Frekans domain’inde İşlemler

BMÜ-357 Sayısal Görüntü İşleme

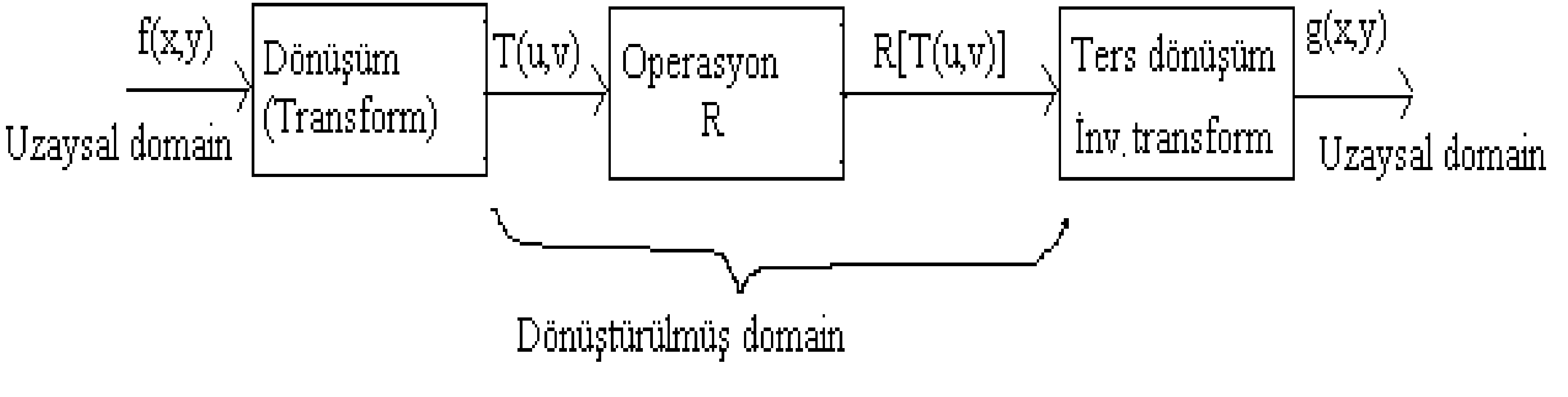
Yrd. Doç. Dr. İlhan AYDIN

Domain Dönüşümü

Dönüşüm, bir sinyalin, başka parametrelerle ifade edilmesi şeklinde düşünülebilir.

Ters dönüşüm ise , sinyalin ilk halindeki parametrelerle ifade edilebilir şekle geri döndürülme işlemidir.

Fourier transformasyonları (dönüştürücüleri) bir sinyalin frekans domanine dönüşümünü sağlar.



Frekans uzayı

- İmge uzayında yapılabilecek işlemlerin yanında, frekans düzlemindeki bilgi de imge işlemede sıkça kullanılmaktadır.
- Daha önce imge süzgeçleme için konvülüsyondan bahsedilmişti. İmge uzayında yapılan bu işlem her bir piksel için tekrarlanmakla birlikte, çekirdek elemanına bağlı olarak hesapsal yükü oldukça fazla olabilmektedir.
- Frekans uzayına geçildiğinde konvülüsyon işlemi çarpma işlemine dönüşeceğinden, bu uzayda yapılacak süzgeçleme işlemlerinde frekans uzayına geçiş ve geri dönüş işlemleri için hesapsal yükten bahsedilebilir.
- Ayrıca frekans uzayında imgedeki piksellerin dağılımına ilişkin bilgileri gözlemlemek de mümkündür.
- Frekans uzayına geçiş için genellikle Fourier dönüşümü kullanılmaktadır.

Fourier Dönüşümü

- Bu dönüşüm, görüntü işlemenin çok önemli konularından biridir. Uzaysal domainde başarılması zor işlemleri, frekans domain'inde başaracak yapıda olan bu dönüşüm,

  "görüntüyü oluşturan frekans bileşenlerini biribirinden ayırt edebildiği için değişik derecelerden alçak ve yüksek geçiren filtreleme işlemleri"

  kolaylıkla başarılabilir.
- Önce tek boyutlu sonra iki boyutlu Fourier dönüşümünü kısaca hatırlayalım;

Temel Bilgi notu(Fourier analizi)

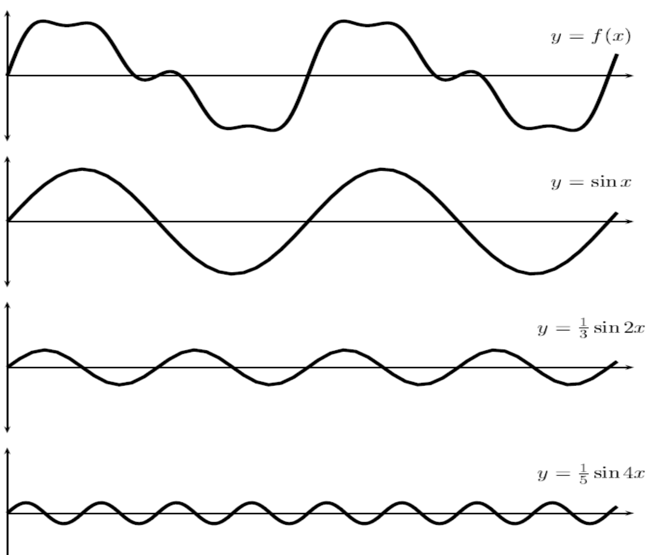
Periodik ve sonlu değer alabilen her fonksiyon, değişik frekanslarda titreşen sinüs veya cosinüslü bileşenlerin toplamından oluşur.

$$f(x) = f(x+T) \quad (-\infty < x < \infty)$$

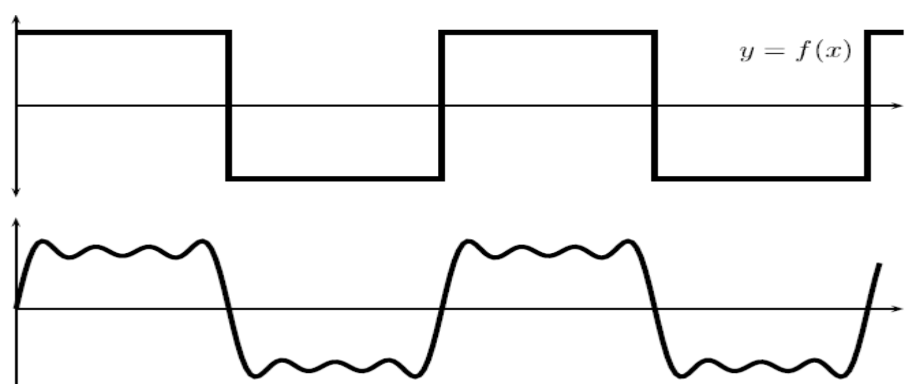
$$f(x) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} a_k \cos(\omega k x) + b_k \sin(\omega k x) \qquad (\omega = \frac{2\pi}{T})$$

$$a_k = \frac{2}{T}\int_0^T f(x)\cos(\omega k x) \quad dx$$

$$b_k = \frac{2}{T}\int^T f(x)\sin(\omega k x)dx$$



$f(x) = \sin x + \frac{1}{3}\sin 2x + \frac{1}{5}\sin 4x$



$f(x) = \sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x$

Tek boyutlu ayrık Fourier dönüşümü (DFT)

Herhangi bir fonksiyon ayrıklaştırıldığında sonlu sayıda elemanlı bir dizi şeklinde ifade edilir. Örneğn bir kare dalganın ayrık zamanlı dizi şeklinde ifadesi;

1 1 1 1 -1 -1 -1 -1 şeklinde olabilir.

$$\mathbf{f} = [f_0, f_1, f_2, \ldots, f_{N-1}]$$

Eleman sayısı N olan bir ayrık f fonksiyonun, ayrık Fourier katsayıları dizisi aşağıda tanımlansın.

