

Görüntü Restorasyonu

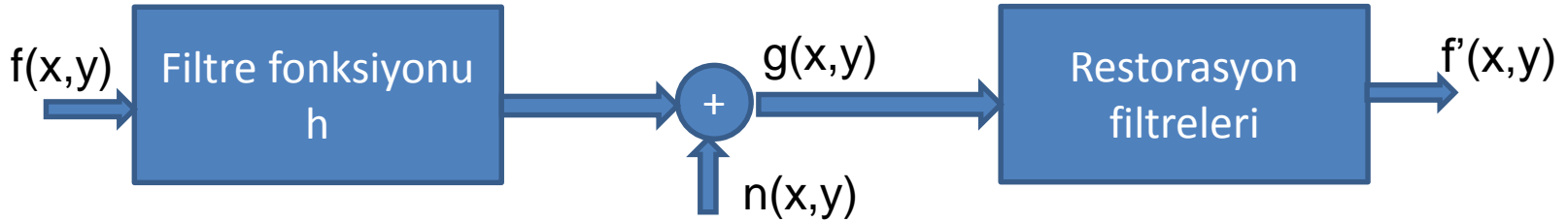
BMÜ-357 Sayısal Görüntü İşeme

Yrd. Doç. Dr. İlhan AYDIN

Görüntü İyileştirme (Image restoration)

- **Görüntü restorasyonu konusu**, bir görüntünün oluşumu esnasında oluşabilen veri kayıplarını veya bozulmaları azaltma veya tamamen yok etme konularını kapsayan; önemli konulardan birisidir.
- Görüntüdeki verim kaybı (görüntünün görünme derecesinin düşürülmesi) gürültülerden kaynaklanır. Gürültü pikselin gerçek değerindeki sapmadır.
- **Gürültü**; hareket veya atmosferik kararsızlık nedeniyle meydana gelen bulanıklaşma veya resmi çekerken yanlış ışık etkisinden dolayı focus bulanıklaşması, kusursuz olmayan lenslerden kaynaklanan geometrik bozulma ve elektronik kaynaklardan gelen hatalar olarak verilebilir.
- Bu etkileri yok etmek için kullanılan görüntü iyileştirme yöntemlerini göreceğiz. Bu teknikler uzaysal domainde komşuluk ilişkili yöntemler veya frekans domeninde yapılan işlemlerdir.
- Görüntü iyileştirme yöntemleri bir görüntünün görünüşünü, **görüntü derecesini aşağı düşüren bir matematik model** kullanan görüntü iyileştirme süreci kullanarak geliştirme işlemidir.

Derece düşürme Sistem Modeli



- Uzaysal domandede $f(x,y)$ gibi bir görüntümüz ve $h(x,y)$ gibi bir filtre fonksiyonumuz olsun. Bu $h(x,y)$ fonksiyonuna derece düşürme fonksiyonu da denir.

$$g(x,y) = f(x,y) * h(x,y)$$

İşleminden elde edilen $g(x,y)$ yeni görüntü, orijinal $f(x,y)$ görüntüsüne göre derecesi düşürülmüş bir görüntüdür. Ancak bu görüntü hala tam bozuk bir görüntü değildir. Modelin tamamlanabilmesi için buna bir gürültü fonksiyonu $n(x,y)$ eklemeliyiz.

$$g(x,y) = f(x,y) * h(x,y) + n(x,y)$$

Bu bağıntı uzaysal domainde, gürültüyü de göz önüne alan derece düşürme yöntemine göre sistemin modelidir.

Uzaysal alandaki modelde bulunan konvolusyon işlemi, frekans uzayındaki çarpıma eşdeğer olduğu için, frekans domainde derecesi düşürülmüş model aşağıdaki gibidir.

$$G(i,j) = F(i,j).H(i,j) + N(i,j)$$

Burada;

$G(i,j)$ = derecesi düşürülmüş görüntünün fourier dönüşümü.

$H(i,j)$ = Derece düşürme fonksiyonunun fourier dönüşümü

$F(i,j)$ = orjinal görüntünün fourier dönüşümü

$N(i,j)$ = eklenmiş gürültü fonksiyonunun fourier dönüşümü.

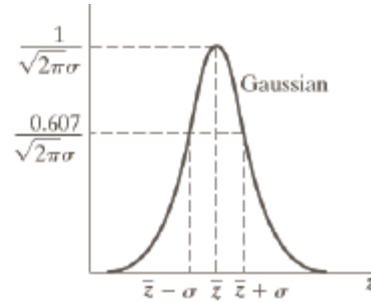
Bu bağıntılara göre; görüntü restorasyonun amacının, $h(x,y)$ veya $H(i,j)$ ve $n(x,y)$ veya $N(i,j)$ fonksiyonlarını bulmak olarak açıklayabiliriz

Gürültü (Noise)

- Gürültü, görüntüyü kirleten ve görüntü kalitesini düşüren harici kaynaklarda oluşan etkidir. Gürültü bir görüntü içerisinde bir çok farklı kaynaklardan meydana gelebilir.
- Optik bir görüntüyü, sürekli bir elektriksel sinyale çeviren, sayısal görüntü elde etme süreci, sayısal görüntülerde gürültünün ilk ortaya çıktığı süreçtir. Süreçteki her adım ise doğal olaylardan dolayı dalgalanmalar meydana getirir ve bu pikselin gerçek parlaklık değerine bir tesadüfi sayı ekler.
- Eğer hangi adımda hangi tip gürültünün oluştuğu bilinebilseydi bunun temizlenmesi de sıfır hata ile gerçekleşebilirdi. Ancak bu durum pratikte mümkün değildir. Bunun için standart tanımlanmış bazı gürültülere göre işlem yapılacaktır. Bu gürültü örneklerinin en önemlileri;
 - Tuz-Biber (Salt-pepper) gürültüsü
 - Gaussian (normal) gürültü
 - Uniform Gürültü

- **Gaussian Gürültü:** Bir olasılık yoğunluk fonksiyonu ile karakterize edilen rastgele değişkenler Matlab fonksiyonu randn

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

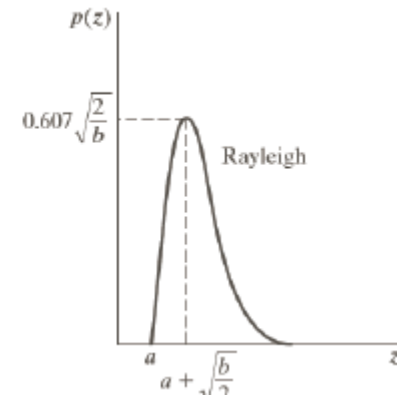


- **Rayleigh Gürültüsü:**
Matlab fonksiyonu $z=a+\text{sqrt}(b*\ln[1-U(0,1)])$

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

Mean: $\bar{z} = a + \sqrt{\pi b/4}$

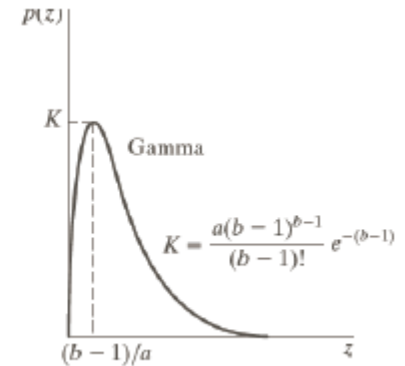
Variance: $\sigma^2 = \frac{b(4-\pi)}{4}$



