

39612

TOPLAMSAL GAUSS GÜRÜLTÜLÜ GÖRÜNTÜLERİN
KALMAN SÜZGEÇİ İLE ONARIMI

YÜKSEK LİSANS TEZİ

Müh. Yaşar BECERİKLİ

Tezin Enstitüye Verildiği Tarih : 13 Haziran 1994
Tezin Savumıldığı Tarih : 27 Haziran 1994

Tez Danışmanı : Yrd.Doç.Dr. M. Ertuğrul ÇELEBİ
Diğer Jüri Üyeleri : Prof. Dr. Erdal PANAYIRCI
Prof. Dr. Ergül AKÇAKAYA

**F.G. YÜKSEKÖĞRETİM MURULU
DOKÜMAN YAYIN HİZMETLERİ**

HAZİRAN - 1994

ÖNSÖZ

Bu tez çalışmasında, görüntü işleme konularından biri olan görüntü onarımı Kalman süzgeci yardımıyla yapılmış ve değişik gürültü seviyelerinde performansları karşılaştırılmıştır.

Bu tezin hazırlanmasında bana değerli yardımlarını esirgemeyen sayın hocam Yrd.Doç.Dr. M.Ertuğrul Çelebi'ye ve bana yardımda bulunan tüm arkadaşlarıma teşekkür ederim.

İstanbul, Haziran 1994

Yaşar BECERİKLİ

İÇİNDEKİLER

SİMGELER.....	v
KISALTMALAR.....	vii
ÖZET.....	viii
SUMMARY.....	ix
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. GÖRÜNTÜ ONARIMI PROBLEMİ.....	4
2.1. Problemin İfadesi.....	4
2.2. Kullanılan Onarım Teknikleri.....	6
2.3. Bazı Genel Kestiriciler.....	7
BÖLÜM 3. GÖRÜNTÜLERİN MODELLENMESİ.....	10
3.1. Bazı Tanım ve Notasyonlar.....	10
3.1.1. Nedensellik Tanımı.....	11
3.2. 2-B Özbağlanımlı (AR) Modelleme.....	13
3.2.1. 2-B Özbağlanımlı (AR) Modelin Kararlılığı.....	15
3.3. 2-B Doğrusal Tahmin.....	19
3.3.1. Bir Görüntünün Özilişki Fonksiyonunun Bulunması.....	20
3.3.1.1. Ayrılabilir Özilişki Fonksiyonu....	21
3.3.1.2. Ayrılamaz Özilişki Fonksiyonu.....	22
BÖLÜM 4. 2-B ARDIŞIL NEDENSEL SÜZGEÇLEME.....	25
4.1. Boyut Problemi.....	25
4.2. 1.Dereceden 1-B'lu Ardışıl Süzgeçleme Yapısı Ve Kalman Denklemleri.....	26
4.3. 1.Dereceden 2-B'lu Ardışıl Süzgeçleme	

Yapısı Ve Kalman Denklemleri.....	32
4.3.1. Attasi Durum Uzayı Modeli.....	33
4.3.1.1. Skaler Attasi Modelin Kalman Denklemleri.....	35
4.4. Model Parametrelerinin Bulunması.....	36
4.4.1. Ayrılabilir Özilişki Fonksiyonuyla Bulma.....	37
4.4.1.1. Simülasyon Sonuçları.....	38
4.4.2. Gürültülü Gözlemden AR Parametreleri- nin Uyarlanabilir Olarak Güncelleş- tirilmesi.....	40
4.4.2.1. Simülasyon Sonuçları.....	44
BÖLÜM 5. SONUÇLAR VE ÖNERİLER.....	46
5.1. Sonuçlar.....	46
5.2. Öneriler.....	47
KAYNAKLAR.....	58
EK	
EK A.....	60
ÖZGEÇMİŞ.....	77

SİMGELER LİSTESİ

$x(m,n)$	: Durum işareti
$y(m,n)$	: Gözlem işareti
$w(m,n)$	: Beyaz giriş gürültüsü
$v(m,n)$	: Beyaz gözlem gürültüsü
σ_v^2	: Giriş gürültü varyansı
σ_v^2	: Gözlem gürültü varyansı
σ_x^2	: Durum işaretinin varyansı
σ_y^2	: Gözlem işaretinin varyansı
$\hat{\theta}$	: Özbağlanım katsayı vektörü
I	: Birim matris
$()^T$	: Devrik
G	: Süzgeç kazancı
$E()$	: Beklenen değer
$p()$	: Olasılık yoğunluk fonksiyonu
P	: Ortalama karesel hata
$\delta(i,j)$	: Kronecker sembolü
$Q(m,n)$	: Giriş gürültüsü kovaryans matrisi
$R_v(m,n)$	: Gözlem gürültüsü kovaryans matrisi
$\hat{x}$	: Kestirilen durum
$\bar{x}$	: Tahmin edilen durum
$P(m,n)$	: Süzgeç tahmin varyansı
$M(m,n)$	: Kestirim tahmin varyansı

$K(m,n)$	: Kalman kazancı
$R_x(k,1)$	: Durum işareti özilişkisi
$R_y(k,1)$	: Gözlem işareti özilişkisi
a_1, a_2	: Özbağlanım katsayıları
M	: Resim boyutu
W	: Pencere boyutu
F_1, F_2	: Özbağlanım katsayı matrisi
H	: Bulanıklık (gözlem) matrisi
$h(m,n)$	: Zamanla değişmeyen sistemin dürtü cevabı
$S(z_1, z_2)$	: Spektral yoğunluk fonksiyonu
$L(z_1, z_2)$	: Yenilik süzgeci
$\Gamma(z_1, z_2)$	: Beyazlatma süzgeci

KISALTMALAR

- SNR : "Signal to Noise Ratio". İşaret Gürültü Oranı.
- MSE : "Mean Square Error". Ortalama Karesel Hata.
- BIBO : "Bounded Input-Bounded Output". Sınırlı Giriş-Sınırlı Çıkış.
- AR : "Auto Regressive". Özbağlanım.
- MLE : "Maximum Likelihood Estimation". En Büyük Olabilirlik Tahmini.
- MAP : "Maximum a Posteriori". En Büyük Sonsal (Bayes'ci En Büyük Benzerlik).
- CM : "Conditional Mean". Şartlı Ortalama.
- NSHP : "Non Simetric Half Plane". Simetrik Olmayan Yarı Düzlem.
- UMVU : "Uniformly Minimum Variance Unbiased". Düzgün Olarak En Küçük Yansız Varyanslı.
- MMS : "Minimum Mean Square". En Küçük Karesel.
- WSS : "Wide Sense Stationary". Geniş Anlamda Durağan.