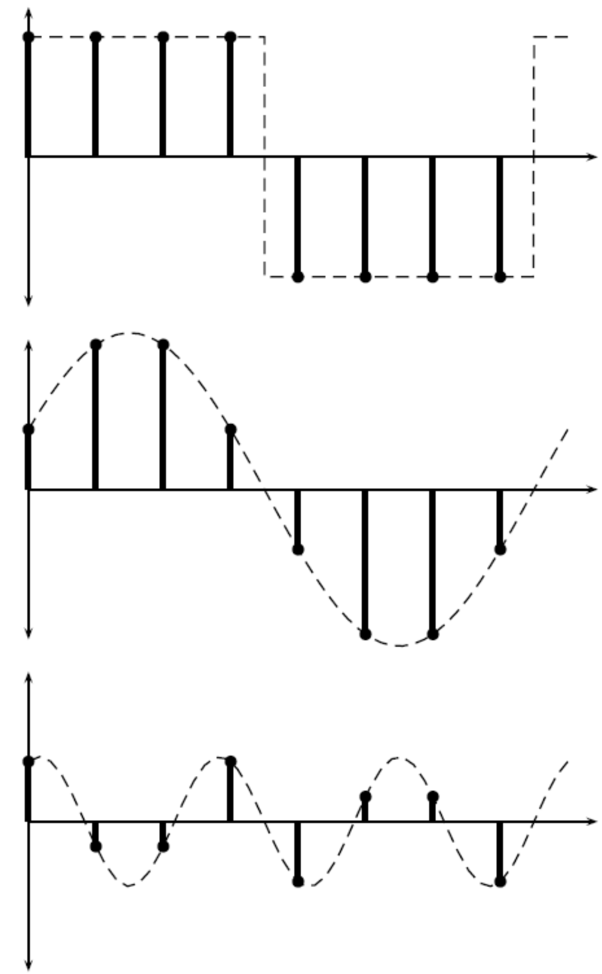
$$\mathbf{F} = [F_0, F_1, F_2, \ldots, F_{N-1}]$$

Burada; u.ayrık fourier bileşninin katsayısı aşağıdaki bağıntıyla hesaplanır (DFT).

$$F_u = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot \exp\left[-2\pi i \frac{xu}{N}\right]$$

Aynı şekilde, fourier katsayılarından dizi elemanını elde etmek için ters foürier dönüşümü yapılır (IDFT).

$$f(x) = \sum_{u=0}^{N-1} F(u) \cdot \exp\left[2\pi i \frac{xu}{N}\right]$$



Frekans Uzayı – Ayrık Fourier Dönüşümü

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

$$u = 0, 1, 2, \ldots, M-1$$

- $F(u)$ yu bulmak için:
  - u=0 için x'in bütün değerlerinde yukarıdaki toplamı hesapla,
  - u=1 için x'in bütün değerlerinde yukarıdaki toplamı hesapla,
  -
  -
  -
  - u=M için x'in bütün değerlerinde yukarıdaki toplamı hesapla.

Frekans Uzayı – Ayrık Fourier Dönüşümü

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

$$u = 0,1,2,\ldots,M-1$$

- $F(u)$ yu bulmak için:
  - Bu işlem için M^2 çarpma ve toplama gerekli.

- Euler teoremine göre: $e^{j\theta} = Cos\theta + jSin\theta$

$$e^{-j\theta} = Cos\theta - jSin\theta$$

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\left[Cos(2\pi ux/M) - jSin(2\pi ux/M)\right]$$

$$u = 0,1,2,\ldots,M-1$$

Frekans Uzayı – Ayrık Fourier Dönüşümü

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\left[Cos\left(2\pi ux / M\right) - jSin\left(2\pi ux / M\right)\right]$$

$$u = 0,1,2,\ldots,M-1$$

- Özetle $f(x)$, farklı frekanslardaki Sin ve Cos bileşenleri ile çarpılıyor.

$$\left.\begin{array}{l} F(0) \\ F(1) \\ . \\ . \\ . \\ F(M) \end{array}\right\} \text{Dönüşümün frekans bileşenleri}$$

$$F(u) = \left|F(u)\right| e^{-j\varphi(u)}$$

$$\left|F(u)\right| = \left[R^2(u) + I^2(u)\right]^{1/2} \qquad \text{Genlik}$$

$$\varphi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right] \qquad \text{Faz açısı}$$

Frekans Uzayı – Ayrık Fourier Dönüşümü

Örnek: f = [1, 2, 3, 4] ve N=4 için

$$\begin{aligned}\omega &= \exp\left[\frac{-2\pi i}{4}\right]\\ &= \exp\left[-\frac{\pi i}{2}\right]\\ &= \cos\left(-\frac{\pi}{2}\right) + i\sin\left(-\frac{\pi}{2}\right)\\ &= -i.\end{aligned}$$

$$\mathcal{F} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & (-i)^2 & (-i)^4 & (-i)^6 \\ 1 & (-i)^3 & (-i)^6 & (-i)^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