Erlang (Gamma) Gürültüsü: Matlab fonksiyonu $z=z+(-1/a)*\log(1-\text{rand}(M,N))$

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \quad \begin{matrix} a > 0 \\ b \text{ positive integer} \end{matrix}$$

$$\text{Mean: } \bar{z} = \frac{b}{a} \quad \text{Variance: } \sigma^2 = \frac{b}{a^2}$$

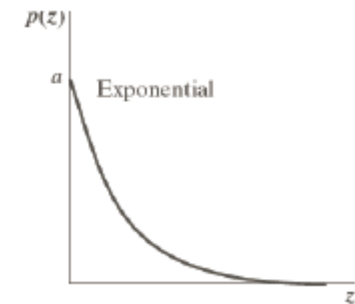


Exponential Noise:

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

(cf. Erlang noise with $b = 1$)

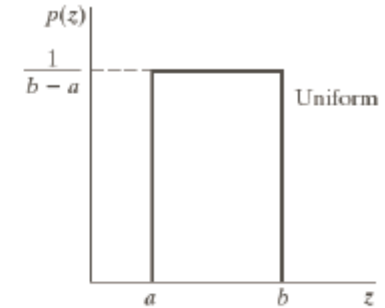
$$a > 0$$



- Uniform Gürültü: Matlab fonksiyonu rand

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

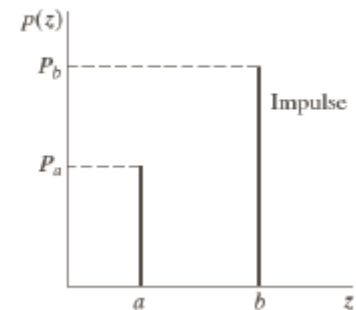
$$\text{Mean: } \bar{z} = \frac{a+b}{2} \quad \text{Variance: } \sigma^2 = \frac{(b-a)^2}{12}$$



Bipolar impulse noise (salt-and-pepper)

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

$$P_a = P_b \Rightarrow \text{unipolar noise}$$

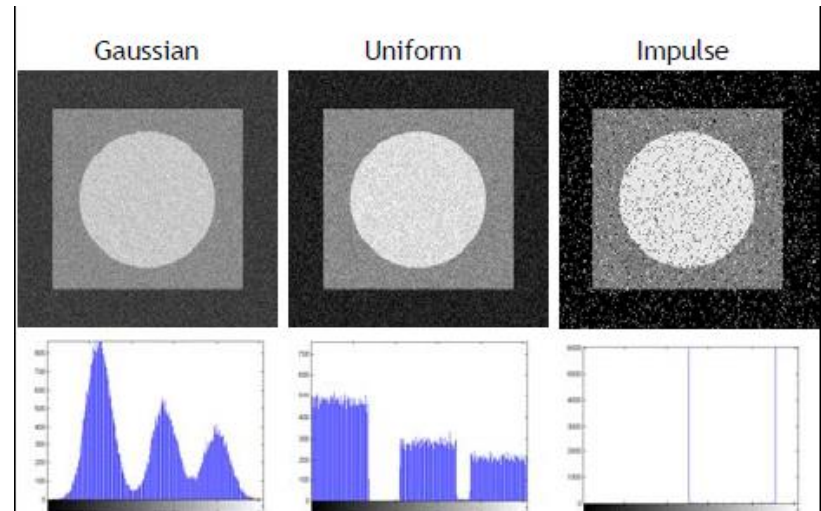
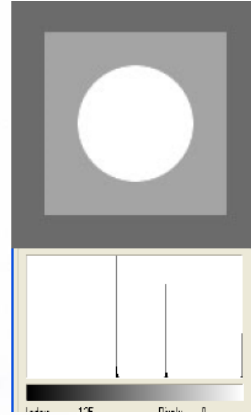
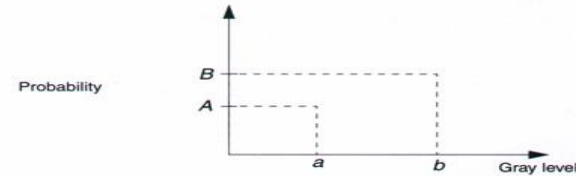
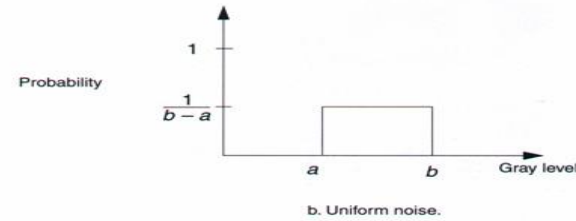
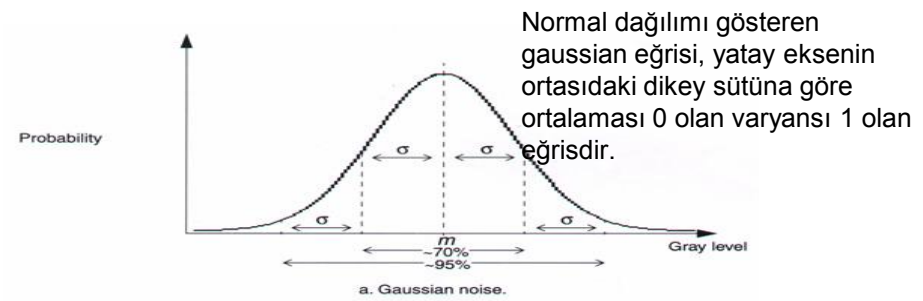


Bu gürültü tiplerinin dağılımının şekli, bir histogram olarak modellenebilir ve Şekil'de görülmektedir.

Şekil a 'da gauss gürültü dağılımının çan şeklinde değiştiği görülmektedir. Bunun anlamı gürültünün gri seviye değerlerinin dağılım olasılığının şekildeki gibi olacağıdır.