ÖZET

Bu çalışmada, görüntü işleme konularından özel öneme sahip görüntü onarımı ele alınmıştır. Görüntüler değişik parlaklık seviyelerden meydana gelir. Bu parlaklık seviyelere eklenen başka parlaklık seviyeler bozulmayı oluşturur.

Görüntü onarımı bu eklenen istenmeyen seviyelerin ortadan kaldırılmasıyla ilgilenir. Bunun için değişik metodlar geliştirilmiştir.

Bu tezde, toplamsal beyaz Gauss gürültülü görüntülerin onarımı incelenmiştir. Burada özbağımlı (AR) model kullanılmıştır. Özbağımlı (AR) katsayılar iki değişik yolla bulunmuştur. Birincisi, gürültülü tüm resim üzerinden katsayılar özilişki metoduyla bulunmuştur. İkincisi ise resim küçük pencerelere ayrılıp, her nokta için pencere kaydırılarak uyumlu bir biçimde parametreler bulunmuştur. Bu sonuçlar skaler Attasi durum uzayı modeline uygulanarak Kalman denklemleri çıkarılmıştır.

Son olarak ta, yöntemler bir resim üzerine uygulanmıştır. Değişik işaret-gürültü oranlarında denemeler yapılarak kazanç (G) ifadesine göre karşılaştırmalar yapılmıştır.

SUMMARY

RESTORATION OF IMAGES DEGRADED BY ADDITIVE GAUSSIAN NOISE BY MEANS OF KALMAN FILTERING

A fundamental problem in images analysis is restoration. Images consist of lots of gray levels.

The model of the image formation system for blurred and noisy image is

$$y(m,n) = \sum_{kl} h(k,l)x(m-k,n-l) + v(m,n) \quad (1)$$

where $h(k,l)$ is the spatial operator (point spread function (PSF)) representing the blurring effect which may be caused by such phenomena as atmospheric turbulence, relative motion between the camera and the object being photographed. $x(m,n)$ represents the uncorrupted image of size $M \times M$. $y(m,n)$ is the observed image. And $v(m,n)$ represents the noise in the digitized image which may be due to transmission media, quantization effects, etc. The noise is a stochastic phenomenon. Generally, it may be considered to be white Gaussian.

(1) can be written in the vector form as follow,

$$y = Hx + v \quad (2)$$

where y is observation vector, H is blurring matrix, x is original image vector and v is additive white noise. In this thesis, it is assumed no blurring effects.

The main aim of the image restoration is to estimate $x(m,n)$ from the observed image $y(m,n)$ given the PSF

and some statistical information of the noise.

It is used several approach in the image restoration. One of them is minimum mean square (MMS) method which is well-known. This method is implemented to two types approaches. The first method is based on batch processing. This method is processed all images. Also, images are had lots of data. Therefore, too many calculations are necessary in this method. And, it is not suitable to process in the real time. The second one is based on recursive processing which is generally used in Kalman filtering. In this thesis, recursive image restoration techniques based on Kalman approach has been used. Kalman filtering technique has a lot of advantages. For instance, memory problem is alleviated. And there are no many calculations due to recursivity. We can say that Kalman filtering technique is a very efficient solution for image restoration.

An image is generally characterized by its autocorrelation function. Autocorrelation function of image can be generally factorized. This factorization of it into causal and noncausal parts results in innovation and whitening filters, respectively. These filters are used for 2-D image modelling. An important model is autoregressive model.

Autoregressive (AR) model parameters are found from autocorrelation function. First order AR scalar image model corresponding to separable and exponential autocorrelation function is as follow,

$$\begin{aligned} x(m,n) = & a_1x(m-1,n) + a_2x(m,n-1) - a_1a_2x(m-1,n-1) \\ & + w(m,n) \end{aligned} \quad (3)$$

where a_1 and a_2 are the correlation of adjacent points in the image in horizontal and vertical dimensions, respectively. $w(m,n)$ is white Gaussian noise with zero mean and with, variance σ^2 . Furthermore the following corruption model is assumed,

$$y(m,n) = x(m,n) + v(m,n) \quad (4)$$

where $v(m,n)$ is a white noise. Note that no blurring effect is assumed.

Estimation of AR parameters is very important in image processing. In this thesis, autocorrelation method is used for the estimation of AR parameters. Autocorrelation function is estimated from the noisy image. Wide-sense stationary (WSS) autocorrelation function of images is described as follow,

$$R_x(k,l)=E\{x(m,n)x(m+k,n+l)\} \quad (5)$$

where $E\{.\}$ denotes mathematical expectation. Mean of the image is as follow,

$$m_x=E\{x(m,n)\} \quad (6)$$

where m_x is a constant due to wide sense stationarity.

Instead of expectation, sample autocorrelation and sample mean are generally used. They are,

$$\hat{R}_x(k,l)=1/(MM)\sum_{m=0}^{M-1-k}\sum_{n=0}^{M-1-l}x(m,n)x(m+k,n+l) \quad (7)$$

$$\hat{m}_x=1/(MM)\sum_{m=0}^M\sum_{n=0}^Mx(m,n) \quad (8)$$

Then, AR parameters are calculated as follows,

$$a_1=R_x(1,0)/R_x(0,0) \quad (9)$$

$$a_2=R_x(0,1)/R_x(0,0) \quad (10)$$

Also, the variance of the driving noise can be found as,

$$q=(1-a_1^2)(1-a_2^2)R_x(0,0) \quad (11)$$

Finally, the parameters are estimated from degraded image as follows,

$$a_1 = R_y(1,0) / (R_y(0,0) - \sigma_v^2) \quad (12)$$

$$a_2 = R_y(0,1) / (R_y(0,0) - \sigma_v^2) \quad (13)$$

$$q = (1 - a_1^2)(1 - a_2^2)(R_y(0,0) - \sigma_v^2) \quad (14)$$

where $R_y(.)$ is autocorrelation function of degraded image, σ_v^2 is variance of observation noise.

Second method of AR parameter estimation is adaptive estimation. This method is given as follows,

$$x(m,n) = \sum_{kl \in D} a_{kl}(m,n)x(m-k,n-l) + w(m,n) \quad (15)$$

$$\hat{\Theta}(m,n) = \left[\sum_W y_1^T(m,n) y_1(m,n) - N I \sigma_v^2 \right]^{-1} \left[\sum_W y_1^T(m,n) y(m,n) \right] \quad (16)$$

where,

$$y_1(m,n) = \{y(m-L,n-L), \dots, y(m-L,n+L), \dots, y(m,n-1)\}$$

$$\Theta = \{a_{-L-L}, a_{-LL}, a_{-L+1-L}, \dots, a_{0-L}, \dots, a_{0-1}\}$$

and N is the number of observation in the window W . D is the $L \times L$ model support region.

This method is given very efficient solution for estimation of AR parameters. In this method, W window is an important size. The size of W must be small enough for the homogeneity assumption to hold and yet large enough to be able to sufficiently reduce the effects of any noise.