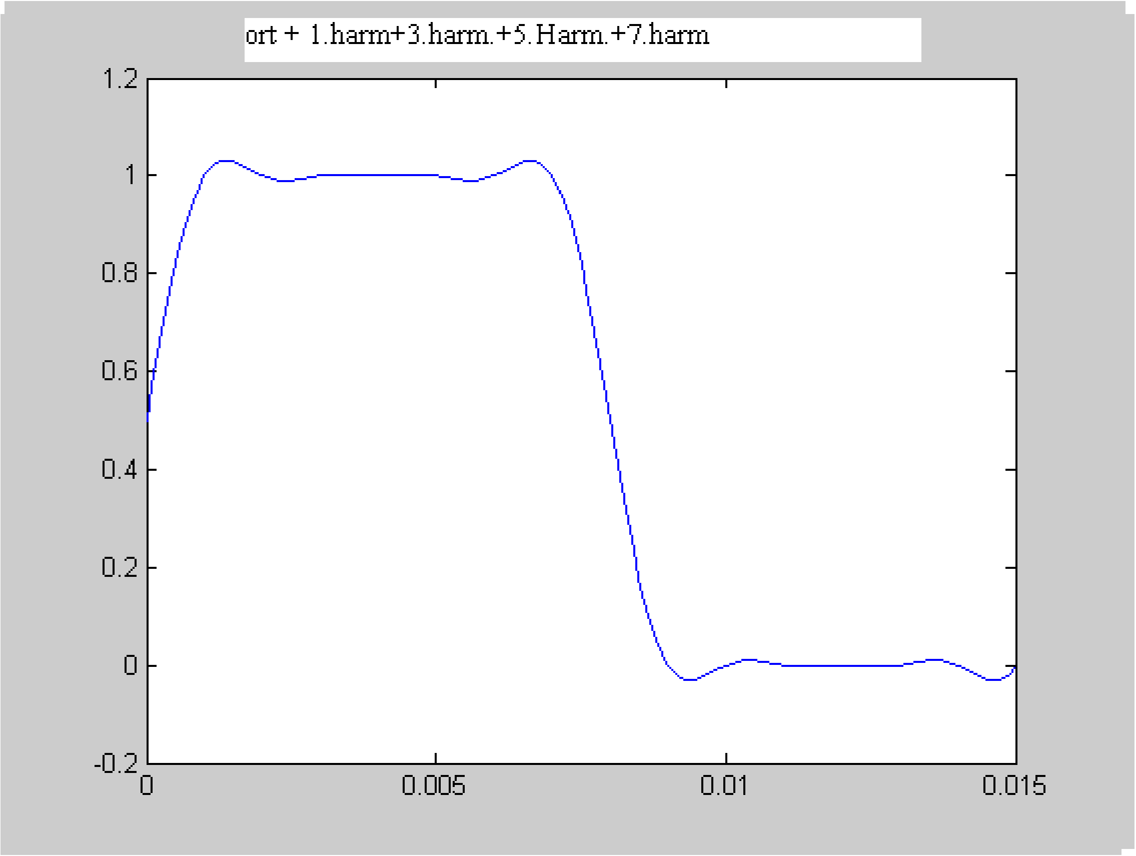
$$\mathbf{F} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 10 \\ -2+2i \\ -2 \\ -2-2i \end{bmatrix}$$

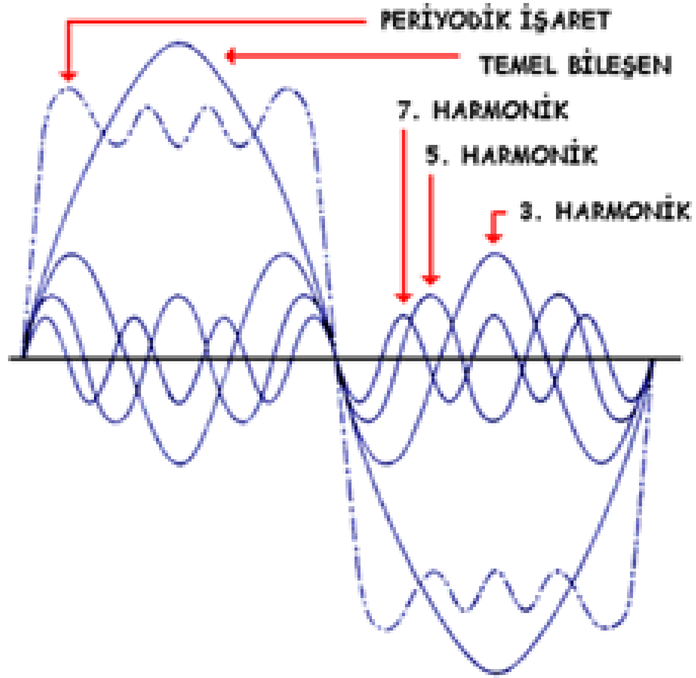
- FFT aslında bir matris çarpımı olarak gösterilebilir.
- $F = \mathcal{F} f$
- Bu denklemde $\mathcal{F}$ NxN boyutunda bir matris olup
- $\mathcal{F} = \frac{1}{N}\exp\left[-2\pi i\frac{mn}{N}\right]$
- Denklemde $\omega = \exp\left[\frac{-2\pi i}{N}\right]$ alınırsa
- $\mathcal{F}_{m,n} = \frac{1}{N}\omega^{mn}$ *olur*.

$$\mathcal{F} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & \cdots & \omega^{3(N-1)} \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & \cdots & \omega^{4(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \omega^{4(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$$

Tek boyutlu DFT işlemiyle ilgili….

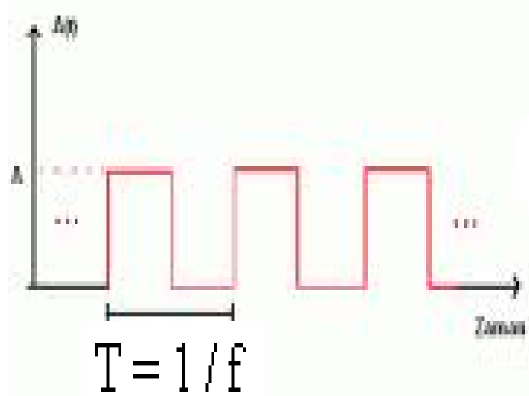
$F_{(n)} = (1,1,1,1,0,0,0,0)$

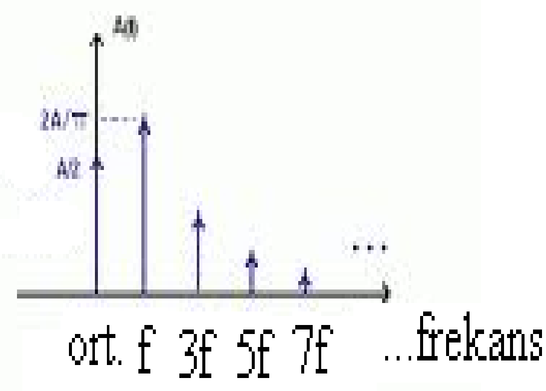




Zaman üzerinde kare şeklinde değişen sinyal

Bu kare dalganın Harmonik bileşen katsayıları (Max.genlik) değerleri)





Fast Fourier Dönüşümü

- DFT, ayrık fonksiyonların fourier dönüşümünü yapmak için çok önemli bir yapı olmakla beraber, dizi eleman sayısının çok fazla olduğu dönüşümler için uzun zaman harcamaktadır. Özellikle resim işleme gibi data sayısı fazla olan işlemler için yavaştır.
- Bunun yerine FFT(Hızlı fourier dönüşümü) ile bileşen katsayıları bulunur. Burada dikkat edilecek konunun dizinin 2’nin kuvveleri sayısında elemandan oluşturulmasıdır.

2-D (2 Boyutlu) DFT

2 boyutlu DFT; Matris şeklinde bir diziyi giriş kabul edip, yine matris şeklinde bir çıkış elde eder. Farz edelim ki;

$f(x, y)$, $x = 0, 1, 2, . . ., M − 1$ ve $y = 0, 1, 2, . . ., N − 1$,

$M \times N$ boyutunda bir görüntü olsun. Bu f görüntüsünün 2-D DFT'sindeki F(u,v) katsayıları aşağıdaki formülden hesaplanır. F'nin Fourier dönüşümü;

$F = \mathcal{F}(f)$

Ters fourier dönüşümü ise;

$f = \mathcal{F}^{-1}(F)$

2-D DFT

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

for $u = 0, 1, 2, \ldots, M-1$ and $v = 0, 1, 2, \ldots, N-1.$

2-D IDFT

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

for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1.$

2-D DFT

Eğer; f(x, y) dizisi reel ise, bunun fourier dönüşümü genellikle kompleks sayılardır. Dönüştürülmüş bir resmi frekans uzayında görmek (görsel analizi) onun spetrumunu ifade etmektir.