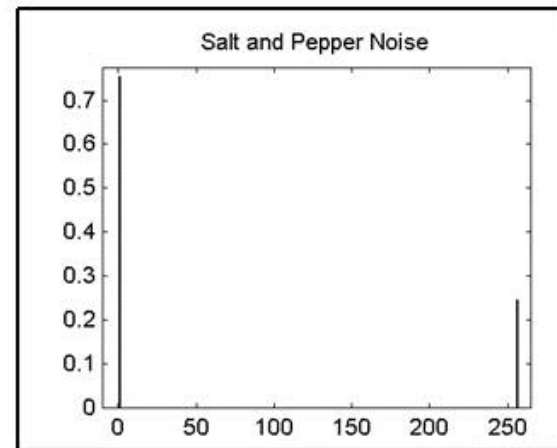
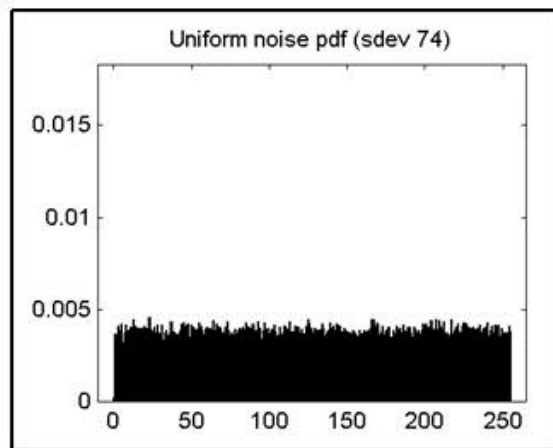
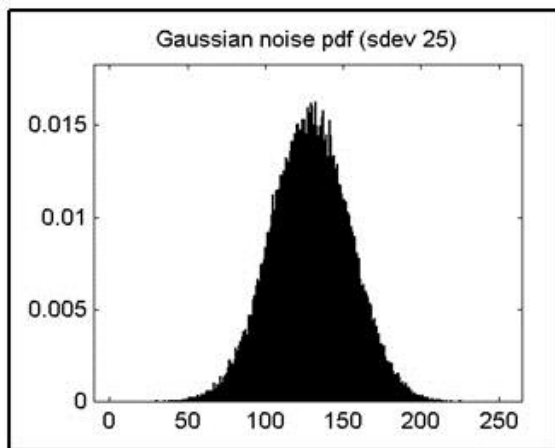
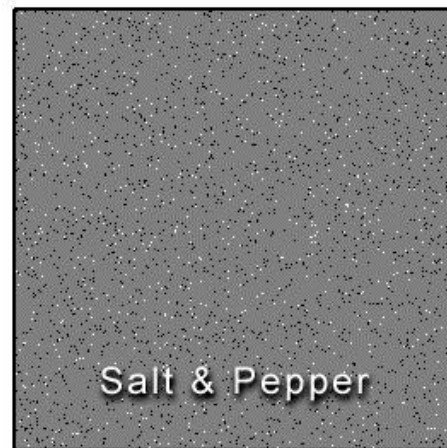
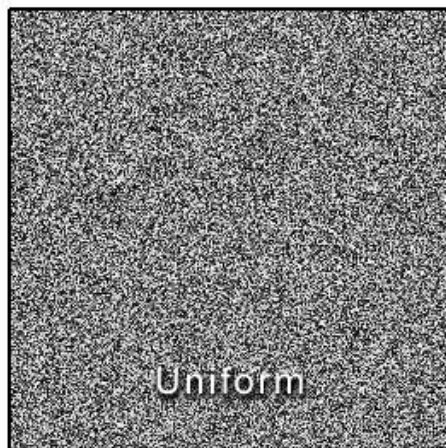
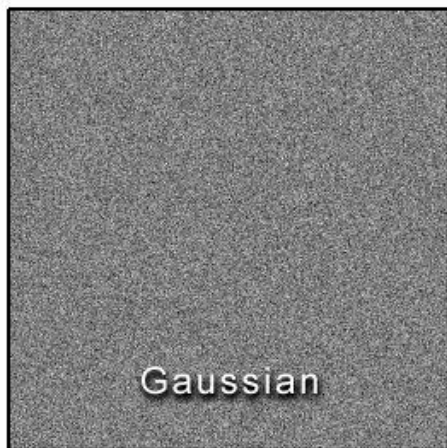
Şekil b'de Uniform dağılımlarda, gürültünün gri seviyesi değerlerinin belirli bir aralık boyunca düzenli bir şekilde dağıldığı görülmektedir. Bu değer tüm aralık boyunca (0-255, 8 bit için) olabilir veya tüm aralığın küçük bir kesiti olabilir.

Şekil c 'de bir tuz-biber (salt and pepper) dağılımı görülmektedir. Tuz-biber gürültü modelinde a ve b gibi sadece iki mümkün değer vardır ve her birisinin olasılığı 0.1 'den daha düşüktür. Bundan daha büyük değerlerde gürültü görüntü üzerinde hakim duruma geçer.



Tuz - Biber Gürültüsü

- İmpulse gürültüsü veya Binary gürültü olarak tanımlanır. Bu gürültü görüntüde keskin dağılımlıdır. Görüntü üzerinde belirgin olarak siyah ve beyaz noktalar şeklindedir.
- Tuz biber modeli tipik olarak kameranın algılayıcılarındaki piksel elemanlarının çalışmalarındaki bozukluklardan, hatalı bellek bölgelerinden yada sayısallaştırma sürecindeki zamanlama hatalarından kaynaklanan gürültüleri modeller.
- 8 bitlik bir görüntü için tipik değerler biber için 0, tuz için 255 olarak alınabilir



Tuz-Biber gürültüsü

Orijinal görüntü piksellerini gelişigüzel olarak siyah veya beyaz yapar.

Matlabda;

A= imnoise (b, 'salt & pepper', d);

ile işlenir. Burada b orijinal görüntü, d ise tuz-biber gürültüsünün derecesidir.

Default değeri 0.05'dir. D 0.1'den büyük olmamalıdır.

```
>> b=imread('cameraman.tif');
```

```
>> imshow(b)
```

```
>> c=imnoise(b,'salt & pepper');
```

```
>> imshow(c)
```

```
>> d=imnoise(b,'salt & pepper',0.1);
```

```
>> imshow(d)
```

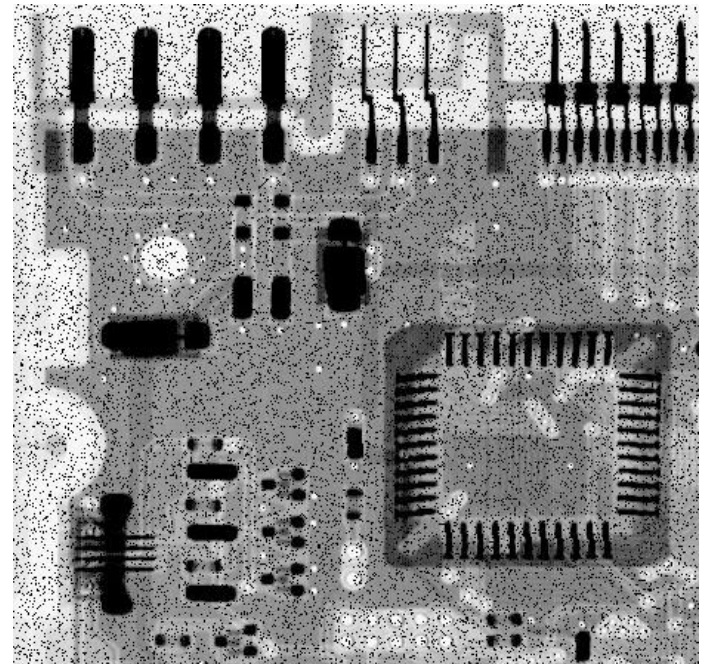
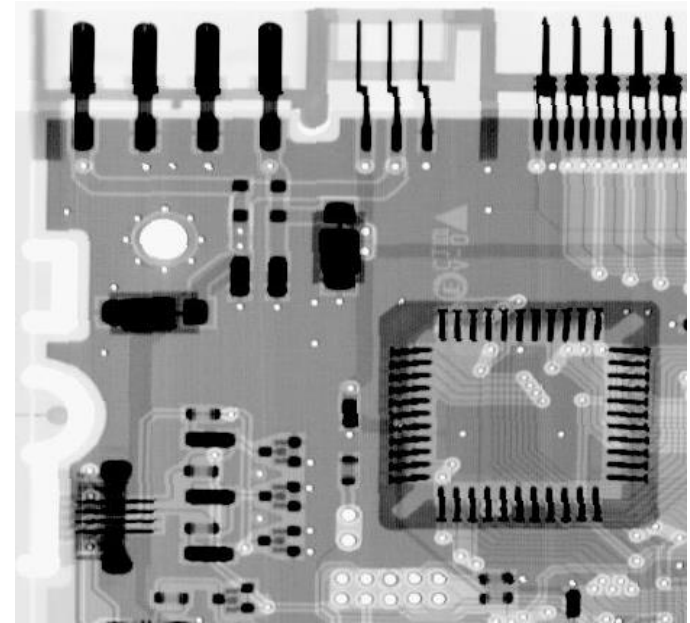