Both methods are incorporated into Kalman filtering equations. Kalman filtering equations are found in

vector form and scalar form. Based on the state space representation of the model Kalman filter for the observation state is designed which yields the best estimates for the both methods. The equations of the Kalman filter are given as follows,

First, the state and observation equations are,

$$x(m,n)=a_1 x(m-1,n)+a_2 x(m,n-1)-a_1 a_2 x(m-1,n-1) \quad (17)$$

$$y(m,n)=x(m,n)+v(m,n) \quad (18)$$

Prediction (time update) equations,

$$\bar{x}(m,n)=a_1 x(m-1,n)+a_2 x(m,n-1)-a_1 a_2 x(m-1,n-1) \quad (19)$$

$$M(m,n)=a_1^2 P(m-1,n)+a_2^2 P(m,n-1)-a_1 a_2 P(m-1,n-1)+q \quad (20)$$

where q is driving noise variance.

Filtering (measurement update) equation,

$$K(m,n)=M(m,n)[M(m,n)+r]^{-1} \quad (21)$$

$$\hat{x}(m,n)=\bar{x}(m,n)+K(m,n)[y(m,n)-\bar{x}(m,n)] \quad (22)$$

$$P(m,n)=M(m,n) r [M(m,n)+r]^{-1} \quad (23)$$

where r is observation noise variance.

All those equation are applied to LENA image step-by-step and the results are compared each other. For making comparision, input SNR and output SNR are used. And defining the filter gain (G), improvement is observed between input and output.

As a result, we can say that the second method is very efficient than the other. The second method updates the parameters and the filter predicts the

original data from noisy image.

If the input SNR is too low then the filter performance is to be bad. Therefore, in the low SNR, this method must be improved which may be second order modelling, or another parameter estimation. But; high order modelling is required too many calculations.



BÖLÜM 1. GİRİŞ

İşaret işlemede ulaşılan gelişmeler, son 15-20 yıldan bu yana görüntü işlemeye de başarıyla uygulanmaktadır. Sayısal görüntü işlemedeki bu gelişmeler, bir çok uygulama alanına girmiştir. Özellikle, tıp, haberleşme, jeoloji, nükleer enerji mühendisliği, uzay mühendisliği gibi.

Bu kadar geniş uygulama alanına sahip olan bu konu bir çok alt konulara ayrılmıştır. Bu konuların her biri derin araştırmaları gerektirmektedir. Görüntü iyileştirme, görüntü kodlama, kenar sezme, parametre belirleme, görüntü onarımı gibi.

Bu tez çalışmasında sayısal görüntü işleme konularından "görüntü onarımı" ile ilgilenilecektir. Görüntü onarımı, yukarıda bahsedilen konular içinde önemli bir yer tutmaktadır.

Elimizde bulunan görüntü, çeşitli olaylardan dolayı bozulmuş olabilir. Görüntü onarımı şu şekilde tanımlanabilir: Çeşitli nedenlerle bozulmuş resmi orjinal hâle getirmeye "görüntü onarımı" denir.

Görüntü onarımı konusunun önemi, kendini her sahada göstermesiyle belli eder. Bir uzay çalışması düşünersek, milyonlarca kilometre uzaktan uyduların veya uzay

araçlarının gönderdikleri fotoğraflar ne kadar net olabilir ? Mutlaka uzaydan gelişi sırasında çeşitli etkilerin sonucunda bozulmaya uğrayacaktır. Gelen resim en kısa zamanda temizlenerek inceleme altına alınması, net hale getirilmesi gerekmektedir. İşte devreye buradahiemen görüntü onarımı girmektedir. Amacımız, işareti ve bozulmayı modelleyip, bilinen verilerden yararlanarak onarımı gerçekleştirmektir.

Görüntü işareti çok fazla veri taşır. Bu yüzden çok fazla işlem gerektiren hesaplamalar her zaman problem olur. Bilgisayar uygulamalarında bu problem işlem zamanının çok sürmesi ve çok fazla bellek gereksinimi olarak karşımıza çıkar. Bundan dolayı bu külfetlerden kurtulma yolları aranmaktadır.

Görüntü onarımında bir çok yaklaşımlar kullanılır. En iyi bilinen yöntemlerden biri en küçük kareler yöntemi ile kestirimdir. Bu yöntemin uygulanmasına iki türlü yaklaşılr: Birincisi, yığın işlemeye dayanan metoddur. Bu metod tüm resmi ele alarak işlem yapmaya dayanır. Zaten resim fazla veri taşımaktadır. Sonuçta çok fazla işlem gerektiren hesaplamalara girilecektir. İkinci yöntem, hesapmalardaki fazlalığı azaltan bir metoddur. Kalman sügeci bu yönteme göre, ardışıl olarak fazla hesapları azaltmaktadır. Bu konu Bölüm 2'de anlatılmıştır.

Tez çalışması Kalman süzgeçleme tekniği kullanarak görüntü onarımını içermektedir. Bir görüntü işaretinin matematiksel modeli ele alınarak işe başlanılmıştır. Bu konu Bölüm 3'te incelenmiştir. Bu bölüm, ayrıca özilişki fonksiyonu, varyans ve tezde kullanılan notasyonları

içermektedir.

Modellemelerden yararlanarak istenilen süzgeç yardımıyla onarım işlemi yapılır. Bölüm 2'de anlatılan nedenlerden dolayı tezde Kalman süzgeci tercih edilmiştir. Bölüm 4, iki boyutlu (2-B) ardışıl süzgeç yapısı ve 2-B Kalman süzgeç denklemlerini içermektedir. Ayrıca bu bölümde süzgeç model parametrelerinin bulunmasından bahsedilmiştir. Burada, model parametrelerinin kestirilmesinde iki temel yapı üzerinde durulmuştur. Biri tüm resim için, özilişki fonksiyonu yardımıyla yatay ve düşey özbağlanımlı (AR) katsayıları bulunması, diğeri hızlı ve uyumlu olarak AR katsayıları bulunmasıdır. Burada katsayılar gürültülü resim üzerinden hesaplanmaktadır. Ayrıca simülasyon sonuçları da verilmiştir. Simülasyon sonuçları karşılaştırılmıştır. Karşılaştırmalar giriş işaret-gürültü oranı (SNR_i) ile çıkış işaret-gürültü oranı olan (SNR_o) arasında yapılmaktadır. Ayrıca kazanç (G) olarak bir tanım yapılmış, bu da karşılaştırma faktörü olarak kullanılmıştır. İkinci metoddaki pencere boyutunun sonuçlara etkisi de incelenerek karşılaştırılmıştır.