R(u, v) ve I(u, v) sırasıyla; F(u, v)'nin reel ve imajiner kısmını gösteriyorsa; Fourier ve güç spektrumu aşağıdaki gibi tarif edilir. Artık frekans uzayında bir görüntüyü, fourier spektrumu veya güç spektrumu ile ifade edebiliriz.

F(u,v) nin genliği

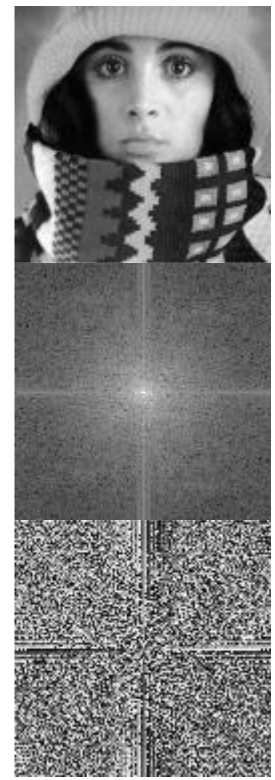
$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

F(U,v) nin faz açısı

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

Güç Spektrumu

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$



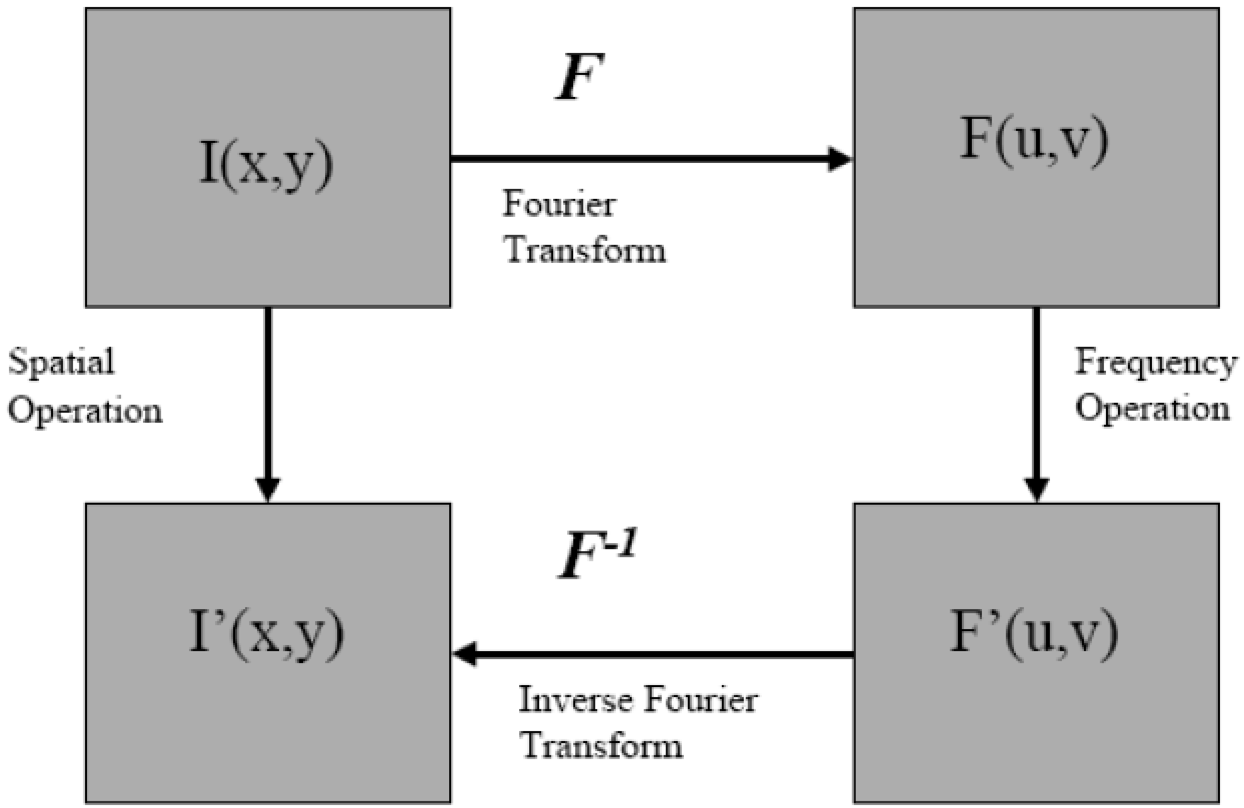
Orijinal

Genlik spk.

Faz spk.

Görüntünün tekrar orijinaline dönüştürülmesi için, hem genlik hem de faz bilgisine ihtiyaç vardır.

Frekans Uzayı – İmge İşleme Aşamaları



$$F'(u,v) = F(u,v)\,H(u,v)$$

H : Süzgeç

Eğer f(x, y) reel ise, onun fourier transformasyonu orijine göre eşlenik (Conjugate) simetriktir. Yani

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

Dolayısıyla aşağıdaki bağıntıda orijine göre simetriktir.

$$|F(u, v)| = |F(-u, -v)|$$

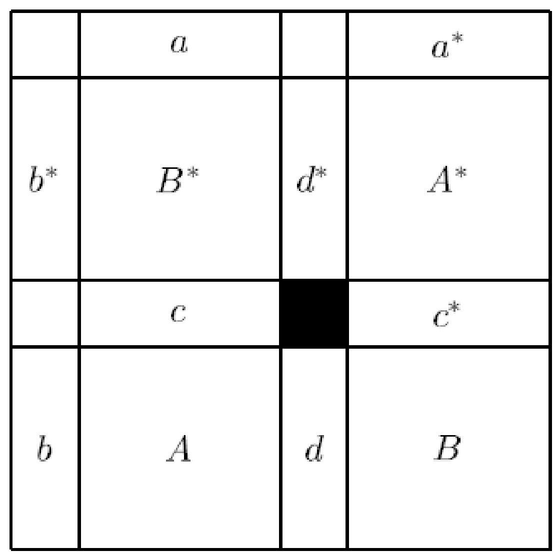


Figure 4.7: Conjugate symmetry in the DFT

- $F(u, v)$'nin aşağıdaki şekilde de yazılabileceğini söyleyebiliriz.

$$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$$

- Başka bir deyişle, DFT hem u yönünde hem v yönünde periyodiktir. Bu periyodiklik M ve N değişkenleri tarafından belirlenmiştir.
- Periodiklik IDFT'de vardır.