Örnek: Tuz biber gürültüsü

```
function R=tuzbiber(M,N,a,b)
if nargin<=2
    a=0.05;
    b=0.05;
end
if (a+b)>1
    error('Pa ve Pb toplami 1 den büyük olamaz');
end
R(1:M,1:N)=0.5;
X=rand(M,N);
c=find(X<=a);
R(c)=0;
u=a+b;
c=find(X>a & X<=u);
R(c)=1;
```

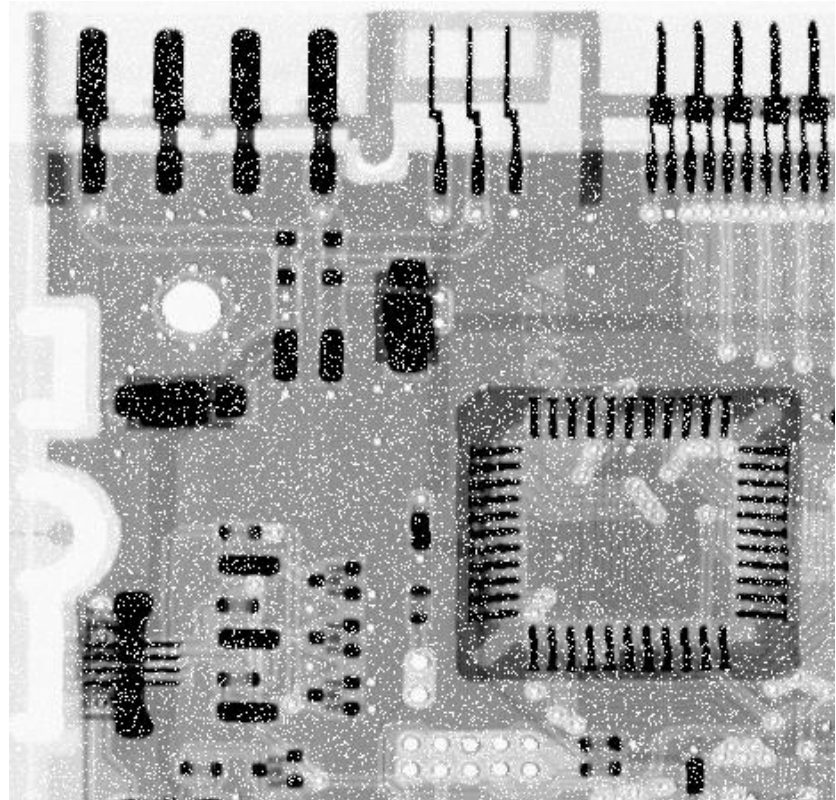
Örnek: biber gürültüsü eklemek

```
f=imread('cameraman.tif');  
figure  
imshow(f)  
[M, N]= size(f)  
R = tuzbiber(M, N, 0.1, 0);  
c = find (R==0);  
gp = f;  
gp(c) = 0;  
figure  
imshow(gp)
```



Örnek: tuz gürültüsü eklemek

```
f=imread('Fig2.tif');  
figure; imshow(f)  
[M, N]= size(f);  
R = tuzbiber(M, N, 0,  
0.1);  
c = find (R==1);  
gs = f;  
gs(c) = 255;  
imshow(gs);
```



Gaussian Gürültüsü

- Gauss modeli doğal gürültü sürecini modellemek için kullanılan en genel yöntemdir. Bu yöntem özellikle görüntü elde etme sürecinde elektriksel olarak ortaya çıkan gürültüleri modellemek için kullanılır. Bu gürültü görüntüde gelişigzel dalgalanmalara sebep olur. Görüntü fonksiyonu $I(x,y)$ ise ve $N(x,y)$ 'de Gaussian gürültüsünü temsil ediyorsa; gürültülü görüntü;

$$g = I + N$$

İle ifade edilir. Matlabda Gaussian gürültüsü eklemek için;

`d=imnoise (g, 'gaussian',M,V)`

Şeklindedir. Burada M ortalama, V variance değerleridir. Default olarak $M=0$ $V=0.01$ alınır.

$d = \text{imnoise}(g, 'gaussian', M, V)$

Original Görüntü

$M=0$, $V=0.01$ (default değerdir)



(a) Original image

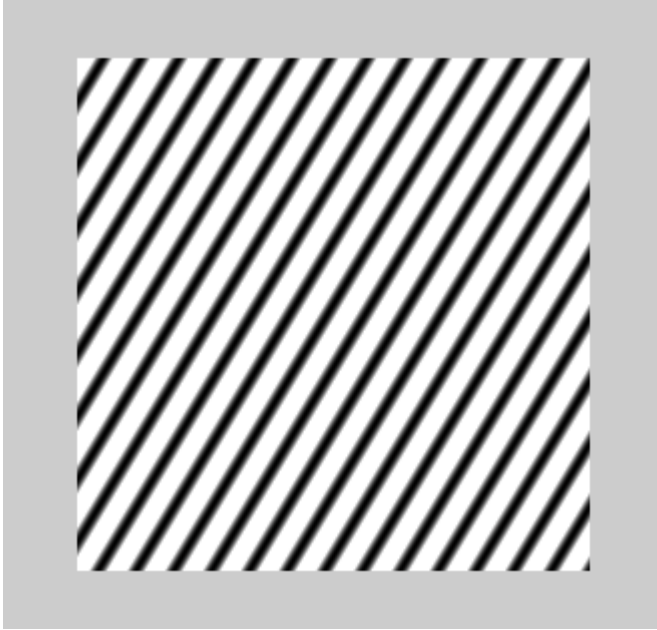


(a) Gaussian noise

Periyodik Gürültü

- Eğer görüntü işaretinde periyodik olarak değişen nesneler var ise; (yani rastgele bir şekilde bozukluk yoksa – örneğin görüntü üzerinde tekrar eden çubuk şeklinde nesneler var ise); biz periyodik bir gürültü tarafından bozulmuş bir görüntü elde ederiz. Matlabdaki `imnoise` fonksiyonu ile böyle bir gürültü elde edemeyiz. Bunun için;
- `>> t=imread('cameraman.tif');`
- `>> s=size(t);` t'nin boyutunu s'ye aktarıyor.
- `>> [x,y]=meshgrid(1:s(1),1:s(2));` x,y boyutlu matris.
- `>> p=sin(x/3+y/5)+1;` sinüs
- `>> imshow(p)`
- `>> k=(im2double(t)+p/2)/2;`
- `>> imshow(k)`

Periyodik gürültü ve gürültü eklenmiş orijinal resim.



- Tuz-Biber, Gaussian ve Uniform gürültüler uzaysal doman filtreleri ile temizlenebilir. Bu uzaysal filtreler küçük sayıda (3×3 11×11) komşuluk ilişkisi ile çalışırlar.
- Fakat periyodik gürültüyü temizlemek için frekans domani filtrelerine ihtiyaç duyulur.