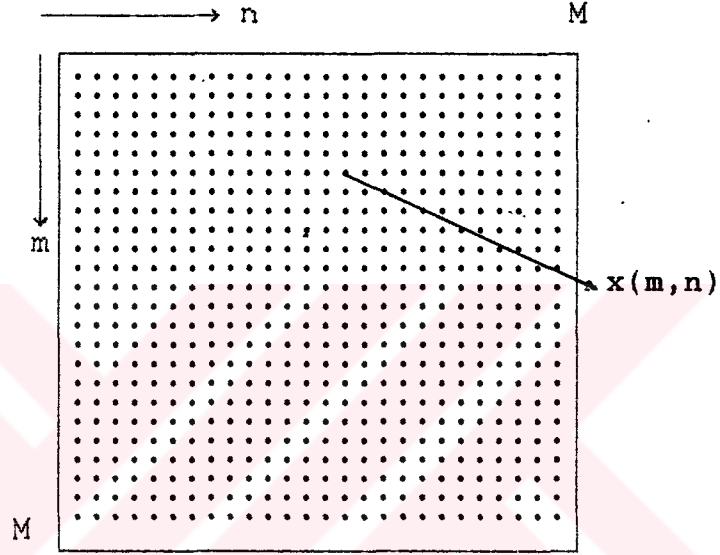
BÖLÜM 2. GÖRÜNTÜ ONARIMI PROBLEMİ

2.1. Problemin İfadesi

Görüntü onarımı, görüntü işleme konularının başlıcalarından biridir. Bozulmuş olan bir resim onarılmalıdır. O halde bozulmuş resmi modellememiz gerekmektedir. Bozulma, iki temel bileşenden meydana gelir ki biri bulanıklık diğeri ise gürültüdür. İkisi birlikte bozulmayı oluşturur. Bulanıklığa atmosferik etkilerden dolayı oluşan gürültü, fotoğraf çekilirken nesne ve kameranın göreceli farkı gibi sebepler neden olur. Gürültüye ise, kuantalama hataları, yayılım ortamı, kaydetme ortamı, iyi olmayan ölçümler neden olabilir [1].

Sayısal olarak bir görüntü, her bir noktaya bir parlaklık seviyesi karşı düşen bir fonksiyonla ifade edilebilir. Tezde sayısal görüntü $x(m,n)$ ile ifade edilecektir. Her (m,n) noktası farklı bir değer alacaktır. O halde $x(m,n)$ rasgele bir süreç ile temsil edilebilir. Yani, fonksiyonu istatistiksel bir modelle modelleyebilmek mümkün olmaktadır. Siyah beyaz resim düşünülürse, her noktaya farklı bir gri seviye karşı düşecektir. Örneğin bu gri seviyeler 8-bit olarak alınırsa her nokta 256 gri seviyeye karşı gelecektir. Şekil 2.1'de $(M \times M)$ boyutlu resmin yatay ve dikey yönleri gösterilmiştir. Genellikle, resmin boyutu 2'nin kuvveti olarak seçilir. Bu hesap kolaylığı ve anlaşılabilirlik sağlar. Ancak

farklı da seçilebilir. Ayrıca yatay ve düşey miktarlar eşit olmayadabilir. Burada, ilgilenilen asıl konu bir resmin nasıl modelleneyeceği konusudur. Bir resim Gauss Markov modeli ile tanımlanabilir. Bu konu Bölüm 3'te açıklanmıştır. Bir de bozulmuş resmin modellenmesi



Şekil 2.1. Bir resmin yatay ve düşey yönlerinin gösterilmesi.

konusu vardır. Bozulmanın matematiksel ifadesini şöyle yazabiliriz [2].

$$y=Hx+v \quad (2.1)$$

Burada y gözlenen görüntü vektörünü, H bulanıklık matrisini, x orjinal görüntü vektörünü, v ise toplamsal gürültü vektörünü göstermektedir.

Problem, bozulmuş gözlem ifadesinden yararlanarak orjinal resmi bulabilmektir. Problemi çözerken mümkün olduğu kadar var olan ön bilgileri kullanmak gerekir. Bu ön bilgiler orjinal resmin istatistiksel özellikleri, bulanıklık ve gürültü hakkında olan istatistiksel ve tanıttıcı bilgilerdir. Bu bilgiler ve varsayımlar ile ilgili bilgiler sonraki konularda verilecektir.

2.2. Kullanılan Onarım Teknikleri

Görüntü onarımı için klasik metodlardan başlayarak bir çok yöntemler geliştirilmiştir. Önceden söylenildiği gibi bir görüntü çok fazla veri taşımaktadır. O halde, sonuçta karşılaşılan ikinci önemli problem çok fazla hesap gerektiren işlemlerdir.

Yığın işlemeye dayanan metod tüm resmi ele alarak çözüme ulaşmaya çalışır. Yani bir anda tüm resim işlenmeye çalışılır. Örneğin, en küçük ortalama kareler yöntemine göre (MMS) kestirm şu şekilde verilebilir [2].

$$H^T R^{-1} (y - H\hat{x}) - M^{-1} (\hat{x} - \bar{x}) = 0 \quad (2.2)$$

Burada, M işaretin varyansı, R toplamsal gürültü varyansı, H bulanıklık matrisi, $\hat{x}$ kestirilen işaret ve y ise bozulmuş resimdir. Burada kestirimin bulunabilmesi için $M^2 \times M^2$ adet sistem denklemini çözmek gerekir. Bu fazla zaman alan işlemlerdir. Aynı işlevi ardışıl olarak yapan yöntem Kalman süzgecidir. Diğer yöntemde bellek problemi de vardır. Hesaplamalar yapılırken aktif bellek alanı bulundurma zorunluluğu vardır. Kalman

teknikinde ise bu problemler büyük ölçüde ortadan kalkmaktadır. Kalman süzgecinin bir çok avantajları vardır. Bunları şöyle anlatabiliriz [2,3].

Kalman süzgeci büyük ölçüde bellek problemini halletmiş durumdadır. Çünkü, burada tüm veriler bir anda kullanılmamaktadır. O an hangi noktanın onarımı yapılacaksa o noktadan önceki veriler kullanılmaktadır. Yani kısmi kestirimlerde bulunmaktadır. Bu ise süzgece bir esneklik vermektedir. Burada bir yaklaşıklık yapılarak sonuca ulaşılmaya çalışılmaktadır. Bu yaklaşıklığın temeli, iki uzak nokta arasındaki karşı ilişkinin göz ardı edilmesi esasıdır. [2]'de Kalman eşitliklerinin bulunmasında bu yaklaşıklık verilmiştir. Sonuçta hesapların düştüğü görülmektedir. Ayrıca Kalman süzgeçleme ile gerçek zamanda işlem uygun olmaktadır.

Bu anlatılanların sonucu olarak tezde Kalman ile ilgili görüntü onarımı ele alınmış ve uygulamaları yapılmıştır. Kalman süzgeci ile görüntü onarımı konusu ileride tartışılacaktır. Şimdi de genel kestiricilerden söz edelim.