$$f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$$

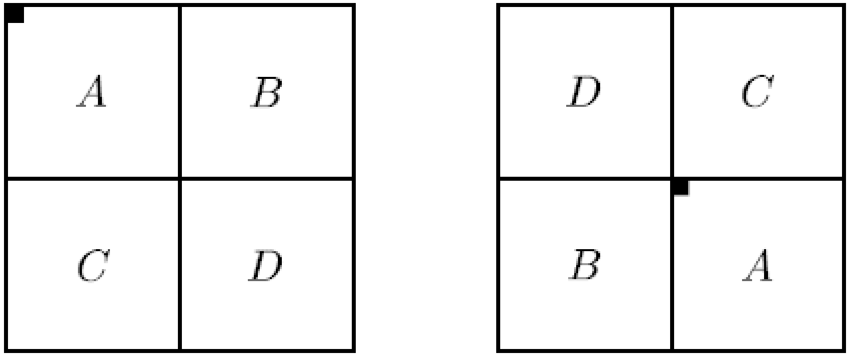
DC katsayısı ve kaydırma (shifting)

***DFT'deki DC katsayısı F(0,0) değeridir.* Aşağıdaki denklemde v=u=0 konulduğunda bulunur. Buna göre Dc katsayısının değeri orijinal görüntü matrisindeki tüm eleman değerlerinin toplamı olarak ortaya çıkar. Bu değer yeni matrisin en üst sol köşesindedir.**

Kaydırma: Görüntüleme amacı için; DC katsayısının matrisin orta nokta elemanı olarak oluşması uygundur. Bunun için kaydırma yapılabilir.

Kaydırma işlemi , dönüşümden önce f(x,y) matrisinin tüm elemanlarının $(-1)^{x+y}$ ile çarpılacağı anlamındadır.

$$F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y).$$



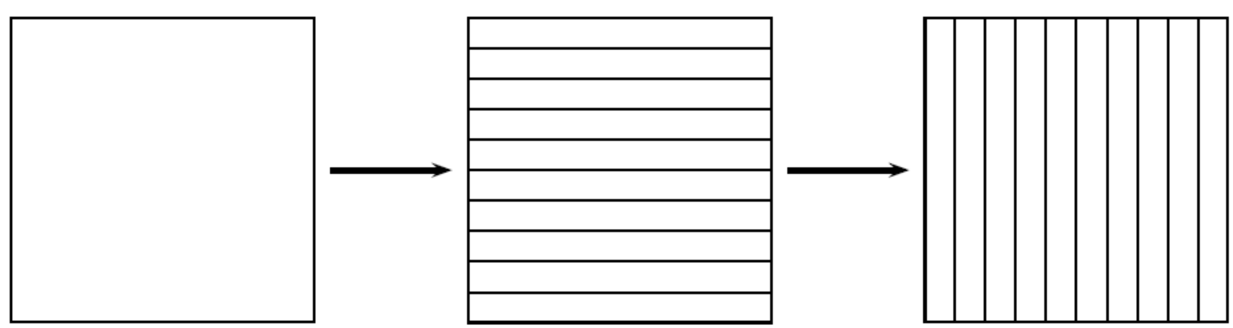
An FFT

After shifting

2-D DFT'nin bazı özellikleri

F ve f'yi hesaplayan denklemlerdeki üstel ifadeleri yandaki gibi ayrıştırabiliriz. Bunun anlamı x ve u 'ya ve y ve v 'ye göre ayrı ayrı işlem yapabiliriz. Yani iki boyutlu DFT tek boyutlu 2 adet DFT işleminden oluşmuştur. Bunuda resimdeki satırların ve sutunların ayrı ayrı DFT'sinin alınıp biribileriyle ilişkilendirilmesi şeklinde düşünebiliriz.

$$\exp\left[2\pi j\left(\frac{xu}{M}+\frac{yv}{N}\right)\right] = \exp\left[2\pi j\,\frac{xu}{M}\right]\exp\left[2\pi j\,\frac{yv}{N}\right]$$



(a) Original image (b) DFT of each row of (a) (c) DFT of each column of (b)

Dönüşümün gösterilmesi

- F(x,y) görüntüsünden elde edilen F(u,v) fourier katsayılar matrisinin elemanları kompleks sayılardır. Kompleks sayılar direkt olarak görüntülenemez.
- Onların magnitüdleri (genlikleri) **|F(u,v)|**alınır . Bu bir takım double sınıfı büyük sayılardır. Bu büyük sahada uğraşmak için;
- 1-) |F(u,v)|'deki en büyük değer **m** bulunur (**Bu DC değerdir**). Ve
  |F(u,v)|/m işlemi yapılır. Böylece imshow ile görüntülenebilir.
- 2-) |F(u,v)| 'yi görmek için mat2gray fonksiyonunu doğrudan kullanabiliriz.
- Genelde DC değerler çok büyük değerler olacağından; Bunun oluşturacağı görüntüde bu değer baskın çıkar. Bunun önüne geçmek için **|F(u,v)|** 'nın logaritmasının alınması daha uygundur.

  log(1+ |F(u,v)|)
- Fourier transformasyonun magnitude'unun (genliğinin) görüntülenmesi, transformasyonun spektrumu diye isimlendirilir.

MATLAB’da Fourier Dönüşümü

fft :Tek boyutlu DFT yapar(çıkışı vektördür)

ifft : DFT vektörünün tersini alır.

fft2: 2 boyutlu DFT yapar. Çıkışı Matristir.

İfft2: DFT matrixinin tersini alır.

fftshift: Bir tranformasyonu kaydırır. (örnek DC değerini merkeze alma gibi)

MATLAB’da Fourier Dönüşümü Örnekler

1- Eleman değerleri 1 olan bir f(x,y) matrisinin DFT’sini alalım.

>> a=ones(8);

>> fft2(a);

fft2 ile elde edilen matrisi yorumlayın?

```
>> a=ones(8)

a =

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

>> fft2(a)

ans =

    64     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
```

Örnek 2

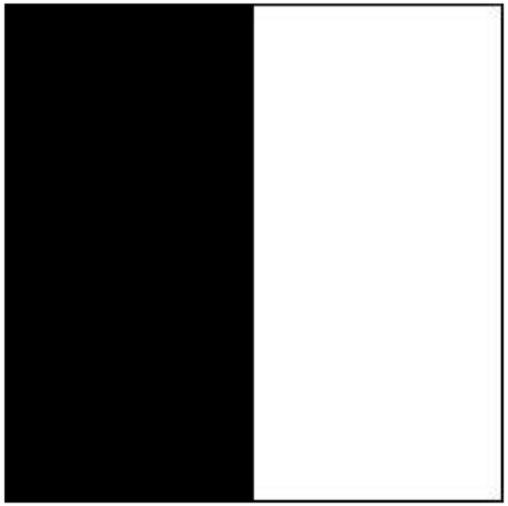
**** f(256,256) bir görüntü oluşturup, fourieir transformasyonunu yapınız.