Tuz-Biber Gürültüsünü Yok etme

- Tuz ve biber gürültüsü yüksek frekans bileşenli görüntüdür. Bunun için alçak geçiren filtreler kullanarak bu gürültü tipini yok edebiliriz veya azaltabiliriz. Bunun için bir ortalama filtre (Mean filtreleri görüntü kenarlarını ve detaylarını bulanıklaştırma dezavantajına sahiptirler - Aslında Mean filtreleri gauss ve uniform gürültü tiplerinde oldukça etkilidir.) deneyebiliriz.
- Yapılması gereken, fspecial('average') ile bir ortalama filtre matrisi oluşturup, bunu gürültülü görüntü ile konvoluse etmektir.

Varsayalım ki t , gürültülü görüntü olsun,

```
>> a=fspecial ('average');  
>> b=fspecial ('average', [7,7]);  
>> tt= filter2(a, t);  
>> tt= filter2(b, t);
```

Not: gürültü resmin üzerine çok etken ise; büyük boyutlu bir ortalama filtre kullandığımızda orijinal resimde oldukça bulanıklaşacaktır.



(b) With added salt & pepper noise



(a) 3×3 averaging



(b) 7×7 averaging

Median Filtreleme

Median filtreleme, tuz-biber gürültüsünü yok etmek için çok uygundur. Medyan filtreler nonlineer uzaysal filtrelerdir. Maskeyi oluşturan boyuttaki resim piksel değerlerinin küçükten büyüğe sıralanıp ortadaki değeri merkez piksele atama işlemiydi. Örneğin;

50	65	52
63	255	58
61	60	57

 → 50 52 57 58 **60** 61 63 65 255 → 60

```
>>t=imread('cameraman.tif');  
>> c=imnoise(t,'salt & pepper',0.1);  
>> imshow(c)  
>> d=medfilt2(c);  
>> imshow(d)
```



- Maksimum ve minimum filtrelerde , tuz-biber (impulse) gürültüsünü yok edebilmek için kullanılan iki order filtresidir.
- Maksimum filtre sıralanmış piksel değerlerinin en büyüğünü seçerken, minimum filtre en küçük değeri seçer.
- Minimum filtreleme tuz tipi gürültünün hakim olduğu durumlarda kullanılırken maksimum filtreler biber tipi gürültü için en iyi sonucu vermektedir.
- Bu filtrelerin bu tip gürültüler için, en az detay kaybı ile birlikte çok iyi çalıştığını görmekteyiz. Pencere boyu büyüdükçe bilgi kaybının oranı gittikçe artmaktadır.

Örnek: Max filtre ile gürültü temizleme

```
function sonuc=maksfiltre(A)
figure,imshow(A),title('ORIGINAL IMAGE');
B=zeros(size(A));
modifyA=padarray(A,[1 1]);
x=[1:3]';
y=[1:3]';
for i= 1:size(modifyA,1)-2
    for j=1:size(modifyA,2)-2
        window=reshape(modifyA(i+x-1,j+y-1),[],1);
        B(i,j)=max(window);
    end
end
sonuc=uint8(B);
figure,imshow(sonuc),title('IMAGE AFTER MAX FILTERING');
```

Çalıştırılması:

ORIGINAL IMAGE



IMAGE AFTER MAX FILTERING



Gaussian Gürültüsünün temizlenmesi

Görüntünün ortalamasının alınması ile:

Bazen Gauss gürültüsü ile bozulmuş sadece bir görüntü yerine, bunun çok farklı bir kopya olduğu ortaya çıkabilir. Örneğin uydu görüntülemeye olduğu gibi; bir uydu birçok kez, aynı yerde birçok farklı görüntüler elde edecek şekilde geçebilir. Başka bir örnekte, bir mikroskoptan aynı görüntüye bakarak farklı image'ler elde edilebilir.

Böyle bir durumda Gauss gürültüsünü temizlemek için çok basit bir yaklaşım; sadece tüm görüntülerin ortalamasını almaktır.

Elimizde , bizim görüntünün 100 kopyasının olduğunu varsayın. Bunların herbiride gürültülüdür. İ'ci gürültülü görüntü;

$M+N_i$ dir. Burada M orijinal değerler matrisi, N_i ortalaması 0 olan normal dağıtılmış gelişigüzel değerler matrisidir. Biz bu görüntüleri kullanarak M' ortalama görüntü matrisini bulabiliriz.

$$\begin{aligned} M' &= \frac{1}{100} \sum_{i=1}^{100} (M + N_i) \\ &= \frac{1}{100} \sum_{i=1}^{100} M + \frac{1}{100} \sum_{i=1}^{100} N_i \\ &= M + \frac{1}{100} \sum_{i=1}^{100} N_i \end{aligned}$$

N_i 'ler, normal dağılımlı ortalamaları 0 olduğundan, tüm N_i 'lerin ortalamalarının 0'a çok yakın veya 0 olduğunu söyleyebiliriz. Bunun anlamı $M' = M$ 'dir.

Ortalama Filtre

Mean filtreleri NxN penceresi içerisinde bir ortalama değeri bulmaya çalışarak işlerler. Bunu yaparken görüntü üzerinde gezen bir kayan pencere yöntemi kullanılır. Bu filtrelerin en basitlerinden birisi aritmetik mean filtresidir. Bu filtre pencere içerisindeki değerlerin aritmetik ortalamasını bulurlar.

- Arithmetic mean (corresponds to box filter)

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- Geometric mean (nonlinear!)

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{1/mn}$$

- Harmonic mean (nonlinear!)

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

Aritmetik mean filtresi, görüntü içerisindeki yerel değişimleri yumuşatır yani aslında bu bir alçak geçiren filtredir. Bu, tüm katsayıları $1/N^2$ ile çarpılarak, maskenin görüntü üzerine konvolusyonu ile elde edilebilir. Bu filtre gürültü etkilerini yok ederken görüntüyü bulanık bir hale getirmeye çalışacaktır. Maske boyutları büyüdükçe bulanıklaşma etkileri o kadar çok meydana gelmektedir. Bu tip filtreler en iyi şekilde gauss ve uniform gürültülerde çalışırlar. Eğer gauss gürültüsünün ortalaması 0'sa gürültü filtreleme ile 0 yapılabilir. Ancak bir miktarda görüntüde bulanıklaşmayı göz önüne alınız. Filtre boyutu 3x3 veya 5x5 olabilir.

```
>> a3=fspecial('average');  
>> a5=fspecial('average',[5,5]);  
>> tg3=filter2(a3,t_ga);  
>> tg5=filter2(a5,t_ga);
```



(a) 4×3 averaging



(b) 5×5 averaging

Adaptif Filtreler

Adaptif filtreler, maskesi altında gri ton değerlerine göre özelliklerini değiştirebilen filtre sınıfındandır. Onlar görüntü içinde bulundukları yere bağlı olarak, daha çok medyan filtre gibi, veya daha çok ortalama filtreler gibi hareket edebilirler.

Filtre davranışının temel kriteri yerel karakteristiklerdir ve genelde yerel gri seviyesi istatistikleri ile ölçülürler.

