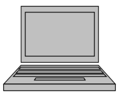
LAB A: Operationer på gråskalebilder, fortsättning...

Maria Magnusson
Avdelningen för Datorseende, Institutionen för Systemteknik,
Linköpings Universitet

februari 2016

1 Introduktion



Som förut innebär en datorsymbol att en fil ska skapas och skickas till läraren. Kopiera bilderna `math_logo.mat, peppersmall.mat` till ditt hembibliotek.

2 Fouriertransform

2.1 Displaying Fourier transforms in Matlab

The discrete Fourier transform (DFT) can be fast computed by the fast Fourier transform (FFT). The Matlab function `fft` assumes the origin of the data to be at the first position of the input vector. When dealing with images we usually want the origin of the data to be in the center instead. When the data is of even size, the central position can be located at the pixel slightly to the right of the geometrical center. The `ifftshift` shifts the origin in the signal domain from the middle position to the first position as shown in Figure 1. The result from `fft` has the Fourier origin, that is the DC-component, in the first position of the output vector. The `fftshift` command shifts it to the middle position where we expect to find it when we plot the function. Consequently, if f is the function in the signal domain, its Fourier transform is calculated as

```
F = fftshift(fft(ifftshift(f)));
```

and if F is the function in the Fourier domain, its inverse Fourier transform is calculated as

```
f = fftshift(ifft(ifftshift(F)));
```

The 2D case for even sized data is illustrated in Figure 2. Consequently, if f is the function in the spatial domain, its Fourier transform is calculated as

```
F = fftshift(fft2(ifftshift(f)));
```

and if F is the function in the Fourier domain, its inverse Fourier transform is calculated as

```
f = fftshift(ifft2(ifftshift(F)));
```

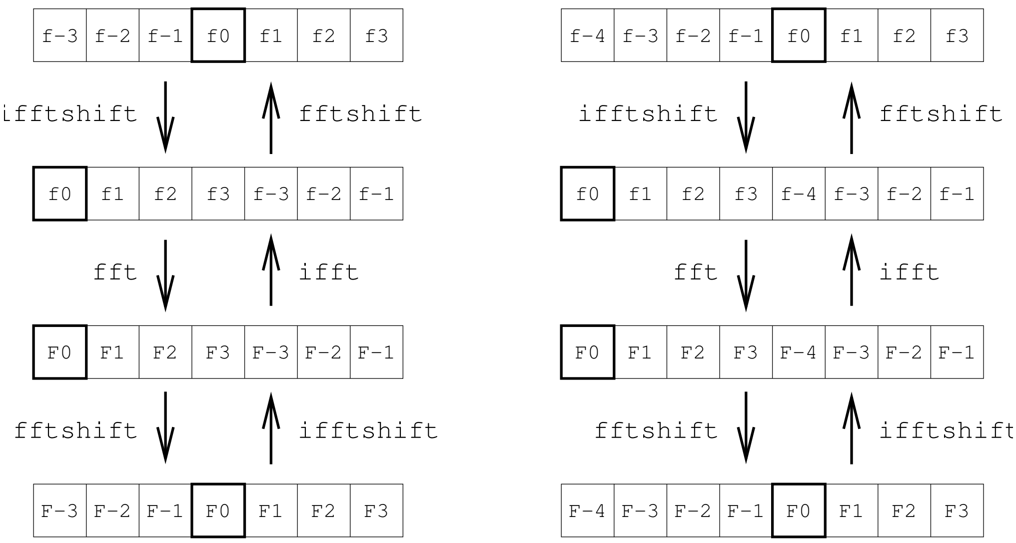


Figure 1: The functions **fftshift** and **ifftshift** are used to place the origin at the correct positions when transforming vectors. Left: odd number of samples. Right: even number of samples.

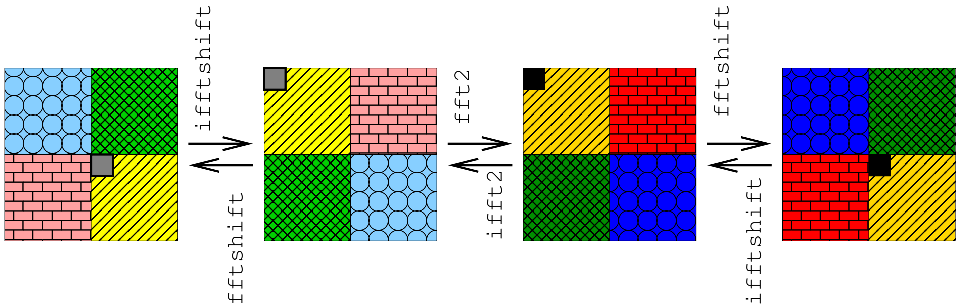


Figure 2: The functions **fftshift** and **ifftshift** are used to place the origin at the correct positions when transforming matrices. Illustrated for even number of samples.