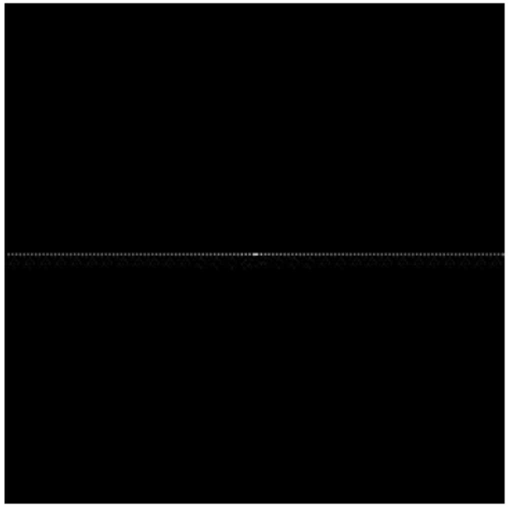
```
>> a=[zeros(256,128) ones(256,128)];

>> imshow(a)

>> af = fftshift (fft2(a));

>> imshow (mat2gray (log(1+abs(af))))
```

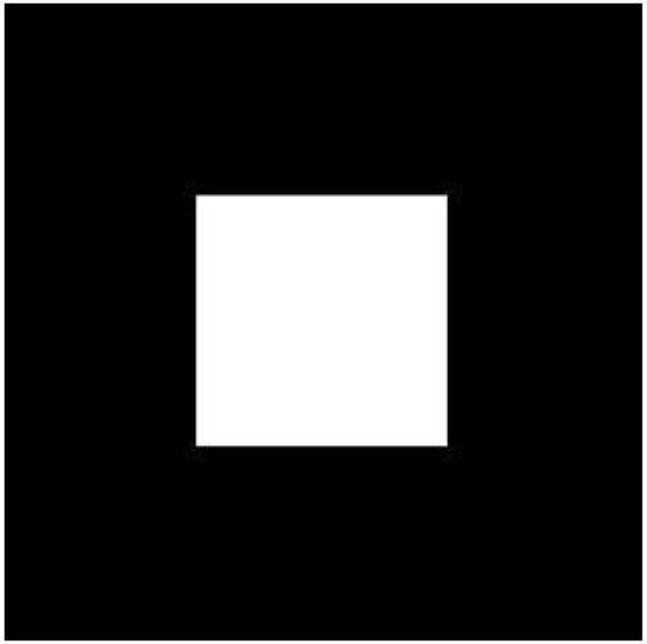


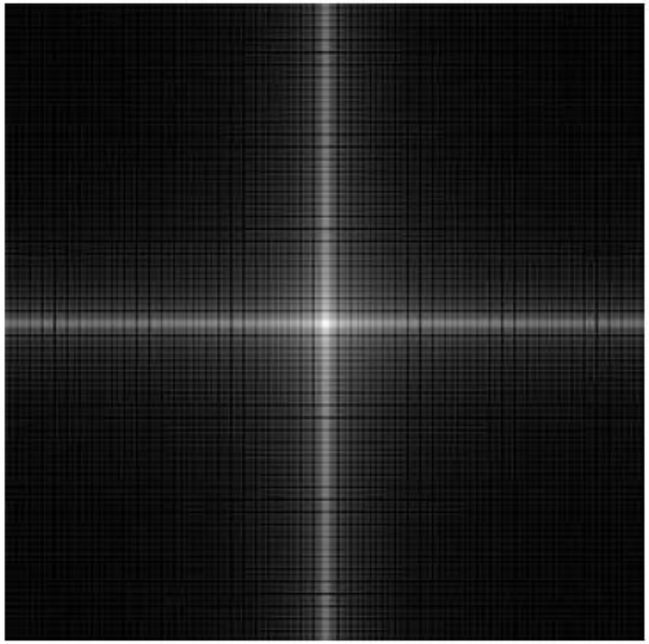


örnek

İç içe 2 dikdötrtgen (siyah içinde beyaz) yaratıp bunun fourier dönüşümünü alalım.

```
>> a=zeros(256,256);
>> a(78:178, 78:178)=1;
>> imshow(a)
>> af=fftshift(fft2(a));
>> imshow(mat2gray(log(1+abs(af))))
```





Frekans Uzayı – Matlab'da Frekans Uzayı İşlemleri

- MATLAB'da frekans uzayı filtreleme için:
  1. İmgenin AFD'sini al (**fft2** fonksiyonu).
  2. Karmaşık imgeye kaydırma işlemi uygula (**fftshift** fonksiyonu).
  3. Kaydırılmış *F(u,v)*'yi *H(u,v)* ile noktasal çarp.
  4. Karmaşık imgeye tekrar kaydırma işlemi uygula (**fftshift**).
  5. Karmaşık imgenin ters AFD'sini al (**ifft2** fonksiyonu).
  6. Elde edilen sonucun gerçek kısmını süzgeçlenmiş imge olarak kullan (**real** fonksiyonu).

Frekans Domain’de FİLTRELEME

Fourier dönüşümü bir domain dönüşüm işlemiydi. Filtreleme de bir filtre matrisinin görüntü matrisi ile konvolusyonu olduğundan Frekans domeninde filtreleme için fourier dönüşümü çok uygundur.

Filtre matrisi yaratmanın en çok kullanılan yolu ise; Matlab kodu aşağıda verilen yapıdır. Burada 1.komut ile 256x256lık matris oluşturulur (Bunların eleman değerleri -128 den başlayıp 1’er artarak 127 ye kadar değer alır.

Burda z herbir x,y koordinatının genlik değeridir.

```
>> [x,y]=meshgrid(-128:217,-128:127);
>> z=sqrt(x.^2+y.^2);
>> c=(z<15) ; Alçak geçiren Filtreye örnek
>> c=(z>15) ;Yüksek geçiren Filtreye örnek
```

İdeal Alçak Geçiren Filtre

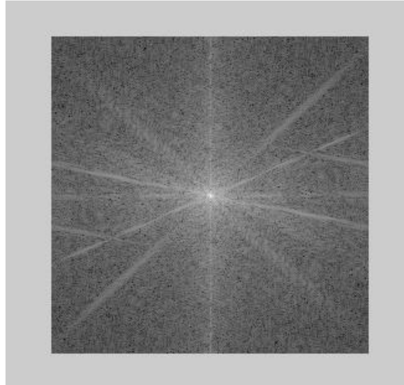
Varsayılım ki bir görüntünün Fourier dönüşüm matrisi F elimizde mevcut olsun. Ve bunun DC katsayısını merkeze kaydırdığımızda, dolayısıyla düşük frekanslı bileşen katsayıları da merkeze kaymış olacaktır.

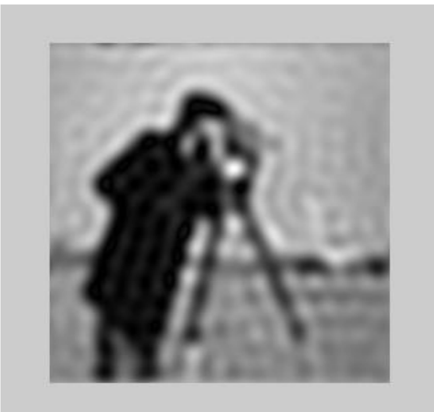
Bu durumda bir alçak geçiren filtre matrisi (c) ile bu F dönüşüm matrisini uygun bir şekilde işleme tabi tutarak (elementer çarpım işlemi .* (F.*c)), merkezdeki ve yakınındaki (alçak frekanslı bileşenler) değerler korunur ve merkezden uzak değerler (yüksek frekanslı bileşenler) ya küçülür veya yok olur. Böylece alçak geçiren filtre işlemi gerçekleşmiş olur.

