

IN2070 - Fourier 2

17. mars 2021

- Repetisjon fra forrige uke
- Konvolusjonsteoremet
- Filtre og filtrering i frekvensdomenet
- Bruk av vinduer

Repetisjon: Basisbilder

1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

← Eksempel på basisbilder for 4×4 bilder

Eksempel:

$$\begin{bmatrix} 2 & 3 & 0 & 5 \\ 1 & 5 & 4 & 0 \\ 6 & 3 & 1 & 2 \\ 5 & 1 & 0 & 7 \end{bmatrix} = 2 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \dots + 7 \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Digitale gråtonebilder av størrelse $M \times N$ kan representeres ved en vektet sum av disse $M \times N$ sinus- og cosinus-bildene:

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

for frekvensene

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

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

Digitale gråtonebilder av størrelse $M \times N$ kan representeres ved en vektet sum av disse $M \times N$ sinus- og cosinus-bildene:

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

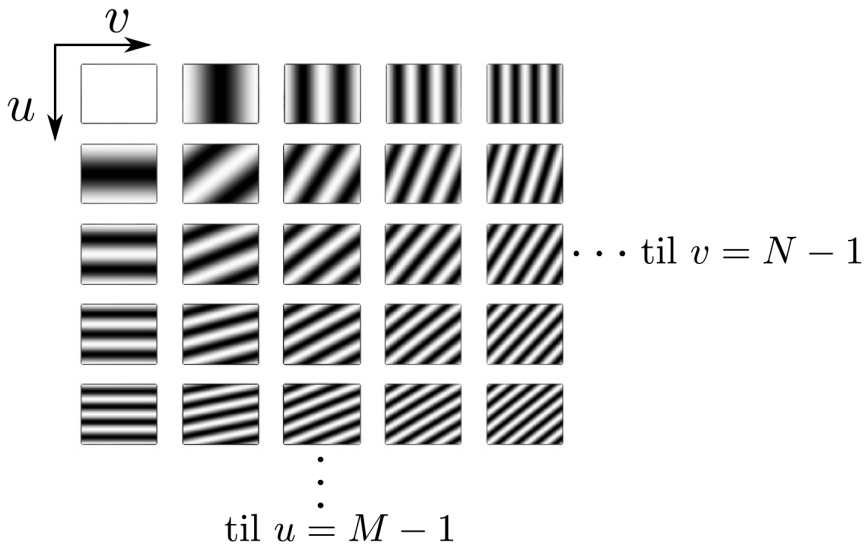
for frekvensene

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

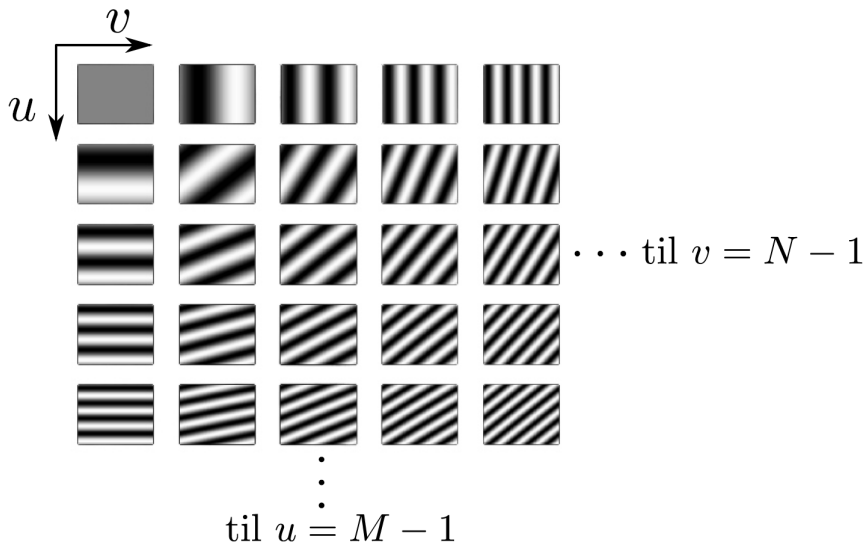
$$v = 0, 1, \dots, N - 1$$

Alle digitale gråtonebilder kan representeres ved hjelp av disse basisene!

Repetisjon: Basisbilder - cosinus



Repetisjon: Basisbilder - sinus



Repetisjon: 2D diskret Fouriertransformasjon (DFT)

2D Fourier - transformasjon:

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

Repetisjon: 2D diskret Fouriertransformasjon (DFT)

2D Fourier - transformasjon:

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

Eulers formel:

$$e^{j\theta} = \cos(\theta) + j \sin(\theta)$$

Repetisjon: 2D diskret Fouriertransformasjon (DFT)

2D Fourier - transformasjon:

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

Eulers formel:

$$e^{j\theta} = \cos(\theta) + j \sin(\theta)$$

Ved å bruke formelen, får vi:

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

Repetisjon: 2D diskret Fouriertransformasjon (DFT)

2D Fourier - transformasjon:

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

Eulers formel:

$$e^{j\theta} = \cos(\theta) + j \sin(\theta)$$

Ved å bruke formelen, får vi:

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

Invers transformasjon:

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

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Amplitudespekteret er gitt ved $|F(u, v)|$

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Amplitudespekteret er gitt ved $|F(u, v)|$
- Konjugert symmetri: Er $f(x, y)$ reell, så er

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

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Amplitudespekteret er gitt ved $|F(u, v)|$
- Konjugert symmetri: Er $f(x, y)$ reell, så er $F(u, v) = (F(-u, -v))^*$
- Forskyver spekteret for å få origo midt i bildet

- Fouriertransformasjonen $F(u, v)$ er periodisk,

$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Amplitudespekteret er gitt ved $|F(u, v)|$
- Konjugert symmetri: Er $f(x, y)$ reell, så er $F(u, v) = (F(-u, -v))^*$
- Forskyver spekteret for å få origo midt i bildet
- 2D DFT er separabel i to 1D DFT