2.2 Undersökning av fouriertransformen

Skapa en fil `FFTmathlogo.m` med nedanstående innehåll och exekvera den.

```
load math_logo
im = math_logo;
IM = fftshift(fft2(ifftshift(im)));
maxIM = max(max(abs(IM)));
shiftim = circshift(circshift(im,5,2),10,1);
figure(1); colormap gray;
subplot(121); imagesc(im);
axis image; title('im')
colorbar('SouthOutside')
subplot(122); imagesc(shiftim);
axis image; title('shifted im')
colorbar('SouthOutside')
figure(2); colormap gray;
subplot(221); imagesc(abs(IM), [0 0.02*maxIM]);
axis image; colorbar; title('abs(F[im])')
subplot(222); imagesc(angle(IM), [-pi pi]);
axis image; colorbar; title('angle(F[im])')
subplot(223); imagesc(real(IM), [-0.02*maxIM 0.02*maxIM]);
axis image; colorbar; title('real(F[im])')
subplot(224); imagesc(imag(IM), [-0.02*maxIM 0.02*maxIM]);
axis image; colorbar; title('imag(F[im])')
```

FRÅGA 1: I `figure(2)` tittar vi på fouriertransformen $F(u, v)$ av en bild $f(x, y)$ på olika sätt. Det gäller att fouriertransformen av en reell bild är hermitisk, dvs $F(u, v) = F^*(-u, -v)$, där $*$ noterar komplex-konjugat. Utgående från denna formel kan man visa att olika typer av symmetri gäller. Man talar om udda och jämna funktioner (se kompendiet). Vilka typer av symmetri gäller för bilderna i `figure(2)`? Ge både en liten formel och nämn hur det yttrar sig i respektive figur.

FRÅGA 2: Hur lyder translationsteoremet i 2D? (Leta upp det i tabell 3.1 i kompendiet.)

Bilden `shiftim` till höger i `figure(1)` är skiftad jämfört med bilden `im` till vänster. Lägg till kod i `FFTmathlogo.m` där fouriertransformen av `shiftim` beräknas och visas i `figure(3)`.

FRÅGA 3: Vilken av fouriertransformbilderna `abs( )`, `angle( )`, `real( )` eller `imag( )` förändras inte av translation? Är detta i enlighet med translationsteoremet?

FRÅGA 4: Visa `abs(IM)` i `figure(2)` utan att specificera min och maxvärden, dvs ta bort `[0.02*maxIM]`. Då kan man se endast en frekvenskomponent. Vilken?

FRÅGA 5: Normalt sett dominerar denna frekvenskomponent kraftigt för vanliga bilder. Den är lika med summan av pixelvärdena i bilden. Kontrollera detta både för bilden (med `sum`) och för fouriertransformen (klicka) och ge värdet nedan.

2.3 Bild/fouriertransform vid faltning/filtrering

Filterkärnan `aver` av storleken 3×3 kan implementeras i MATLAB. 3 st ihopfaltade `aver` ger sedan filterkärnan `aver3`.

```
aver = [1 2 1; 2 4 2; 1 2 1] /16;
aver3 = conv2(conv2(aver,aver,'full'),aver,'full');
```

FRÅGA 6: Vilken storlek kommer `aver3` att ha? Kontrollera ditt svar genom att utföra ovanstående kommandon i MATLAB.

Skapa en fil `FFTmathlogo2.m` med nedanstående innehåll och exekvera den.

```
load math_logo
im = math_logo;
aver = [1 2 1; 2 4 2; 1 2 1]/16;
aver3 = conv2(conv2(aver,aver,'full'),aver,'full');
imconv = conv2(im,aver3,'same');
aver3im = 0*im;
center = size(im,1)/2+1;
aver3im(center-3:center+3, center-3:center+3) = aver3;
IM      = fftshift(fft2(ifftshift(im)));
AVER3IM = fftshift(fft2(ifftshift(aver3im)));
IMCONV  = fftshift(fft2(ifftshift(imconv)));
maxIM   = max(max(abs(IM)));
figure(1); colormap gray;
subplot(131); imagesc(im);
axis image; title('im'); colorbar('SouthOutside')
subplot(132); imagesc(aver3im);
axis image; title('aver3im'); colorbar('SouthOutside')
subplot(133); imagesc(imconv);
axis image; title('imconv'); colorbar('SouthOutside')
figure(2); colormap gray;
subplot(131); imagesc(abs(IM), [0 0.01*maxIM]);
axis image; title('abs(F[im])'); colorbar('SouthOutside')
subplot(132); imagesc(abs(AVER3IM));
axis image; title('abs(F[aver3im])'); colorbar('SouthOutside')
subplot(133); imagesc(abs(IMCONV), [0 0.01*maxIM]);
axis image; title('abs(F[imconv])'); colorbar('SouthOutside')
```

I `figure(1)` ser vi originalbilden `im`, en bild av faltningskärnan `aver3im`, samt faltningsresultatet `imconv`. I `figure(2)` ser vi deras respektive fouriertransformer.