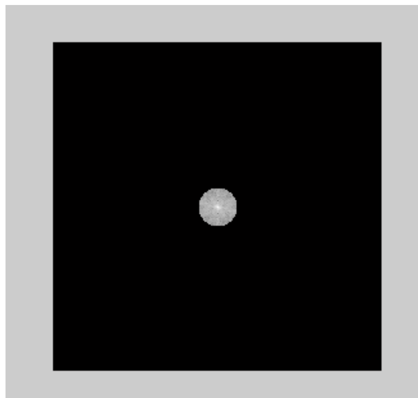
$$c(x,y) = \begin{cases} 1 & \text{Eğer } (x,y)\text{, merkeze } \mathbf{D} \text{ den daha yakın ise} \\ 0 & \text{Eğer } (x,y)\text{, merkeze } \mathbf{D} \text{ den daha uzakta ise} \end{cases}$$

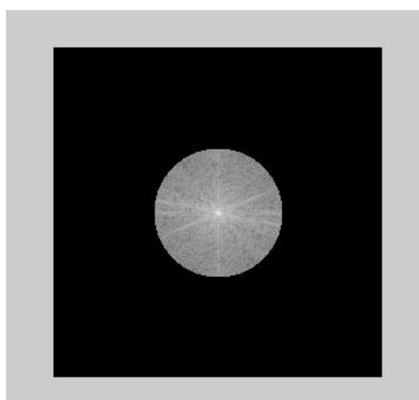
Bu filtrelenmiş FFT (DFT) matrisin inversi $F^{-1}(F.c)$ alınarak uzaysal domaine dönülmüş olur.

Filtrenin yarıçapı ne kadar küçükse elenen yüksek frekanslı bileşenler o kadar büyük olur. Resim bulanıklaşır









Frekans Domeninde ALÇAK GEÇİREN FİLTRE

```
>> cm=imread('cameraman.tif');
>> imshow(cm)
>> cf=fftshift(fft2(cm));
>> imshow(mat2gray(log(1+abs(cf))))
```

Buradan merkeze kaydırılmış transform matrisini (cf) bir daire matrisi ile çarparak (* sembolu MATLAB'da iki matrisin element-wise çarpımına eşittir)

```
>> [x,y]=meshgrid(-128:127,-128:127);
>> z=sqrt(x.^2+y.^2);
>> c=(z<15);
>> cf1=cf.*c;
>> imshow(mat2gray(log(1+abs(cf1))));

>> s=ifft2(cf1);
>> imshow(mat2gray((1+abs(s))));

> c=(z<50);
>> cf1=cf.*c;
>> imshow(mat2gray(log(1+abs(cf1))));
>> s=ifft2(cf1);
>> imshow(mat2gray((abs(s))));
```

Frekans domaininde yüksek geçiren filtre

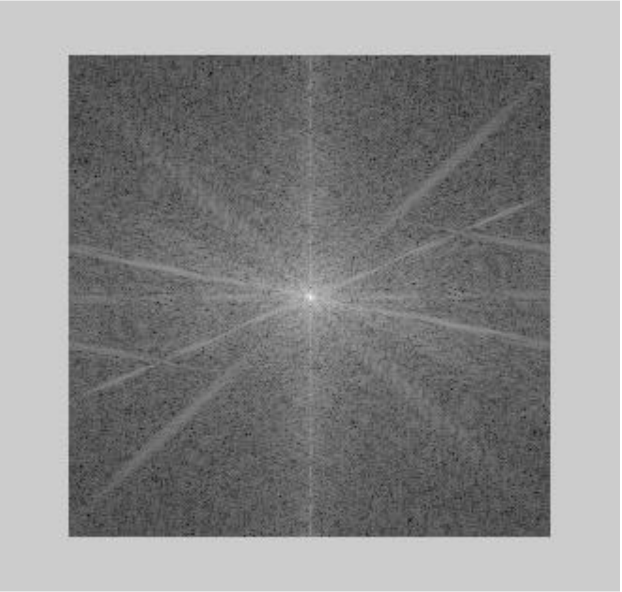
- Yüksek geçiren filtre ise, DC değeri merkeze kaydırılmış F matrisinin merkez veya merkeze yakın eleman değerlerini (Alçak bileşenleri) yok etmek veya zayıflatmak, merkezden uzak eleman değerlerini ise korumak şeklindedir. Filtre matrisi oluşturma işlemi alçak geçiren filtredeki gibidir. Tek fark aşağıdaki deyimdedir. Buradaki 15 örnek olarak verilmiştir.

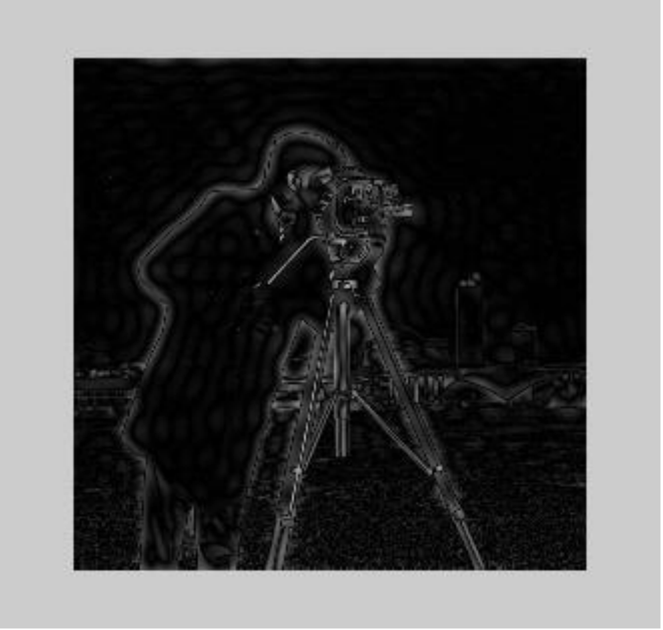
$$c = (z > 15)$$

- Oluşturulan c filtre matrisi ile F fft matrisi elemanter çarpma işlemine tabii tutulur. Filtreleme işlemi başarılmıştır.
- Bunun ters FFT'si İfft2 alınarak, filtrelenmiş görüntü uzaysal domaine dönüştürülür.
  - Yüksek geçiren filtreleme görüntüdeki nesne kenarlarının belirginleşmesine, gövdenin ise siyahlaşmasına neden olur.

Frekans domaininde yüksek geçiren filtre

```
>> cm=imread ('cameraman.tif');
>> [x,y]=meshgrid (-128:127,-128:127);
>> z=sqrt(x.^2+y.^2);
>> c=(z>15);
>> cf= fftshift(fft2(cm));
>> cfh=cf.*c;
>> imshow(mat2gray(log(1+abs(cfh))));
>> cfhi=ifft2(cfh);
>> imshow(uint8(abs(cfhi)))
```





Frekans domainde BUTTERWORTH Filtre

İdeal alçak ve yüksek geçiren filtreleme en kolay filtreleme teknikleridir. Ancak sonuç görüntüde nesneler çevresinde istenmeyen halkalar oluşur. Bundan kaçınmanın bir yolu, daha az keskin kesimli bir dairesel filtre matrisi kullanmaktır. Popüler bir seçim Butterworth Filtrelerdir.