2.3. Bazı Genel Kestiriciler

Genel olarak olabilirlik ilkesine dayalı olarak kestiriciler düşünülür. En büyük benzerlik (MLE) ve Bayes 'ci en büyük olabilirlik (maximum a Posteriori-MAP) kestiriciler bunlardan önemlileridir.

En büyük olabilirlik tahmin edicisi (MLE) olabilirlik fonksiyonu olarak tanımlanan $p(x/\theta)$ 'yi θ 'ye göre

en büyük yapmaya çalışır. Yani,

$$\max_{\theta} p(x^n/\theta) \longrightarrow \theta_{ML}(x^n) \quad (2.3)$$

Burada, θ rastgele olmayan bir bilinmeyen parametredir. x^n ise n boyutlu bir örnek vektördür. $x^n \equiv (x_1, x_2, \dots, x_n)$ olarak gösterilebilir. MLE asimtotik olarak Cramer-Rao sınırına yaklaşır. Eğer dağılım fonksiyonu üstel bir yapıya sahipse MLE en azından yeterli bir istatistiktir. Eğer MLE 'nin çözümü (2.3)'deki gibi verilirse ve θ 'nin bir yansız tahmin edicisi varsa, o zaman en büyük tahmin edici (MLE) düzgün olarak en küçük varyanslı yansız (UMVU) tahmin edicidir. Ve θ 'nin varyansı Cramer-Rao sınırına karşı gelir [2].

$$\left. \frac{\partial}{\partial \theta} \log p(x/\theta) \right|_{\theta=\hat{\theta}} = 0 \quad (2.4)$$

MAP kestiriciler, sonsal yoğunluğu ($p(\theta/x^n)$) en büyük yaparak bulunur. Yani,

$$\max_{\theta} p(\theta/x^n) \longrightarrow \theta_{MAP}(x^n) \quad (2.5)$$

Burada Bayes kuralını uygularsak,

$$p(\theta/x) = p(x/\theta) p(\theta)/p(x) \quad (2.6)$$

(2.5) eşitliğinden de görüleceği gibi paydadaki fonksiyon $p(x)$ normalize edilebilir. Çünkü en büyükleme θ 'ye

göre yapılmaktadır. Sonuç olarak sonsal yoğunluk fonksiyonu, önsel bilgi $p(\theta)$ ile olabilirlik fonksiyonunun çarpımıdır.

Bir de şartlı ortalama (CM) kestiricisi vardır ki şöyle ifade edilir,

$$\hat{\theta}_{CM} = E[\theta/x] = \int \theta p(\theta/x) d\theta \quad (2.7)$$

Burada $E[.]$ beklenti işlemcisidir. Eğer, θ ve x Gauss dağılımıyla veriliyorsa,

$$\hat{\theta}_{CM} = \hat{\theta}_{MAP} \quad (2.8)$$

olur.

BÖLÜM 3. GÖRÜNTÜLERİN MODELLENMESİ

Modelleme bir fiziksel olayın matematiksel ifadesinin bulunmasıdır. Modellemenin iyi olması gerçeğe yakınlığı ile ilgilidir. Tam olarak gerçek bir modelleme yapabilmek çok zordur. Ancak, çoğu yerde tam olarak bir modelleme yapmak gerekmez. Yeteri kadar iyi yaklaşıklık modeller yapılabilir. Görüntü modellemede de bunlar geçerlidir. Görüntü onarımı da bu modellemeler yardımıyla yapılmaktadır. Daha önceden bilinmesi veya belirlenmesi gereken istatistiksel parametreler, matematiksel modellerin yardımıyla çok daha iyi anlaşılacaktır.

3.1. Bazı Tanım ve Notasyonlar

Konuya girmeden önce bazı tanımlamaları verelim. İki boyutlu (2-B) ayrık görüntü işareti Bölüm 2'de gösterildiği gibi $x(m,n)$ ile gösterilecektir. Burada, $x(m,n)$ (m,n) noktası bir parlaklık seviyesine karşı gelir. $x(m,n)$, 2-B rasgele bir süreç olarak düşünülecektir. Rasgele bir alana sahip olan $x(m,n)$ işaretinin ortalamasını $m_x(k,l)$ ve özilişkisini ise $R_x(k,l;i,j)$ ile gösterilirse, aşağıdaki eşitlikler yazılır.

$$m_x(k,l) = E \{ x(k,l) \} \quad (3.1)$$

$$R_x(k,l;i,j) = E \{ x(k,l)x(i,j) \} \quad (3.2)$$

Burada $E \{.\}$ matematiksel beklentidir.

Durağanlık varsayımı çok kullanılan ve görüntülere uygulanabilen bir yaklaşıklıktır. Eğer burada geniş anlamda durağanlık kullanılırsa, işaret ortalaması sabit bir değer olur. Özilişki ise uzay (düzen,zaman) kaymalarından bağımsız olur.

$$m_x(k,l) = m_x \quad (3.3)$$

$$R_x(p,r) = E \{ x(k,l)x(k+p,l+r) \} \quad (3.4)$$

Ayrıca ileride kullanılacak olan Gauss-Markov olma durumunda, ortalama ve özilişki fonksiyonları rasgele süreci tamamen karakterize etmek için yeterlidir.

3.1.1. Nedensellik Tanımı

Nedensellik, alışılmış ve çok kullanılan bir ifadedir. Şimdi bir resmin raster tarandığını düşünelim. Resim soldan sağa doğru Şekil 3.1'deki gibi taransın. Tarama birinci satırın başından başlayarak sonuna kadar devam edip, bir sonraki satırın başına gelmektedir.

Şimdi de Şekil 3.1'den yola çıkarak bir tanımlama yapılabilir. Resmin herhangi bir (m,n) noktasında bulunulsun. O anda bulunulan noktaya şimdiki durum, o noktanın evveline geçmiş ve sonrasına ise gelecek denilmektedir. Bu durum da Şekil 3.2'de gösterilmiştir. Matematiksel olarak bu durumlar yazılabilir. Şöyleki,

Geçmiş $(m,n) = \{ (i,j) : i=m, 0 \leq j \leq n-1 \}$

$$\cup \{ (i,j) : 0 \leq i \leq m-1, 0 \leq j \leq M-1 \}$$

(3.5)