FRÅGA 7: Zooma i bilden `aver3im` och kontrollera att du ser den lilla faltningskärnan. Klicka i bilden och kontrollera att centrumvärdet är korrekt. Vilket är det korrekta värdet?

FRÅGA 8: Hur lyder faltningsteoremet i 2D? (Leta upp det i tabell 3.1 i kompendiet.)

FRÅGA 9: Studera fouriertransformerna i `figure(2)` och beskriv hur man kan se att de uppfyller faltningsteoremet. (Tips: Tala om multiplikation i din beskrivning.)

Vi ska nu relatera bildegenskaper till fouriertransformen. Nollfrekvensen $(u, v) = (0, 0)$ är belägen mitt i fouriertransformbilden på koordinat (62,62). Den har ett stort värde. Det konstaterade vi tidigare. Länge ut mot kanterna på fouriertransformbilden blir frekvenserna högre ty $\sqrt{u^2 + v^2}$ blir större.

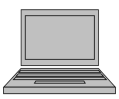
FRÅGA 10: Fyll nu i orden *låga frekvenser* eller *höga frekvenser* i boxarna i texten nedan! Tänk på att bilderna `im`, `imconv` och fouriertransformerna `IM`, `IMCONV` ska stämma med din text.

En vanlig bild domineras oftast av jämna ytor med olika kontrast. Detta svarar mot $\boxed{\qquad\qquad \textit{frekvenser}}$ *i fouriertransformbilden.*

Bilden `im` *har både jämna ytor och skarpa kanter. De skarpa kanterna svarar mot* $\boxed{\qquad\qquad \textit{frekvenser}}$ *i fouriertransformbilden.*

Bilden `imconv` *har jämna ytor och ganska suddiga, mjuka kanter. Därmed är de* $\boxed{\qquad\qquad \textit{frekvenserna}}$ *undertryckta i fouriertransformbilden.*

> **Slutsats:** Vi har alltså faltat bilden `im` med ett medelvärdesbildande filter och erhållit `aver3im`. Alternativt säger man att `im` lågpassfiltreras till `aver3im`. De låga frekvenserna bevaras medan de höga frekvenserna dämpas.



Kopiera filen `FFTmathlogo2.m` till `FFTpeppers.m` men byt bild till `peppersmall` istället för `math_logo`.

FRÅGA 11: Varför får bilden `imconv` en mörk ram? (Tobias, jag vet att du svarade på detta förut...) Den mörka ramen verkar yttra sig som något i fouriertransformbilden. Välj mellan stjärna/kors/ruta/prick!

Det finns olika sätt att motverka den mörka ramen. Man kan t ex extrapolera värden utanför bilden innan faltning. Man kan också utföra faltningen via multiplikation i fourierdomänen, så som det illustreras i Fig. 3.9 i kompendiet. Implementera detta i `FFTpeppers2.m`. Du har redan kommit en bit på vägen i `FFTpeppers.m`.

FRÅGA 12: Jämför din nya bild med `imconv` i `FFTpeppers.m`. Beskriv skillnader och likheter.

I det här läget kanske man blir sugen på att utföra en ideal lågpassfiltrering. Kopiera `FFTpeppers2.m` till `FFTpeppers3.m`. Byt ut `AVER3IM` mot ett idealt cirkulärt lågpassfilter som kan skapas enligt:

```
siz = size(im,1);
[u,v] = meshgrid(-siz/2:siz/2-1,-siz/2:siz/2-1);
IDEALFILT = sqrt(u.^2+v.^2) < myradius;
```

Sätt ett lämpligt värde på `myradius` så att `IDEALFILT` blir så likt `AVER3IM` som möjligt.

FRÅGA 13: Kontrollera så att de lågpassfiltrerade bilderna `imconv` och din nya bild ser ungefär lika suddiga ut. Din nya bild har dock en typ av störning som är den två-dimensionella motsvarigheten till Gibb's fenomen. Beskriv hur störningen yttrar sig i bilden!

Slutsats: Ideal filtrering är inte bra. Man får störningar av typen Gibb's fenomen.