Minimum mean square error (MMSE) filtresi ise adaptive filtrelerine uygun bir örnektir. Nonlineer bir filtredir. Bu filtre yerel görüntü istatistiklerine göre değişim göstermektedir. MMSE filtresi en iyi sonuçlarını gauss ve uniform gürültülerde vermektedir. Aşağıda MMSE'nin fazla kullanılan bir versiyonunun bağıntısı verilmektedir.

$$\text{MMSE} = d(r, c) - \frac{\sigma_n^2}{\sigma_l^2} [d(r, c) - m_l(r, c)]$$

σ_n^2 = gürültü değişimi (Gürültü varyansı)

σ_l^2 = Lokal değişim (Yerel Varyans) (pencere faktörü altındaki)

m_l = Lokal ortalama (pencere faktörü altında ortalama).

$d(r, c)$ = Orijinal, filtrelenmemiş görüntü.

- Görüntüdeki gürültüsüz durumlarda gürültü değişimi sifıra eşittir, ve bu eşitlik geriye orjinal filtrelenmemiş bir görüntü döndürür. Görüntünün arka plan bölgelerinde, bozulmamış, orjinal görüntüdeki sabit değerlerde gürültü varyansı yerel varyansa eşit olur ve eşitlik bir mean filtresi üretir.
- Yerel varyansların, gürültü varyanslarından daha büyük oldukları bölgelerde eşitlik filtrelenmemiş veriye yakın bir değer döndürür. Bu, yüksek yerel varyansın yüksek detay (kenarlar) içerdiği durumlarda istenilen bir şeydir, adaptive filter burada detayları saklamaya çalışır. Genelde MMSE filtresi, yerel ortalamanın eklenmiş olduğu orjinal değerlerin bir kısmının filtrelenmemiş değerden farkının alınmış halini geri döndürür. Orjinal ve yerel ortalama miktarı orjinali değiştirmek için σ_n^2 / σ_i^2 gürültünün yerel varyansa oranını alır. Bu oranın artmasıyla penceredeki birincil gürültü önem kazanır ve filtre yerel ortalamayı döndürür. Oran azaldıkça yüksek yerel detaylar baskın olur ve orjinal filtrelenmemiş görüntüyü döndürür. Bu anlamda çalışarak MMSE filtresi kendisini görüntünün yerel özelliklerine uyarlar. Böylece gürültüleri yok ederken görüntü detaylarını saklamaya çalışır.

Matlab'da adaptif Filtre kullanımı

Matlabda adaptif filter uygulamalarında genellikle **wiener2** fonksiyonu kullanılır.

```
>> t1=wiener2(t_ga);  
>> t2=wiener2(t_ga,[5,5]);  
>> t3=wiener2(t_ga,[7,7]);  
>> t4=wiener2(t_ga,[9,9]);
```

Resimde gaussian gürültülü bir görüntünün (t_ga) değişik maske boyutlu adaptif filtre ile temizlenmesi görülmektedir.

Alçak geçiren filtre olan adaptif filtrelemede kenarlarda bulanıklık eğilimi yok. Görüntüde yüksek frekans bileşenler etkilenmemiş.



(a) 3×3 filtering



(b) 5×5 filtering



(a) 7×7 filtering



(b) 9×9 filtering

Periyodik Gürültünün yok edilmesi

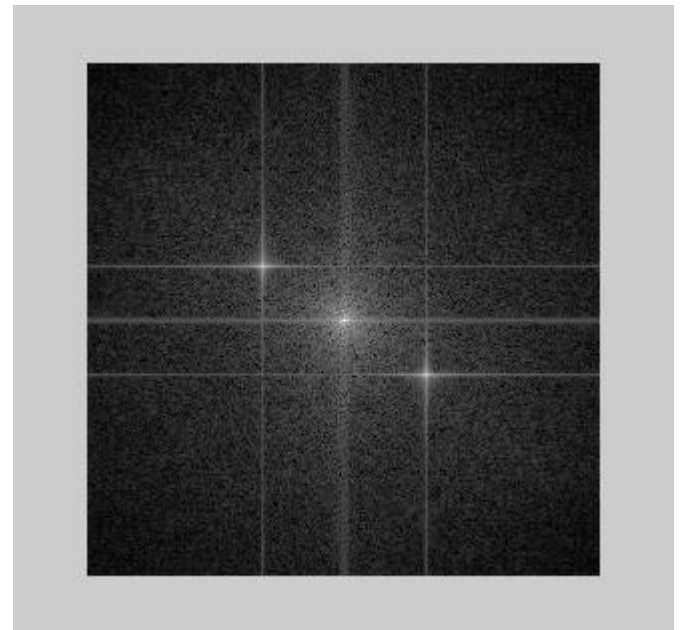
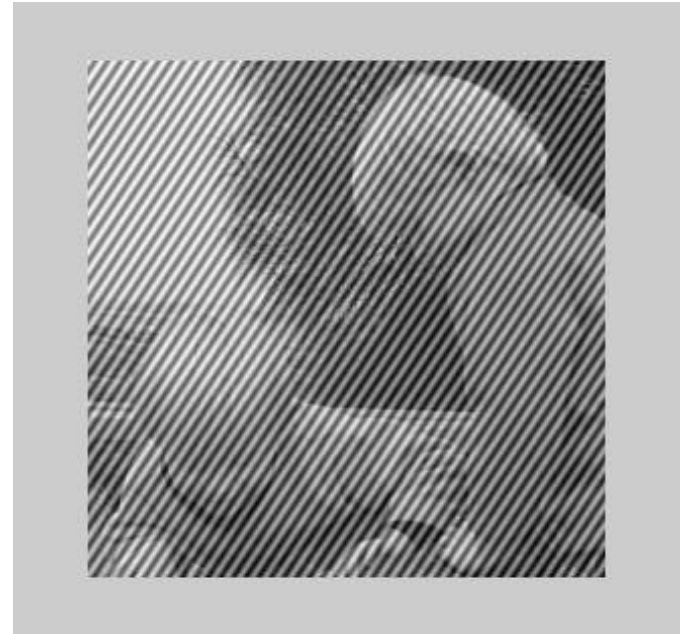
Görüntü işareti periyodik olarak davranan bozucu büyüklükler var ise; (yani rastgele bir şekilde bozukluk yoksa- Görüntü üzerinde örneğin çubuk şeklinde nesneler varsa); biz periyodik bir gürültü tarafından bozulmuş bir görüntü oluştuğunu daha önceki kısımlarda açıklamıştık. Biz bu periyodik gürültüyü trigonometrik fonksiyonlar ile yaratıp orijinal görüntünün üzerine ekleyebiliriz.

Bu aşağıda verilmektedir.

```
>> a=imread('twins.tif');  
>> b=rgb2gray(a);  
>> [x,y]=meshgrid(1:256,1:256);  
>> p=1+sin(x+y/1.5);  
>> tp=(double(b)/128+p)/8;  
>> imshow(tp);  
>> af=fftshift(fft2(tp));  
>> imshow(mat2gray(log(1+abs(af))));
```

Burada dikkat edilecek husus şudur; spectrumda, merkezdeki beyazlık görüntünün ortalama değeridir, ve merkeze yakın ve çaprazlarındaki beyazlıklar ise genelde periyodik gürültünün simetrik ortalama değerini ifade eder.

Merkezde olmayan bu DC değerlerin anlamı periyodik gürültünün yüksek frekanslı (küçük mesafede, gri seviide büyük değişim etkisi) olduğudur. Gürültü DC değerleri merkeze ne kadar uzaksa, gürültünün periyodu o kadar küçüktür denir.