- Fouriertransformasjonen $F(u, v)$ er periodisk,

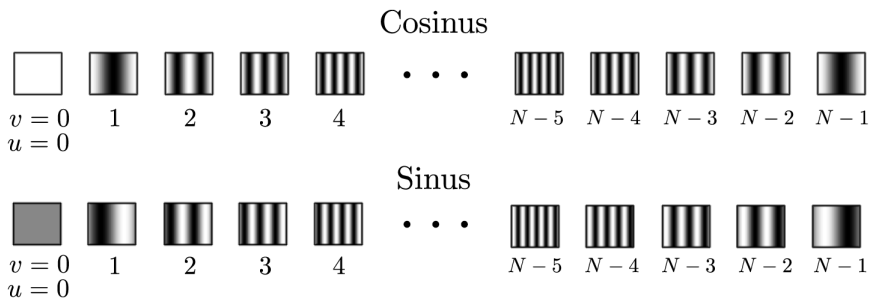
$$F(u, v) = F(u + kM, v + kN)$$

- Bildet $f(x, y)$ er implisitt periodisk,

$$f(x, y) = f(x + kM, y + kN)$$

- Amplitudespekteret er gitt ved $|F(u, v)|$
- Konjugert symmetri: Er $f(x, y)$ reell, så er $F(u, v) = (F(-u, -v))^*$
- Forskyver spekteret for å få origo midt i bildet
- 2D DFT er separabel i to 1D DFT
- Shift-teoremet: $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-2\pi j(\frac{ux_0}{M} + \frac{vy_0}{N})}$

Repetisjon: Symmetri i basisbildene



Konvolusjon i billedomenet \Leftrightarrow elementvis multiplikasjon i frekvensdomenet,

$$f * h \Leftrightarrow F \odot H$$

Konvolusjon i billedomenet \Leftrightarrow elementvis multiplikasjon i frekvensdomenet,

$$f * h \Leftrightarrow F \odot H$$

Motsatt gjelder også:

$$f \odot h \Leftrightarrow F * H,$$

altså elementvis multiplikasjon i billedomenet \Leftrightarrow konvolusjon i frekvensdomenet

Konvolusjon i billedomenet \Leftrightarrow elementvis multiplikasjon i frekvensdomenet,

$$f * h \Leftrightarrow F \odot H$$

Motsatt gjelder også:

$$f \odot h \Leftrightarrow F * H,$$

altså elementvis multiplikasjon i billedomenet \Leftrightarrow konvolusjon i frekvensdomenet

Vi snakker om sirkelkonvolusjon her!

Anvendelser av konvolusjonsteoremet

- Analyse av konvolusjonsfiltre
- Fouriertransformasjonen til et filter h gir oss innblikk i frekvensresponsen til filteret

Anvendelser av konvolusjonsteoremet

- Analyse av konvolusjonsfiltre
- Fouriertransformasjonen til et filter h gir oss innblikk i frekvensresponsen til filteret
- Filterdesign
 - Kan designe filter i frekvens- og billedomenet

- Analyse av konvolusjonsfiltre
- Fouriertransformasjonen til et filter h gir oss innblikk i frekvensresponsen til filteret
- Filterdesign
 - Kan designe filter i frekvens- og billedomenet
 - Kan implementeres for konvolusjon i billedomenet, eller som multiplikasjon i frekvensdomenet

- Analyse av konvolusjonsfiltre
- Fouriertransformasjonen til et filter h gir oss innblikk i frekvensresponsen til filteret
- Filterdesign
 - Kan designe filter i frekvens- og billedomenet
 - Kan implementeres for konvolusjon i billedomenet, eller som multiplikasjon i frekvensdomenet
 - Både bilde og filter må ha samme størrelse hvis de skal elementvis multipliseres (nullutvide)

Anvendelser av konvolusjonsteoremet

- Analyse av konvolusjonsfiltre
- Fouriertransformasjonen til et filter h gir oss innblikk i frekvensresponsen til filteret
- Filterdesign
 - Kan designe filter i frekvens- og billedomenet
 - Kan implementeres for konvolusjon i billedomenet, eller som multiplikasjon i frekvensdomenet
 - Både bilde og filter må ha samme størrelse hvis de skal elementvis multipliseres (nullutvide)
- Implementasjon
- Store filtre kan implementeres raskere i frekvensdomenet

Eksempel: Middelverdifilteret

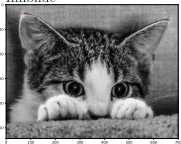
$\frac{1}{25}$

Filter

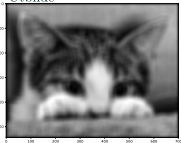
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

*

Innbilde



Utvide



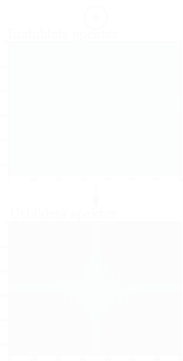
Skaler 1/25 middelværdi filter



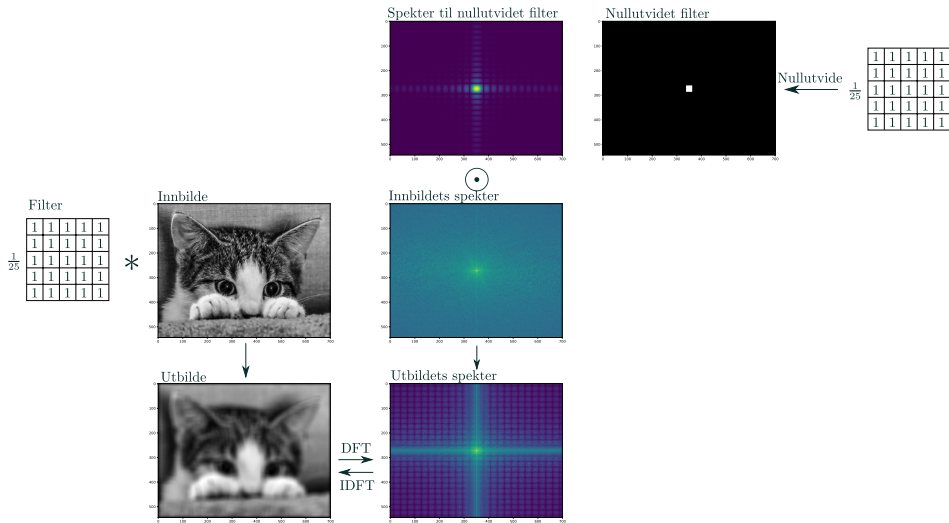
Middelværdi filter



Middelværdi filter



Eksempel: Middelverdifilteret



Konvolusjonsteoremet, en utledning i 1D

$$\begin{aligned}\text{DFT}_k(\mathbf{x} \circledast \mathbf{y}) &= \sum_{n=0}^{N-1} (\mathbf{x} \circledast \mathbf{y})[n] e^{-2\pi j \frac{nk}{N}} \\ &= \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N-1} x[m] y[n-m] \right) e^{-2\pi j \frac{nk}{N}} \\ &= \sum_{m=0}^{N-1} x[m] \sum_{n=0}^{N-1} y[n-m] e^{-2\pi j \frac{nk}{N}}\end{aligned}$$