Butteworth Filtreleri tarif etmeden önce, ideal filtrelere tekrar bakarsak; Bunlar radyal transform merkezine göre simetrik olduğundan, onları sadece merkeze göre tanımlayabiliriz.

Oysa biz merkezden olan x uzaklığının bir fonksiyonu olarak ta Filtre tanımlayabiliriz.

- İdeal alçak geçiren ve yüksek geçiren filtre fonksiyonları tarifi aşağıdaki gibi sıralanmıştı. Bu fonksiyonlarda D sabiti kesim yarıçapı idi.

$$f(x) = \begin{cases} 1 & \text{if } x < D, \\ 0 & \text{if } x \geq D \end{cases}$$

Alçak geçiren Filter Fonksiyonu

$$f(x) = \begin{cases} 1 & \text{if } x > D, \\ 0 & \text{if } x \leq D \end{cases}$$

Yüksek geçiren Filtre fonksiyonu

- Butterworth filtre fonksiyonları ise aşağıdaki bağıntılarla ifade edilir.

Alçak geçiren Butterwort Filtre Fonksiyonu

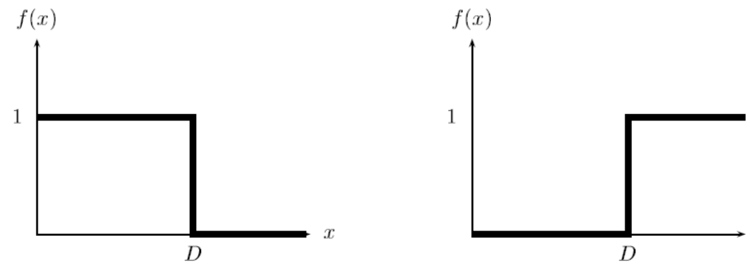
$$f(x) = \frac{1}{1 + (x/D)^{2n}}$$

Yüksek geçiren Butterworth Filtre Fonksiyonu

$$f(x) = \frac{1}{1 + (D/x)^{2n}}$$

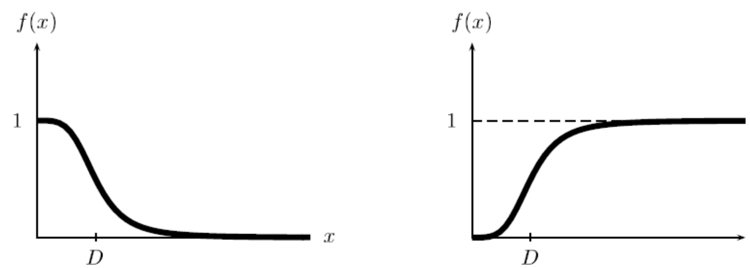
- Burada n filtrenin derecesidir. n’nin değeri kesimin keskinliğini verir.

Filtre Grafikleri



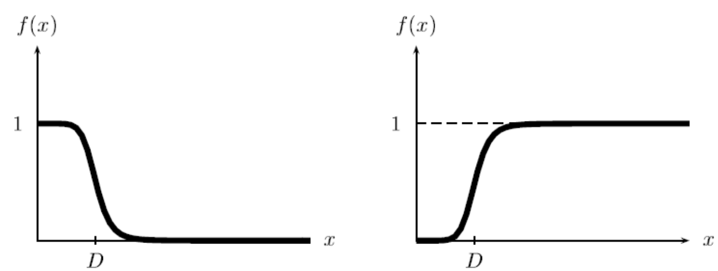
(a) Low pass (b) High pass

Figure 4.18: Ideal filter functions



(a) Low pass (b) High pass

Figure 4.19: Butterworth filter functions with $n = 2$



(a) Low pass (b) High pass

Figure 4.20: Butterworth filter functions with $n = 4$

- Butterworth filtreler MATLAB’da kolaylıkla gerçekleştirililirler. Örnek olarak Filtre boyutu; 256x256 olan D = 15 ve n=2 değerli, bir alçak geçiren filtre için;

```
>> [x,y]=meshgrid(-128:217,-128:127));
>> bl=1./(1+((x.^2+y.^2)/15).^2);
```

- Butterworth yüksek geçiren filtre ise **1-b1** şeklinde gerçekleştirilebilir.

But.worth 1.derece alçak geçiren filtreleme örneği

```
>> a=imread('cameraman.tif');
>> cf=fftshift(fft2(a));
>> [x,y]=meshgrid(-128:127,-128:127);
>> bl=1./(1+((x.^2+y.^2)/15).^1);
>> cfbl=cf.*bl;
>> b=ifft2(cfbl);
>> imshow(uint8(abs(b)))
```

Üsteki şekil ideal Alçak geçiren Filtreleme sonucu, alttaki şekil Butterworth 1. dereceden alçak geçiren filtreleme sonucudur.

Frekans domainde Gaussian Filtre

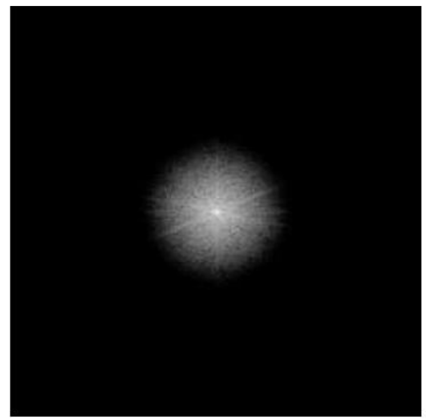
Gaussian filtreler uzaysal domande alçak geçiren filtreler olarak kullanılıyordu. **Frekans domaininde bir Gaussian filtreleme yapmak için; bir gaussian filtre matrisi oluşturun ve onu dönüştürülmüş görüntü matrisi ile çarpın. Sonucun inversini alın.** Uzaysal domandeki gaussian filtreler gibi, frekans domanindeki gausian filtrelerde lineer filtrelerdir ve oldukça iyi sonuçlar veririler

Gaussian filtreler pürüzsülük açısından (yumuşaklık) en iyisidir. Sonra ideal filtreler gelir. Butterwarth filtreler bu açıdan ortadır. MATLAB da Gaussian filtre oluşturmak için

fspecial fonksiyonu kullanılır.

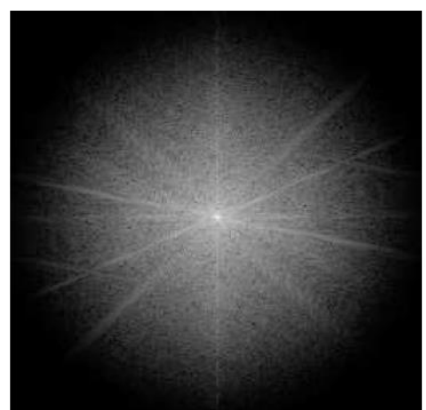
```
>> a=imread('cameraman.tif');
>> cf=fftshift(fft2(a));
>> g1=mat2gray(fspecial('gaussian',256,10));
>> cg1=cf.*g1;
>> cg=ifft2(cg1);
```

Yanda alçak geçiren gaussian filtreleme örnekleri görülmektedir. Yüksek geçiren gaussian filtreleme ise >> h= 1-g1 Şeklinde yazılır.



(a) $\sigma = 10$

(b) Resulting image



(c) $\sigma = 30$

(d) Resulting image