- Bizim bu periyodik gürültüyü yok etmemiz için spekturm resmindeki merkezden uzak pikleri (periyodik gürültü dc değerini) yok etmemiz gerekir.
- Not: Matlab'da elde edilen bir görüntü üzerindeki herhangi bir noktanın x,y olarak koordinatlarını bilmek için;

>> impixelinfo

Komutunu çalıştırın. Resim üzerinde mousla durduğunuz noktanın (pikselin) koordinatları belirir.

Buna göre spectrum resmindeki periyodik gürültünün pik yaptığı pikseller; satır sütun olarak (102,88) ve (156,170) olarak belirlenir. Bu noktaların merkeze olan mesafesi ise 49 olarak elde edilir. Bu koordinatlar bilindiğine göre bu pikleri yok etmek için iki önemli filtreleme kullanabiliriz.

Band söndüren (durduran) Filtreleme

Band söndüren filtrelemenin temeli alt ve üst sınırlar içinde kalan frekans bileşenlerini yok eden bu bantın dışındaki frekans bileşenlerine etki etmeyen bir filtrelemedir.

Buna göre bizim durdurulacak frekanslar için bandımız;

47 ve 51 yarıçaplı iki adet iç içe dairedir (Yarıçapı 49 olan bir bant).

Bunu Matlab'da gerçeklemek için;

```
>> z=sqrt((x-129).^2+(y-129).^2);
```

```
>> z(102,88)
```

```
ans =
```

```
49.0918
```

```
>> z(102,88)
```

```
ans =
```

```
49.0918
```

```
>> br=(z < 47 | z > 51); band geçiren filter
```

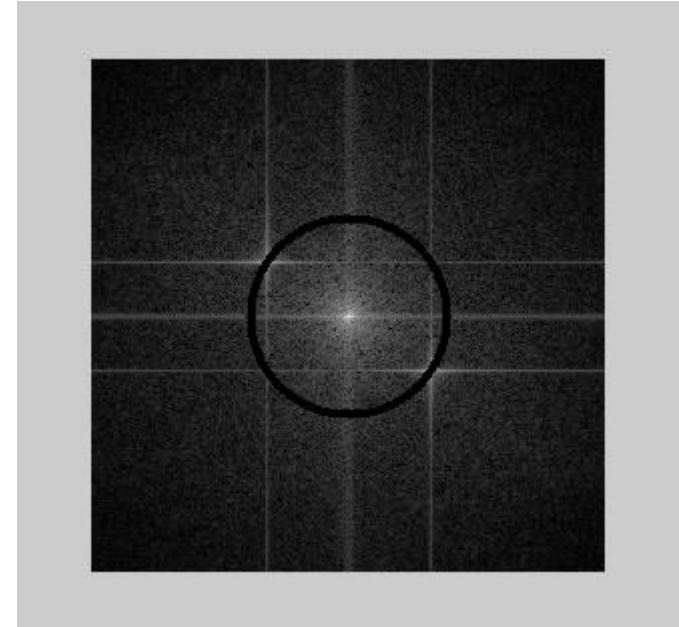
```
>> tbr = af .* br; elemanter çarpma
```

```
>> imshow(mat2gray(log(1+abs(tbr))));
```

```
>> s=ifft2(tbr);
```

```
>> imshow(mat2gray((1+abs(s))));
```

Band durduran filtrenin, periyodik gürültüyü tamamen yok ettiği söylenemez. Çünkü periyodik gürültünün merkeze yakın önemli frekans bileşenleri yok edilmemiştir.

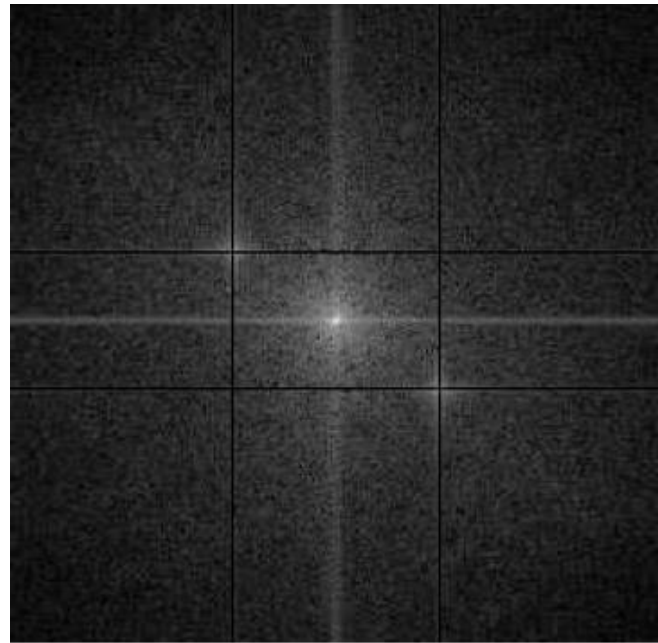


NOTCH (Çentik) FİLTRE

Bu filtreleme işlemi, sadece seçilmiş satır ve sütunlardaki pikleri 0 yapar.

```
>> tf(156,:)=0;  
>> tf(102,:)=0;  
>> tf(:,170)=0;  
>> tf(:,88)=0;
```

Gürültünün büyük bir kısmı bu şekilde yok edilmiş olur.



İnverse Filtering (Ters Filtreleme) ile görüntü düzeltme

Biz frekans domaininde filtreleme yapmak için: Görüntünün FFT'si ile filtrenin fft'sini çarpıyorduk. Bu durum Frekans domaininde konvolusyondur.

$$Y(i, j) = X(i, j)F(i, j)$$

X, görüntünün DFT'si, F filtrenin DFT'si Y ise elde edilen sonuç görüntünün DFT'si idi. Eğer elimizde Y ve F var ise; orijinal görüntünün DFT'sini bulmak için aşağıdaki işlemi yaparız. Buradan da İFFT ile X(x,y) uzaysal domaine dönülür.

$$X(i, j) = \frac{Y(i, j)}{F(i, j)}.$$

Bu durumda çok önemli bir sorun olabilir. Eğer F(i,j) Filtre eleman değerleri çok küçük ise bölüm X(i,j) görüntü FFT'si istenildiği gibi çıkmaz. Bu sorunla 2 şekilde baş edilir;

1- Bölüm terimine bir alçak geçiren filtre L(i,j) uygulanabilir.

$$X(i, j) = \frac{Y(i, j)}{F(i, j)}L(i, j).$$

2- "Limitli Bölme İşlemi": bir d eşik değeri seçilir. Ve $|F(i, j)| < d$ ise bölme yapılmaz.

$$X(i, j) = \begin{cases} \frac{Y(i, j)}{F(i, j)} & \text{if } |F(i, j)| \geq d, \\ Y(i, j) & \text{if } |F(i, j)| < d. \end{cases}$$

Hareket bulanıklığını yok etme (Hareket netleştirme)