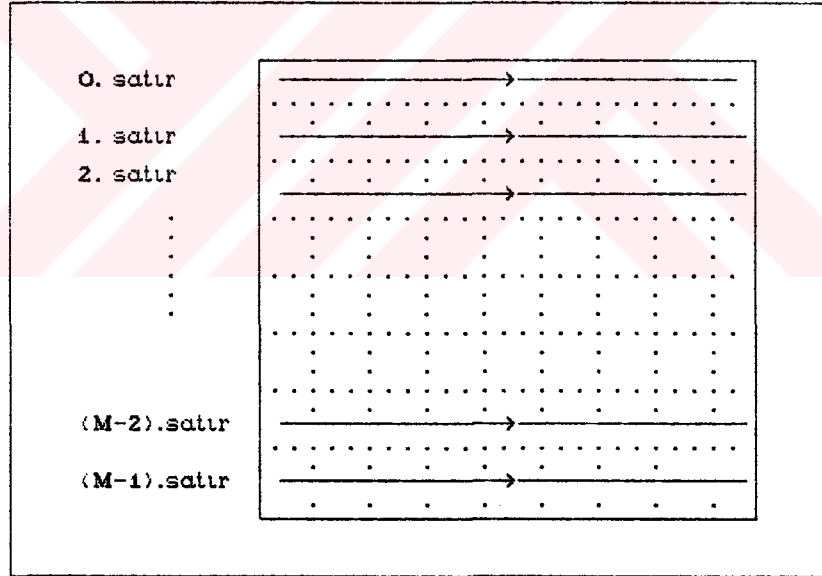
Şimdi $(m,n) = \{ (i,j) : i=m, j=n \}$

(3.6)

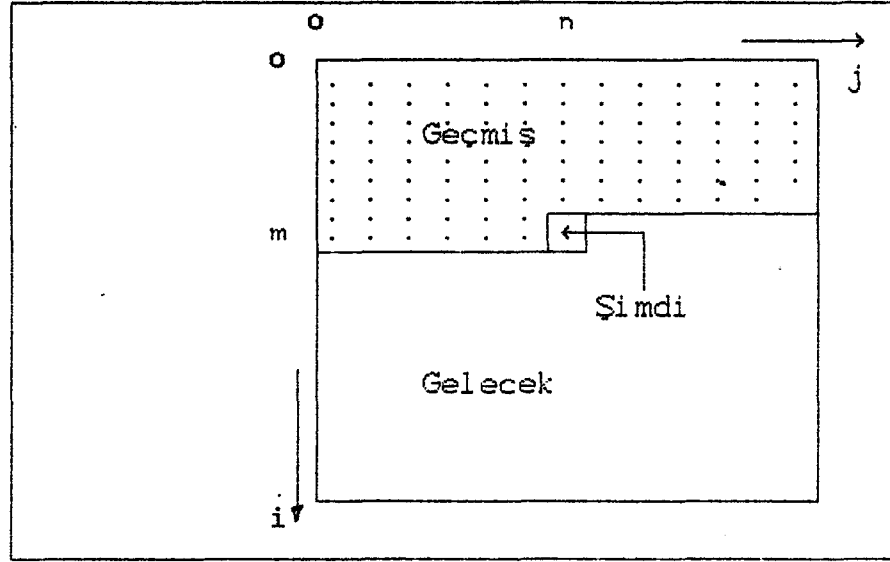
gelecek $(m,n) = \{ (i,j) : i=m, n+1 \leq j \leq M-1 \}$

$$\cup \{ (i,j) : m+1 \leq i \leq M-1, 0 \leq j \leq M-1 \}$$

(3.7)



Şekil 3.1. Bir resmin tarama işlemi



Şekil 3.2. Bir resmin geçmiş, şimdi ve gelecek bölgele-
rinin gösterilmesi

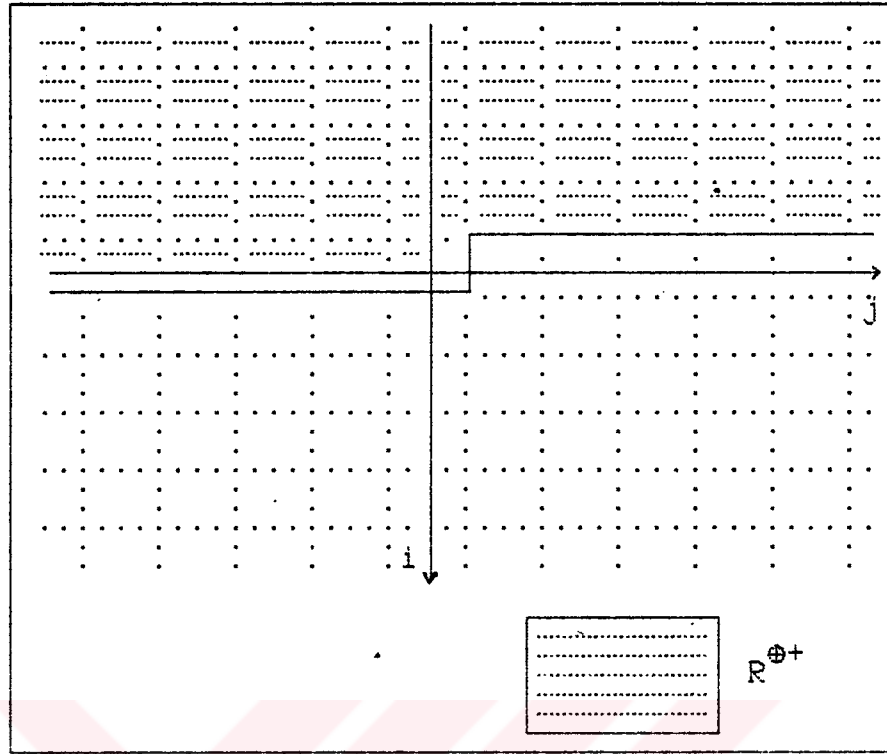
Şekil 3.2'de gösterilen bölge eğer çok büyük bir bölgeyse, geçmiş bölgeye, simetrik olmayan yarı düzlem (NSHP) adı verilir ve $R^{\oplus}$ ile gösterilir. Şimdi ve geçmiş bölgeler birleştirilirse bu bölge de $R^{\oplus+}$ ile gösterilir [3,4]. Bu durumlar Şekil 3.3'de gösterilmiştir. Özet olarak bu bölgeler aşağıdaki biçimde de yazılabilir.