Fra shift-teoremet, er $\sum_{n=0}^{N-1} y[n-m] e^{-2\pi j \frac{nk}{N}} = Y[k] e^{-2\pi j \frac{mk}{N}}$

Konvolusjonsteoremet, en utledning i 1D

$$\begin{aligned}\text{DFT}_k(\mathbf{x} \circledast \mathbf{y}) &= \sum_{n=0}^{N-1} (\mathbf{x} \circledast \mathbf{y})[n] e^{-2\pi j \frac{nk}{N}} \\ &= \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N-1} x[m] y[n-m] \right) e^{-2\pi j \frac{nk}{N}} \\ &= \sum_{m=0}^{N-1} x[m] \sum_{n=0}^{N-1} y[n-m] e^{-2\pi j \frac{nk}{N}}\end{aligned}$$

Fra shift-teoremet, er $\sum_{n=0}^{N-1} y[n-m] e^{-2\pi j \frac{nk}{N}} = Y[k] e^{-2\pi j \frac{mk}{N}}$

$$\begin{aligned}\text{DFT}_k(\mathbf{x} \circledast \mathbf{y}) &= \sum_{m=0}^{N-1} x[m] \sum_{n=0}^{N-1} y[n-m] e^{-2\pi j \frac{nk}{N}} \\ &= \left(\sum_{m=0}^{N-1} x[m] e^{-2\pi j \frac{mk}{N}} \right) Y[k] \\ &= X[k] Y[k]\end{aligned}$$

Implementere filtrering av et filter i frekvensdomenet:

1. Beregn DFT (**fft**) av bildet

Implementere filtrering av et filter i frekvensdomenet:

1. Beregn DFT (**fft**) av bildet
2. Beregn DFT (**fft**) av filterkjernen, med eventuell nullutvidelse

Implementere filtrering av et filter i frekvensdomenet:

1. Beregn DFT (**fft**) av bildet
2. Beregn DFT (**fft**) av filterkjernen, med eventuell nullutvidelse
3. Gjør en elementvis multiplikasjon av de to transformerte matrisene

Design i romlig domene og filtrering i frekvensdomenet

Implementere filtrering av et filter i frekvensdomenet:

1. Beregn DFT (**fft**) av bildet
2. Beregn DFT (**fft**) av filterkjernen, med eventuell nullutvidelse
3. Gjør en elementvis multiplikasjon av de to transformerte matrisene
4. Transformer resultatet tilbake til billedet gjennom invers DFT (IDFT, **ifft**)

Husk at filteret og bildet må ha samme størrelse!

Design i romlig domene og filtrering i frekvensdomenet

Implementere filtrering av et filter i frekvensdomenet:

1. Beregn DFT (**fft**) av bildet
2. Beregn DFT (**fft**) av filterkjernen, med eventuell nullutvidelse
3. Gjør en elementvis multiplikasjon av de to transformerte matrisene
4. Transformer resultatet tilbake til billededomenet gjennom invers DFT (IDFT, **ifft**)

Husk at filteret og bildet må ha samme størrelse!

Det er snakk om sirkelkonvolusjon, må eventuelt nullutvide bildet dersom randen skal håndteres på en alternativ måte

- Ønsker reelle konvolusjonskjerne

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:
0 fjerner frekvens og 1 bevarer frekvens

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:
0 fjerner frekvens og 1 bevarer frekvens
- Hvis nullfrekvens ($u = 0, v = 0$, DC-komponenten), i filteret er 1, så bevares bildets middelvei

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:
0 fjerner frekvens og 1 bevarer frekvens
- Hvis nullfrekvens ($u = 0, v = 0$, DC-komponenten), i filteret er 1, så bevares bildets middelværdi
 - DC er summen av alle gråtoneverdiene

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:
0 fjerner frekvens og 1 bevarer frekvens
 - Hvis nullfrekvens ($u = 0, v = 0$, DC-komponenten), i filteret er 1, så bevares bildets middelværdi
 - DC er summen av alle gråtoneverdier
 - Hvis DC i filteret er 1, så vil DC i utbildet bli lik DC i innbildet

- Ønsker reelle konvolusjonskjerne
- (Konjugert) symmetri i Fourierdomenet
- Ofte er verdiene i filteret mellom 0 og 1:
0 fjerner frekvens og 1 bevarer frekvens
- Hvis nullfrekvens ($u = 0, v = 0$, DC-komponenten), i filteret er 1, så bevares bildets middelværdi
 - DC er summen av alle gråtoneverdier
 - Hvis DC i filteret er 1, så vil DC i utbildet bli lik DC i innbildet
- summen av gråtoner bevares!

Filterdesign i Fourier-domenet: Lavpassfiltre

- Slipper gjennom lave frekvenser

Filterdesign i Fourier-domenet: Lavpassfiltre

- Slipper gjennom lave frekvenser
 - Cut-off-frekvens: En grense D_0 som bestemmer at alle frekvenser lavere enn D_0 skal slippes gjennom

- Slipper gjennom lave frekvenser
 - Cut-off-frekvens: En grense D_0 som bestemmer at alle frekvenser lavere enn D_0 skal slippes gjennom
 - D_0 er ofte et tall mellom 0 og 1

Filterdesign i Fourier-domenet: Lavpassfiltre

- Slipper gjennom lave frekvenser
 - Cut-off-frekvens: En grense D_0 som bestemmer at alle frekvenser lavere enn D_0 skal slippes gjennom
 - D_0 er ofte et tall mellom 0 og 1
- Ideelt lavpassfilter:



a b c

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

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

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

Filterdesign i Fourier-domenet: Lavpassfiltre

- Slipper gjennom lave frekvenser
 - Cut-off-frekvens: En grense D_0 som bestemmer at alle frekvenser lavere enn D_0 skal slippes gjennom
 - D_0 er ofte et tall mellom 0 og 1
- Ideelt lavpassfilter:

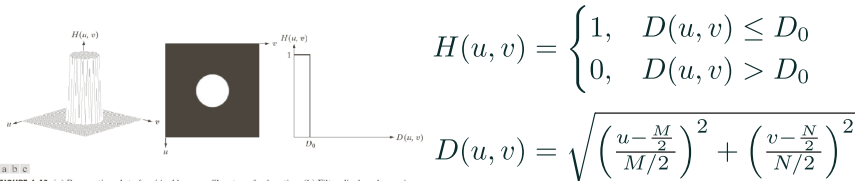


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

- “Ideelt” fra at filteret var 0 eller 1 for *alle* frekvenser.

