask implementasjon av større konvolusjonsfiltre (husk w.a.-feil!).
- Hindre wraparound-feil ved å utvide gråtonebildet.
  - Dersom vi ikke ønsker å benytte sirkulær indeksering.
- Punktvis multiplikasjon med en vindusfunksjon:
  - I bildedomenet: Redusere bilderanddiskontinuiteten.
  - I Fourier-domenet: Definere overgangene i et filter.