$$R^{\oplus} = \text{Geçmiş } \{0,0\} \quad (3.8)$$

$$R^{\oplus+} = \text{Geçmiş } \{0,0\} + \text{Şimdi } \{0,0\} \quad (3.9)$$

3.2. 2-B Özbağımlı (AR) Modelleme

İki boyutlu (2-B) AR modelleme resim için uygun



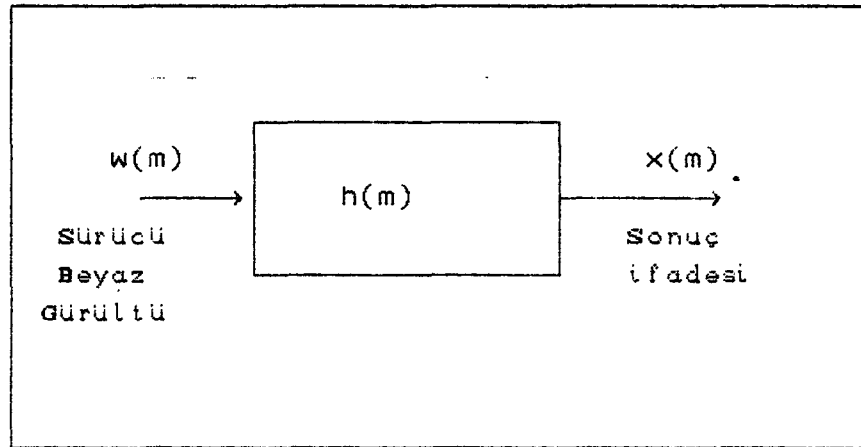
Şekil 3.3. R^{++} bölgesinin gösterimi

bir yapıdır. Önce kısaca bir boyutlu (1-B) özbağımlı (AR) modelleme hatırlanırsa,

$$x(m) = \sum_{k=0}^M a(k)x(m-k) + w(m) \quad (3.10)$$

Burada M sistemin derecesini, $w(m)$ sıfır ortalamalı σ_w^2 varyanslı beyaz sürücü gürültüyü ifade eder. Eğer sürü-
gürültü Gauss ise, $x(m)$ 'de Gauss olacaktır. Çünkü

sistem doğrusaldır. Örnek olarak, eğer sistemin derecesi $M=1$ ise $\sigma_w^2 = (1-a^2)\sigma_x^2$ olur. Şekil 3.4 'de durum gösterilmiştir. Bu konu [5] 'de detaylı bulunabilir.



Şekil 3.4. 1-B Beyaz Sürücü Gürültü Modeli

2-B özbağımlı modellemeye benzer şekilde yapılabilir. Yani,

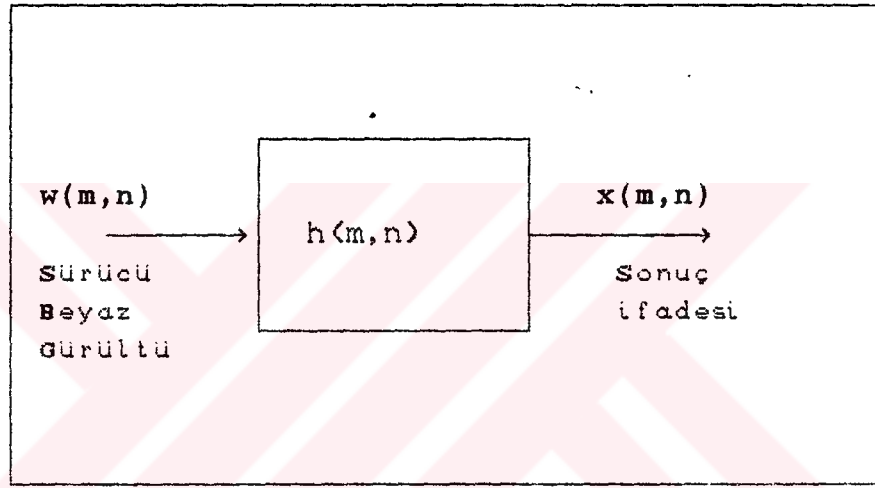
$$x(m,n) = \sum_{k,l \in D} a(k,l)x(m-k,n-l) + w(m,n) \quad (3.11)$$

Burada $a(k,l)$ özbağımlı (AR) katsayılarıdır. D destek bölgesini göstermektedir. $D \subset \mathbb{R}^{\oplus}$ ilişkisi vardır. Yani, nedensellik ilişkisi vardır. Yine burada sürücü gürültü Gauss ise $x(m,n)$ de Gauss olacaktır. Bu durum Şekil 3.5 'de gösterilmiştir. 2-B bir rasgele süreç, beyaz sürücü gürültülü doğrusal model olarak, özilişki fonksiyonuyla ifade edilebilir [4]. $x(m,n)$ geniş anlamda durağan ise (3.4) 'e göre özilişkisi $R_x(k,l)$ olur.

3.2.1. 2-B Özbağımlı (AR) Modelin Kararlılığı

2-B bir sistemin kararlı olması, dikkat edilmesi

gereken en önemli konudur. 2-B özbağımlı (AR) model, sınırlı giriş-giriş sınırlı-çıkış (BIBO) özelliğini sağlaması durumunda model kararlı olur. Çünkü, bir girişe karşılık çıkıştan sınırlı bir işaret elde etmek, sistemin birim dürtü cevabının mutlak toplanabilir olmasını sağlar. Zaten, birim dürtü cevabı mutlak toplanabilir ise, o sistem kararlıdır. Aksi durumda küçük hatalar



Şekil 3.5. 2-B Beyaz Sürücü Gürültü Modeli

sonuçta çok büyük hatalara neden olacaktır.

Kısaca söylemek gerekirse, doğrusal zamanla değişmeyen (LSI) bir sistemde kararlılık için yeter ve gerek şart sistem cevabı $h(m,n)$ 'nin toplanabilir olmasıdır. Yani,

$$\sum_{m,n=-\infty}^{\infty} |h(m,n)| < \infty \quad (3.12)$$

olmalıdır.

Geniş anlamda durağan bir görüntü işareti $x(m,n)$ düşünülün. Bunun özilişkisi (3.4) 'e göre $R_x(k,l)$ idi. Eğer $R_x(k,l)$ 'nin z-dönüşüğü alınırsa, spektral yoğunluk fonksiyonu tanımı elde edilir.

$$S_x(z_1, z_2) = \sum_{k,l} R_x(k,l) z_1^{-k} z_2^{-l} \quad (3.13)$$

Eğer her w_1 ve w_2 için $S_x(e^{jw_1}, e^{jw_2}) \geq 0$ şartını

sağlıyorsa aşağıdaki eşitlik yazılabilir [6].

$$S_z(z_1, z_2) = \frac{\sigma_v^2}{A(z_1, z_2) A(z_1^{-1}, z_2^{-1})} \quad (3.14)$$

Burada,

$$A(z_1, z_2) = 1 - \sum_{k,l \in D} a(k,l) z_1^{-k} z_2^{-l} \quad (3.15)$$

D bölgesi simetrik olmayan yarı düzlem (NSHP) ve genellikle sonsuz uzunluktadır. $A^{-1}(z_1, z_2)$ 'nin kutupları birim dairenin içinde olursa sistem olarak $A^{-1}(z_1, z_2)$ kararlıdır. Ve yenilik süzgeci olarak isimlendirilir. Özel olarak $L(z_1, z_2)$ ile gösterilirse [4],

$$L(z_1, z_2) = A^{-1}(z_1, z_2) \quad (3.16)$$

$L(z_1, z_2)$ 'nin tersini alarak çok kullanılan ve adına beyazlatma süzgeci denilen ve $\Gamma(z_1, z_2)$ ile gösterilen ifade elde edilir.