Filterdesign i Fourier-domenet: Lavpassfiltre

- Slipper gjennom lave frekvenser
 - Cut-off-frekvens: En grense D_0 som bestemmer at alle frekvenser lavere enn D_0 skal slippes gjennom
 - D_0 er ofte et tall mellom 0 og 1
- Ideelt lavpassfilter:

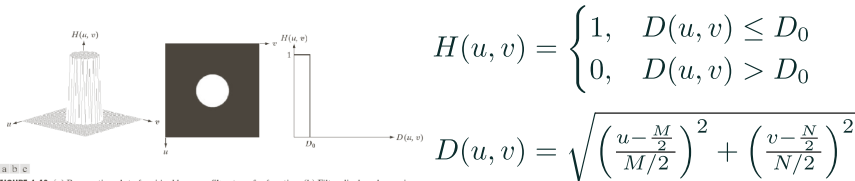


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

- “Ideelt” fra at filteret var 0 eller 1 for *alle* frekvenser. Urealiserbart, da inverstransformasjon vil ikke konvergere numerisk.

Kode-eksempel i Matlab: Enkelt/ideelt lavpass filter

```
f = double(imread(<ditt-bilde>));
[M,N] = size(f);
H = zeros(M,N);
D0 = 0.2;
for u = 0:M-1
    for v = 0:N-1
        if sqrt( ((u-floor(M/2))/(M/2))^2 + ...
                ((v-floor(N/2))/(N/2))^2 ) <= D0
            H(u+1,v+1) = 1;
        end
    end
end
F = fftshift( fft2(f) );
g = ifft2( ifftshift( F.*H ) );
% Fjern eventuelle imaginærtall grunnet avrundingsfeil:
g = real(g);
% Vis fram filtrert bilde:
imagesc(g);
```

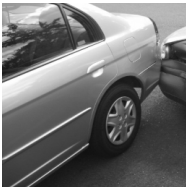
Kode-eksempel i Matlab: Enkelt/ideelt lavpass filter

```
f = double(imread(<ditt-bilde>));
[M,N] = size(f);
H = zeros(M,N);
D0 = 0.2;
for u = 0:M-1
    for v = 0:N-1
        if sqrt( ((u-floor(M/2))/(M/2))^2 + ...
                ((v-floor(N/2))/(N/2))^2 ) <= D0
            H(u+1,v+1) = 1;
        end
    end
end
end
F = fftshift( fft2(f) );
g = ifft2( ifftshift( F.*H ) );
% Fjern eventuelle imaginærtall grunnet avrundingsfeil:
g = real(g);
% Vis fram filtrert bilde:
imagesc(g);
```

Python har tilsvarende fremgangsmåte! Modulen `numpy.fft` inneholder `fft2`, `ifft2`, `fftshift` og `ifftshift`.

Eksempel: ideel lavpass

Innbilde

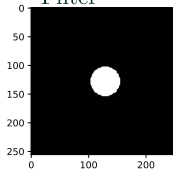


Filtering der $D_0 = 0.2$

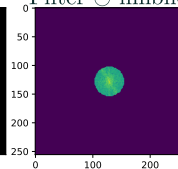
Utbilde



Filter

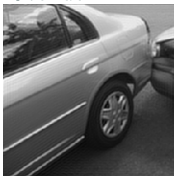


Filter \odot innbildets DFT

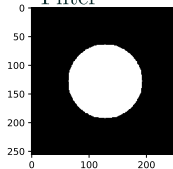


Filtering der $D_0 = 0.5$

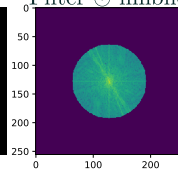
Utbilde



Filter

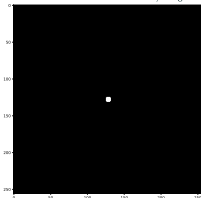


Filter \odot innbildets DFT



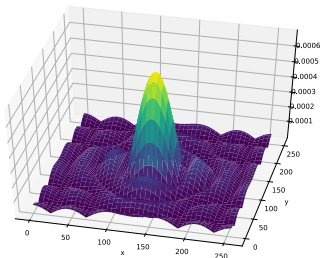
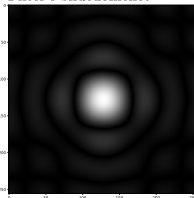
Romlig representasjon av et ideelt lavpass filter

Filter i frekvensdomenet, $D_0 = 0.03$



IDFT →

Filter i billedomenet



Vi får en “ringing”-effekt i bildet

Merk: Bred/smål struktur i bilde- og frekvensdomenet

Butterworth lavpassfilter

- Glattere funksjoner brukes til å redusere ringing-effekten

Butterworth lavpassfilter

- Glattere funksjoner brukes til å redusere ringing-effekten
- Butterworth lavpassfilter av orden n er ett eksempel:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

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

Butterworth lavpassfilter

- Glattere funksjoner brukes til å redusere ringing-effekten
- Butterworth lavpassfilter av orden n er ett eksempel:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

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

- Her beskriver D_0 avstanden der $H(u, v)$ blir halvparten av sin maksverdi

Butterworth lavpassfilter

- Glattere funksjoner brukes til å redusere ringing-effekten
- Butterworth lavpassfilter av orden n er ett eksempel:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