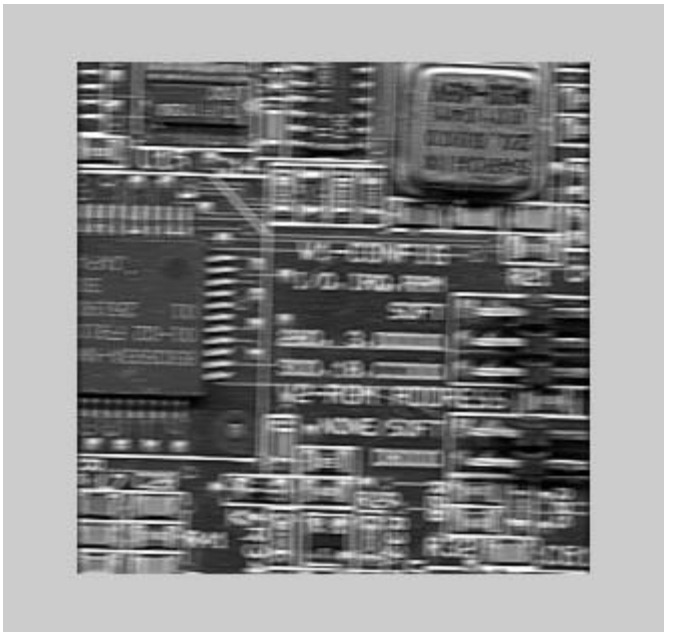
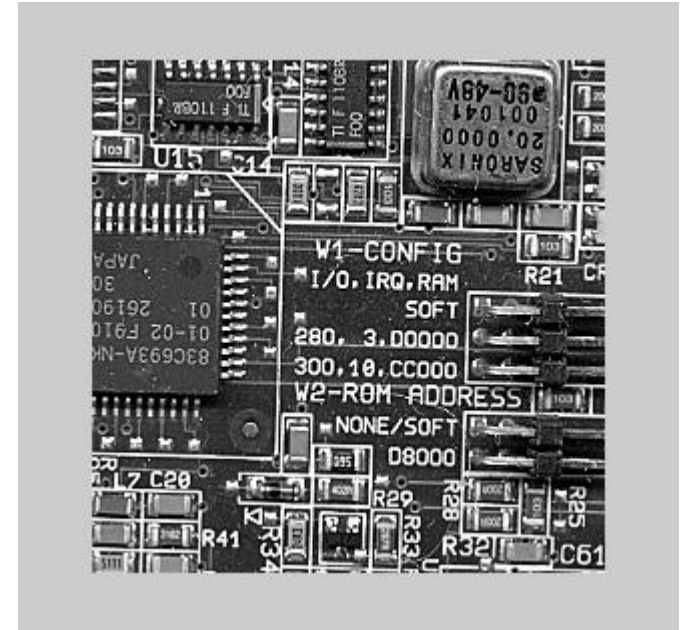
Görüntüyü oluştururken hareket etmesinden dolayı oluşan bulanıklıkları yok etmek için bir filtreleme düşünülebilir. Varsayalımki bir görüntü çekerken küçük miktarda bulanıklık oluşmuştur.

```
>> bc=imread('board.tif');  
>> bg=im2uint8(rgb2gray(bc));  
>> b=bg(100:355,50:305);  
>> imshow(b)  
Bu resmi bulanıklaştırmak için;  
>> m=fspecial('motion',7,0);  
>> bm=imfilter(b,m);  
>> imshow(bm)
```

Şimdi elimizde hareket bulanıklılığına sahip bir görüntü var. Bu görüntüyü netleştirmek için, Bulanık görüntünün FFT'sinin, bulanıklığı yaratan filtrenin FFT'sine bölünmesi gerekir.

Bunun anlamı bizim işimiz; bulanıklık dönüşümüne karşılık gelen bir matris yaratmamızdır.

```
>> m2=zeros(256,256);  
>> m2(1,1:7)=m;  
>> mf=fft2(m2);
```



Şimdi bu FFT' ile görüntü FFT'sini bölelim.

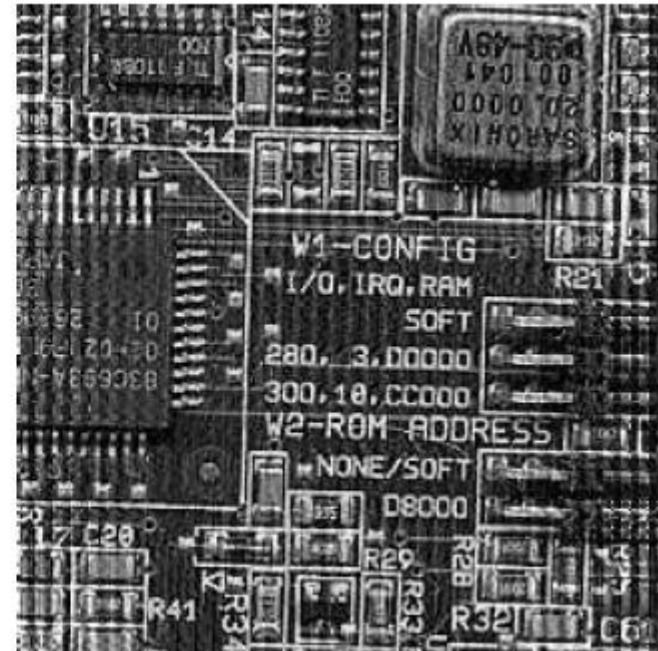
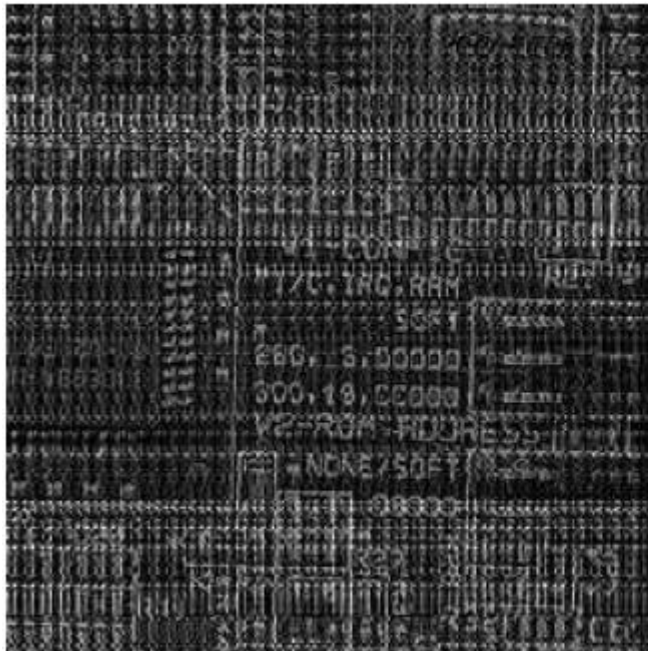
```
>> bmi=ifft2(fft2(bm)./mf);
```

Ancak bu mf filtresinin katsayıları küçük olduğundan görüntü düzeltme kötüdür. Bunun yerine aynı işlemi eşik değeri ile yaparsak ($D=0.02$)

```
>> d=0.02;
```

```
>> mf=fft2(m2);mf(find(abs(mf)<d))=1;
```

```
>> bmi=ifft2(fft2(bm)./mf);
```



Ödev

Teslim Tarihi:
10.12.2013

Bir görüntüye farklı gürültüler eklemek için aşağıdaki şekilde bir fonksiyon yazınız?

$S = \text{gurultuekle}(\text{imge}, M, N, a, b, \text{gurultutipi})$

$\text{imge} \rightarrow$ Gurultu eklenecek gray imge

$M \rightarrow$ imgenin satır sayısı

$N \rightarrow$ imgenin sütun sayısı

$a, b \rightarrow 0$ ile 1 aralığında seçilen gürültü fonksiyonu parametreleridir.