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

- Her beskriver D_0 avstanden der $H(u, v)$ blir halvparten av sin maksverdi
- n påvirker mengden ringing:
 - n liten (lav filterorden): $H(u, v)$ faller langsommere - mindre ringing

Butterworth lavpassfilter

- Glattere funksjoner brukes til å redusere ringing-effekten
- Butterworth lavpassfilter av orden n er ett eksempel:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

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

- Her beskriver D_0 avstanden der $H(u, v)$ blir halvparten av sin maksverdi
- n påvirker mengden ringing:
 - n liten (lav filterorden): $H(u, v)$ faller langsommere - mindre ringing
 - n stor (høy filterorden): $H(u, v)$ faller raskt - mer ringing

Butterworth lavpassfilter

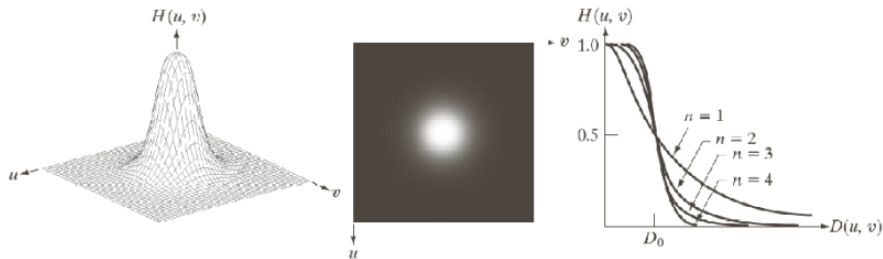
- Glattere funksjoner brukes til å redusere ringing-effekten
- Butterworth lavpassfilter av orden n er ett eksempel:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

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

- Her beskriver D_0 avstanden der $H(u, v)$ blir halvparten av sin maksverdi
- n påvirker mengden ringing:
 - n liten (lav filterorden): $H(u, v)$ faller langsommere - mindre ringing
 - n stor (høy filterorden): $H(u, v)$ faller raskt - mer ringing
- Andre mulige alternativer: Gaussisk, Bartlett, Blackman, Hamming, Hanning

Eksempel: Butterworth lavpassfilter - profiler med ulike n



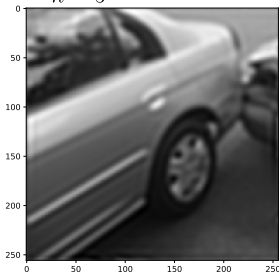
a b c

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.

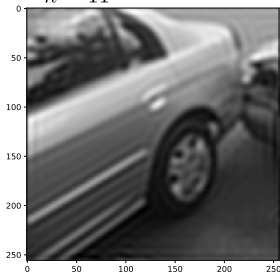
Eksempel: Butterworth lavpassfilter med ulike n på bilder

$$D_0 = 0.2$$

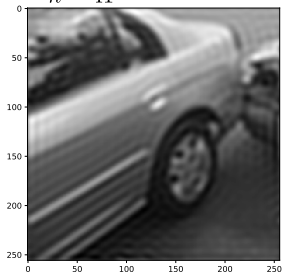
$$n = 5$$



$$n = 11$$



$$n = 41$$



- Gaussisk lavpassfilter med spredning D_0 :

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

med $D(u, v)$ som afstandsmål på tilsvarende måde som i Butterworth lavpass filteret.

- Gaussisk lavpassfilter med spredning D_0 :

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

med $D(u, v)$ som afstandsmål på tilsvarende måde som i Butterworth lavpass filteret.

- $H(0, 0) = 1$ og er strengt avtagende i alle retninger

- Gaussisk lavpassfilter med spredning D_0 :

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

med $D(u, v)$ som afstandsmål på tilsvarende måde som i Butterworth lavpass filteret.

- $H(0, 0) = 1$ og er strengt aftagende i alle retninger
- D_0 angir avstand fra DC til avstanden der $H(u, v) \approx 0.6$

- Gaussisk lavpassfilter med spredning D_0 :

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

med $D(u, v)$ som afstandsmål på tilsvarende måde som i Butterworth lavpass filteret.

- $H(0, 0) = 1$ og er strengt avtagende i alle retninger
- D_0 angir avstand fra DC til avstanden der $H(u, v) \approx 0.6$
- 2D IDFT av et Gaussisk lavpassfilter er også Gaussisk

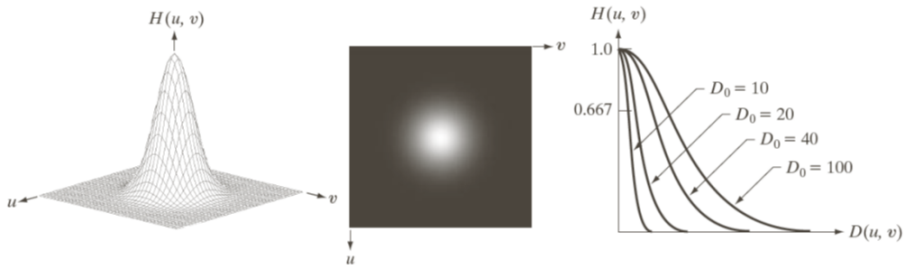
- Gaussisk lavpassfilter med spredning D_0 :

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

med $D(u, v)$ som afstandsmål på tilsvarende måde som i Butterworth lavpass filteret.

- $H(0, 0) = 1$ og er strengt avtagende i alle retninger
 - D_0 angir avstand fra DC til avstanden der $H(u, v) \approx 0.6$
 - 2D IDFT av et Gaussisk lavpassfilter er også Gaussisk
- 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 .

- Et høypassfilter kan defineres ut fra et lavpassfilter,

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

- Et høypassfilter kan defineres ut fra et lavpassfilter,

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

- Ideelt høypassfilter:

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

- Et høypassfilter kan defineres ut fra et lavpassfilter,

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

- Ideelt høypassfilter:

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

- Butterworth høypassfilter:

$$H(u, v) = \frac{1}{1 + \left(\frac{D_0}{D(u, v)}\right)^{2n}}$$

- Et høypassfilter kan defineres ut fra et lavpassfilter,

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

- Ideelt høypassfilter:

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

- Butterworth høypassfilter:

$$H(u, v) = \frac{1}{1 + \left(\frac{D_0}{D(u, v)}\right)^{2n}}$$

- Gaussisk høypassfilter:

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

Båndpass- og båndstoppfiltre

- **Båndpassfilter:** Slipper gjennom bidrag som hører til frekvenser kun i et bestemt intervall (frekvensbånd) $[D_{\text{lav}}, D_{\text{høy}}]$
- **Båndstoppfilter:** Fjerner bidrag til frekvenser i et frekvensbånd $[D_{\text{lav}}, D_{\text{høy}}]$

Båndpass- og båndstoppfiltere

- **Båndpassfilter:** Slipper gjennom bidrag som hører til frekvenser kun i et bestemt intervall (frekvensbånd) $[D_{lav}, D_{høy}]$
- **Båndstoppfilter:** Fjerner bidrag til frekvenser i et frekvensbånd $[D_{lav}, D_{høy}]$
- Butterworth båndstoppfilter:

$$H_{\text{stopp}}(u, v) = \frac{1}{1 + \left(\frac{D(u, v) W}{D^2(u, v) - D_0^2} \right)^{2n}}$$

der W er bredden på båndet

Båndpass- og båndstoppfiltere

- **Båndpassfilter:** Slipper gjennom bidrag som hører til frekvenser kun i et bestemt intervall (frekvensbånd) $[D_{\text{lav}}, D_{\text{høy}}]$
- **Båndstoppfilter:** Fjerner bidrag til frekvenser i et frekvensbånd $[D_{\text{lav}}, D_{\text{høy}}]$
- Butterworth båndstoppfilter:

$$H_{\text{stopp}}(u, v) = \frac{1}{1 + \left(\frac{D(u, v) W}{D^2(u, v) - D_0^2} \right)^{2n}}$$

der W er bredden på båndet

- Butterworth båndpassfilter:

$$H_{\text{pass}}(u, v) = 1 - H_{\text{stopp}}(u, v)$$

Båndpass- og båndstoppfiltere

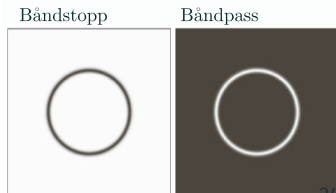
- **Båndpassfilter:** Slipper gjennom bidrag som hører til frekvenser kun i et bestemt intervall (frekvensbånd) $[D_{lav}, D_{høy}]$
- **Båndstoppfilter:** Fjerner bidrag til frekvenser i et frekvensbånd $[D_{lav}, D_{høy}]$
- Butterworth båndstoppfilter:

$$H_{\text{stopp}}(u, v) = \frac{1}{1 + \left(\frac{D(u, v)W}{D^2(u, v) - D_0^2} \right)^{2n}}$$

der W er bredden på båndet

- Butterworth båndpassfilter:

$$H_{\text{pass}}(u, v) = 1 - H_{\text{stopp}}(u, v)$$



- Slipper gjennom (notch-passfiltre) eller stopper (notch-stopppiltre) bidrag fra frekvenser i mindre predefinerte områder i Fourier-spekteret

- Slipper gjennom (notch-passfiltre) eller stopper (notch-stopppiltre) bidrag fra frekvenser i mindre predefinerte områder i Fourier-spekteret
- Kan bruke overganger som bl.a ideelt, Butterworth, Gaussisk

- Slipper gjennom (notch-passfiltre) eller stopper (notch-stopppfiltre) bidrag fra frekvenser i mindre predefinerte områder i Fourier-spekteret
- Kan bruke overganger som bl.a ideelt, Butterworth, Gaussisk
- Pro: Kan være svært nyttige
- Con: Trenger interaktivitet for å definere aktuelle områder

Eksempel: Notch-stopppfilter

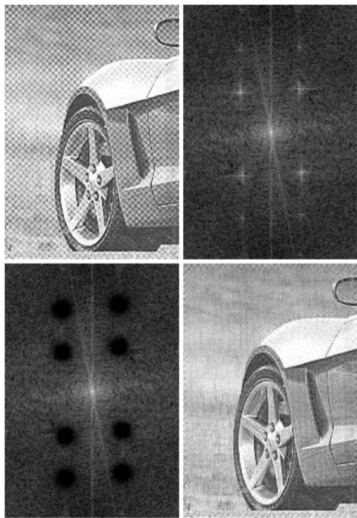


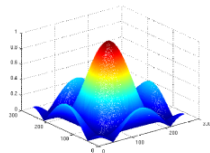
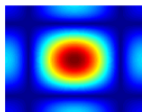
Fig. 4.64 i DIP

Analyse av filtre - Frekvensrespons til noen vanlige filtre

Filtre

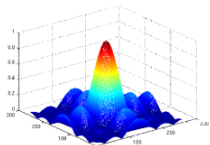
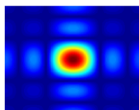
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



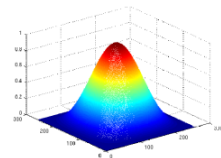
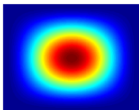
$\frac{1}{25}$

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



$\frac{1}{16}$

1	2	1
2	4	2
1	2	1



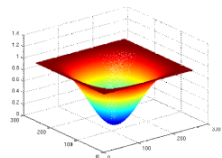
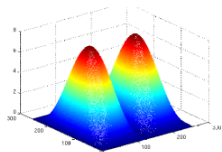
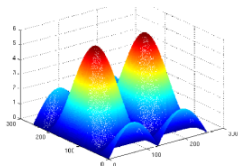
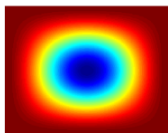
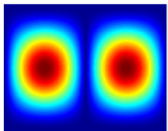
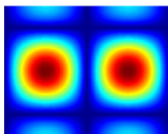
Filtre

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

Frekvensrespons



Filtre

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

*

1
1
1

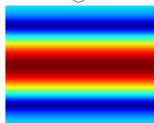
||

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

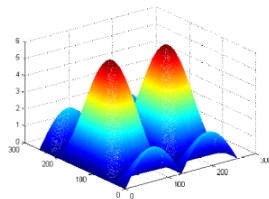
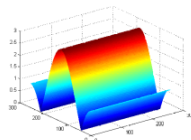
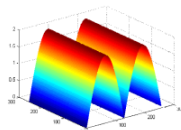
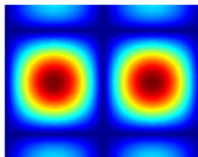
Frekvensrespons



⊙



||



Når er filtrering raskere i frekvensdomenet enn i billedomenet?

Anta bildestørrelse $M \times N$ og filterstørrelse $m \times n$

- Filtrering i billedomenet krever $M \times N \times (m \times n)$ multiplikasjoner og tilsvarende addisjoner

Når er filtrering raskere i frekvensdomenet enn i billedomenet?

Anta bildestørrelse $M \times N$ og filterstørrelse $m \times n$

- Filtrering i billedomenet krever $M \times N \times (m \times n)$ multiplikasjoner og tilsvarende addisjoner
- Filtrering i billedomenet:
 - FFT av bildet og filterkjerne: $2\mathcal{O}((M \times N) \log_2(M \times N))$

Når er filtrering raskere i frekvensdomenet enn i billedomenet?

Anta bildestørrelse $M \times N$ og filterstørrelse $m \times n$

- Filtrering i billedomenet krever $M \times N \times (m \times n)$ multiplikasjoner og tilsvarende addisjoner
- Filtrering i billedomenet:
 - FFT av bildet og filterkjerne: $2\mathcal{O}((M \times N) \log_2(M \times N))$
 - Multiplikasjon i frekvensdomenet: $M \times N$ multiplikasjoner

Når er filtrering raskere i frekvensdomenet enn i billedomenet?

Anta bildestørrelse $M \times N$ og filterstørrelse $m \times n$

- Filtrering i billedomenet krever $M \times N \times (m \times n)$ multiplikasjoner og tilsvarende addisjoner
- Filtrering i billedomenet:
 - FFT av bildet og filterkjerne: $2\mathcal{O}((M \times N) \log_2(M \times N))$
 - Multiplikasjon i frekvensdomenet: $M \times N$ multiplikasjoner
 - Inverstransformasjon av resultatet: $\mathcal{O}((M \times N) \log_2(M \times N))$

Når er filtrering raskere i frekvensdomenet enn i billedomenet?

Anta bildestørrelse $M \times N$ og filterstørrelse $m \times n$

- Filtrering i billedomenet krever $M \times N \times (m \times n)$ multiplikasjoner og tilsvarende addisjoner
- Filtrering i billedomenet:
 - FFT av bildet og filterkjerne: $2\mathcal{O}((M \times N) \log_2(M \times N))$
 - Multiplikasjon i frekvensdomenet: $M \times N$ multiplikasjoner
 - Inverstransformasjon av resultatet:
 $\mathcal{O}((M \times N) \log_2(M \times N))$
- Filtrering i frekvensdomenet er raskere når $(m \times n) \gg \log_2(M \times N)$

- Korrelasjon i billedomenet \Leftrightarrow Elementvis multiplikasjon i billedomenet med $F(u, v)$ kompleks konjugert,

$$\text{corr}(f, h) \Leftrightarrow F^* \odot H$$

F^* er kompleks konjugert til F

- Korrelasjon i billedomenet \Leftrightarrow Elementvis multiplikasjon i billedomenet med $F(u, v)$ kompleks konjugert,

$$\text{corr}(f, h) \Leftrightarrow F^* \odot H$$

F^* er kompleks konjugert til F

- Motsatt gjelder også:

$$f^* \odot h \Leftrightarrow \text{corr}(F, H)$$

- Korrelasjon i billedomenet \Leftrightarrow Elementvis multiplikasjon i billedomenet med $F(u, v)$ kompleks konjugert,

$$\text{corr}(f, h) \Leftrightarrow F^* \odot H$$

F^* er kompleks konjugert til F

- Motsatt gjelder også:

$$f^* \odot h \Leftrightarrow \text{corr}(F, H)$$

- Brukes til å f.eks finne hvor i bildet visse mønstre/objekter befinner seg

- Bildet ses på som periodisk
- Vi får “kunstige bidrag” langs aksene i spekteret grunnet diskontinuitet i bilderand

- Bildet ses på som periodisk
- Vi får “kunstige bidrag” langs aksene i spekteret grunnet diskontinuitet i bilderand
- Vindusfunksjon til å dempe kunstige bidrag i spekteret

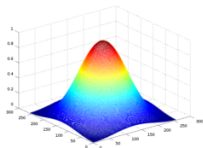
- Bildet ses på som periodisk
- Vi får “kunstige bidrag” langs aksene i spekteret grunnet diskontinuitet i bilderand
- Vindusfunksjon til å dempe kunstige bidrag i spekteret
 - Vindusfunksjonene gjør at pikselverdier går mot null langs bilderand

- Bildet ses på som periodisk
- Vi får “kunstige bidrag” langs aksene i spekteret grunnet diskontinuitet i bilderand
- Vindusfunksjon til å dempe kunstige bidrag i spekteret
 - Vindusfunksjonene gjør at pikselverdier går mot null langs bilderand
 - Gjøres ved: $f_w(x, y) = f(x, y)w(x, y)$

- Bildet ses på som periodisk
- Vi får “kunstige bidrag” langs aksene i spekteret grunnet diskontinuitet i bilderand
- Vindusfunksjon til å dempe kunstige bidrag i spekteret
 - Vindusfunksjonene gjør at pikselverdier går mot null langs bilderand
 - Gjøres ved: $f_w(x, y) = f(x, y)w(x, y)$
 - Ta DFT av $f_w(x, y)$

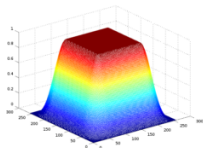
Vindusfunksjoner

Hamming-vindu:



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

Tukey-vindu:



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


- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- Bidragene langs aksene i Fourierspekteret reduseres

Effekten av vinduer

- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- Bidragene langs aksene i Fourierspekteret reduseres
- Vil også påvirke andre frekvenser i bildet!

Effekten av vinduer

- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- Bidragene langs aksene i Fourierspekteret reduseres
- Vil også påvirke andre frekvenser i bildet!
- Konvolusjonsteoremet: Multiplikasjon i billedomenet er en konvolusjon i frekvensdomenet

- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- Bidragene langs aksene i Fourierspekteret reduseres
- Vil også påvirke andre frekvenser i bildet!
- Konvolusjonsteoremet: Multiplikasjon i bildedomenet er en konvolusjon i frekvensdomenet
 - Multiplikasjon av en bred Gaussisk-form i bildedomenet er ekvivalent med en konvolusjon av en smal Gaussisk-form i frekvensdomenet

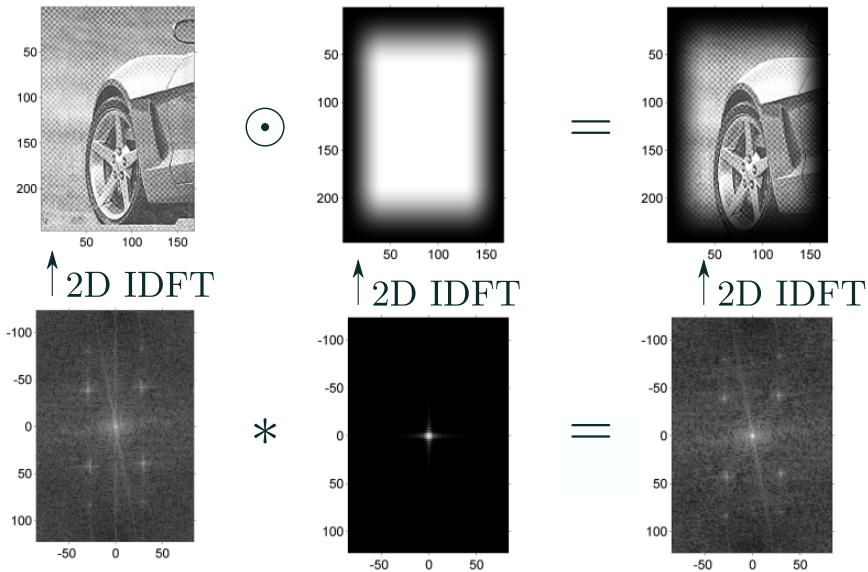
- For å redusere effekten fra bildekantene i spekteret:

$$f_w(x, y) = f(x, y)w(x, y)$$

FØR vi gjør FFT

- Bidragene langs aksene i Fourierspekteret reduseres
- Vil også påvirke andre frekvenser i bildet!
- Konvolusjonsteoremet: Multiplikasjon i bildedomenet er en konvolusjon i frekvensdomenet
 - Multiplikasjon av en bred Gaussisk-form i bildedomenet er ekvivalent med en konvolusjon av en smal Gaussisk-form i frekvensdomenet
 - Bruk av en vindusfunksjon gir en utsmøring/“blurring” av spekteret

Eksempel: bruk av vindusfunksjon



Vindusfunksjoner

- Finnes mange typer vindusfunksjoner

Vindusfunksjoner

- Finnes mange typer vindusfunksjoner
- Ofte defineres i 1D og utvides til 2D ved matrisemultiplikasjon

Vindusfunksjoner

- Finnes mange typer vindusfunksjoner
 - Ofte defineres i 1D og utvides til 2D ved matrisemultiplikasjon
- lager en 1D-variant h (kolonnevektor) og får 2D variant ved å regne ut hh^T

Vindusfunksjoner

- Finnes mange typer vindusfunksjoner
 - Ofte defineres i 1D og utvides til 2D ved matrisemultiplikasjon
- lager en 1D-variant h (kolonnevektor) og får 2D variant ved å regne ut hh^T
- Eksempel på forrige slide brukte *Tukey*-vinduet som i 1D er definert som

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

Vindusfunksjoner

- Finnes mange typer vindusfunksjoner
 - Ofte defineres i 1D og utvides til 2D ved matrisemultiplikasjon
- lager en 1D-variant h (kolonnevektor) og får 2D variant ved å regne ut hh^T
- Eksempel på forrige slide brukte *Tukey*-vinduet som i 1D er definert som

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

α regulerer skarpheten til overgangen

$\alpha = 0$ gir rektangulært vindu, $\alpha = 1$ gir glatt vindu som er et *Hann vindu*.

- Vindusfunksjoner kan også brukes i Fourierdomenet

- Vindusfunksjoner kan også brukes i Fourierdomenet
- Definere overgangene i et filter

- Vindusfunksjoner kan også brukes i Fourierdomenet
- Definere overgangene i et filter
- Butterworth og Gaussisk er eksempler på vindusfunksjoner

- Vindusfunksjoner kan også brukes i Fourierdomenet
- Definere overgangene i et filter
- Butterworth og Gaussisk er eksempler på vindusfunksjoner
- Vindusfunksjoner kan brukes i begge domener

Oppsummering

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt

Oppsummering

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet

Oppsummering

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder
 - Myke overganger \rightarrow redusere ringing

- Konvolusjonsteoremet: Sirkelkonvolusjon i billedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder
 - Myke overganger \rightarrow redusere ringing
 - Analyse av konvolusjonsfiltre gjennom frekvensrespons

- Konvolusjonsteoremet: Sirkelkonvolusjon i bildedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder
 - Myke overganger \rightarrow redusere ringing
 - Analyse av konvolusjonsfiltre gjennom frekvensrespons
 - Rask implementasjon av større konvolusjonsfiltre

- Konvolusjonsteoremet: Sirkelkonvolusjon i bildedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
- Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder
 - Myke overganger \rightarrow redusere ringing
 - Analyse av konvolusjonsfiltre gjennom frekvensrespons
 - Rask implementasjon av større konvolusjonsfiltre
- Vindusfunksjoner på bilder før transformasjonen

Oppsummering

- Konvolusjonsteoremet: Sirkelkonvolusjon i bildedomenet er ekvivalent med elementvis multiplikasjon i frekvensdomenet, og omvendt
 - Anvendelser:
 - Design av filtre i frekvensdomenet
 - Lavpass, høypass, båndpass, båndstopp, notch
 - I praksis: la filteret H være symmetrisk om nullfrekvens. Konjugert symmetri, $H(u, v) = H^*(-u, -v)$ gir reelle utbilder
 - Myke overganger \rightarrow redusere ringing
 - Analyse av konvolusjonsfiltre gjennom frekvensrespons
 - Rask implementasjon av større konvolusjonsfiltre
 - Vindusfunksjoner på bilder før transformasjonen
- \rightarrow Redusere bidrag langs aksene i frekvensspekteret