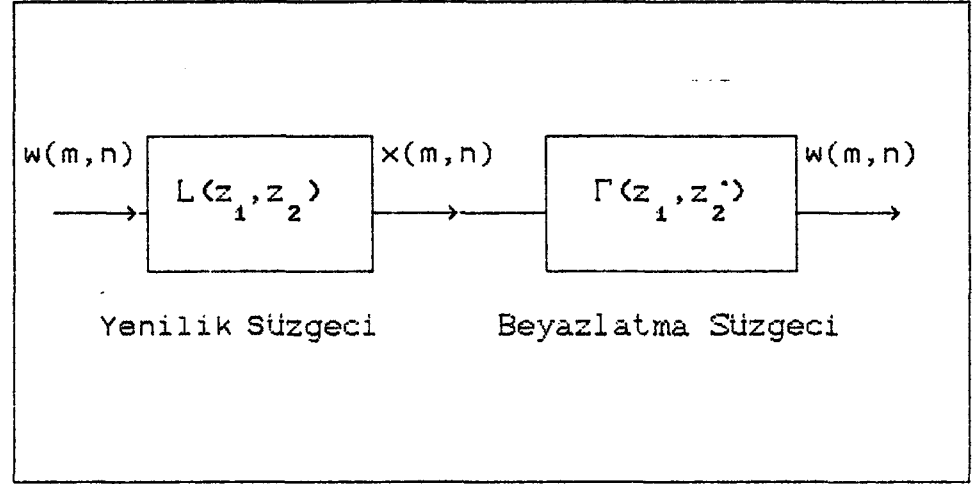
$$\Gamma(z_1, z_2) = L^{-1}(z_1, z_2) \quad (3.17)$$

Ayrıca,

$$\Gamma(z_1, z_2) L(z_1, z_2) = 1 \quad (3.18)$$

Bu anlatılan durumlar Şekil 3.6 'da gösterilmiştir.

Beyazlatma ve yenilik süzgeçleri görüntü işlemede önemli yere sahiptir. Yukarıda anlatılan işlemler öz ilişki veya buna denk olan spektral yoğunluk fonksiyonuyla yapılabilir. Bu süzgeçler, spektral yoğunluk fonksiyonunun ayrılmasıyla da elde edilebilir. Ancak bu ayırma işlemi bir boyutta (1-B) olduğu kadar kolay değildir.



Şekil 3.6. Beyazlatma ve Yenilik Süzgeçlerinin Gösterimi

İki boyutlu (2-B) ifadeler genelde ayrılabilirler [7]. Bu nedenle başka yöntemlere başvurulmuştur. Diğer bölümde kısaca bunlardan bahsedilecektir.

3.3. 2-B Doğrusal Tahmin

Daha önce verilen (3.11) eşitliği, $x(m,n)$ 'yi üreten 2-B bir giriş çıkış ilişkisini gösteren rasgele bir süreçtir. Şimdi, bir kestirici düşünülecektir ki, destek bölgesi D' olup D 'nin bir alt kümesidir. (3.11)'ye göre kestirici şöyle görülebilir,

$$\hat{x}(m,n) = \sum_{k,l \in D'} a'(k,l) x(m-k,n-l) \quad (3.19)$$

Buradan kestirim hatası yazılırsa,

$$e(m,n) = x(m,n) - \hat{x}(m,n) \quad (3.20)$$

Ve ortalama kare kestirim hatası,

$$P = E \{ e^2(m,n) \} = E \{ (x(m,n) - \hat{x}(m,n))^2 \} \quad (3.21)$$

(3.21) ile verilen ifadeyi minimum yapacak $a'(k,l)$ katsayıları için düzenlendiğinde, Normal denklemler elde edilir. Normal denklemlerinin çözümüyle kestirim katsayıları için,

$$R_x(k,l) = \sum_{i,j \in D'} a'(i,j) R_x(k-i, l-j) \quad \forall k,l \in D' \quad (3.22)$$

sonucu kolayca bulunabilir. D' bölgesi, çözülecek denklem sayısını belirlediğinden seçilmesi önem gerektirir. Ayırışabilir özilişki fonksiyonu olma durumunda, doğrusal denklemler Toeplitz yapıya dönüştürülerek çözüm daha hızlı yapılmaktadır [4].

Eğer, D' destek bölgesi D 'ye eşit seçilirse beyaz gürültü modeliyle en küçük varyans ifadelendirmesi birbirlerine denk olur. Ancak, D' genelde küçük seçilerek, denklem sayısı azaltılır. Bunun sonucunda, D' destek bölgesi yenilik süzgeci tarafından üretilen süreç, arzu edilen özilişki $R_x(k,l)$ 'ye sahip olamayacaktır.

3.3.1. Bir Görüntünün Özilişki Fonksiyonunun Bulunması

Bir görüntü modellenirken derecesine göre modellenir. Genelde 1.derece modelleme yeterli olabilmektedir.

Tezde 1.derece modelleme kullanılmıştır. 1.derece modellemede, resmin iki noktası arasındaki ilişki, aralarında uzaklık arttıkça üstel olarak azalır. Bu durum bize bazı kolaylıklar sağlayacaktır. İhmal edilebilecek durumlar ortaya çıkacaktır. Bu durum bir sonraki bölümde Kalman denklemleri verilirken anlatılacaktır.

3.3.1.1. Ayrılabilir Özilişki Fonsiyonu

İki boyutlu özilişki fonksiyonu bir boyutlu özilişki fonksiyonuyla ifade edilebilir. Burada resmin durağan olması varsayımı yapılmıştır. (3.11) ile verilen ifade 1.derece için,

$$x(m,n) = a_1x(m-1,n) + a_2x(m,n-1) - a_1a_2x(m-1,n-1) + w(m,n) \quad (3.23)$$

Bu model Habibi 'nin [8] 'deki modeline benzer. Buna göre özilişki şu şekilde verilir [2].

$$R_x(k,l) = \sigma_x^2 a_1^{|i|} a_2^{|j|} \quad |a_1|, |a_2| < 1 \quad (3.24)$$

Burada σ_x^2 resmin varyansını göstermektedir. a_1 ve a_2 sırayla yatay ve düşey ilişki katsayılarıdır. Bu katsayılar resimden resime değişir. Şekil 3.7 'de $a_1=a_2=0.85$ için özilişki çizilmiştir. Görüldüğü gibi ayrılabilir model dairesel simetriye sahip değildir.

Ayrılabilir model için, ardışıl tahmin edici

aşadaki formda yazılabilir [9].

$$\hat{x}(m,n)=a_1x(m-1,n)+a_2x(m,n-1)-a_1a_2x(m-1,n-1) \quad (3.25)$$

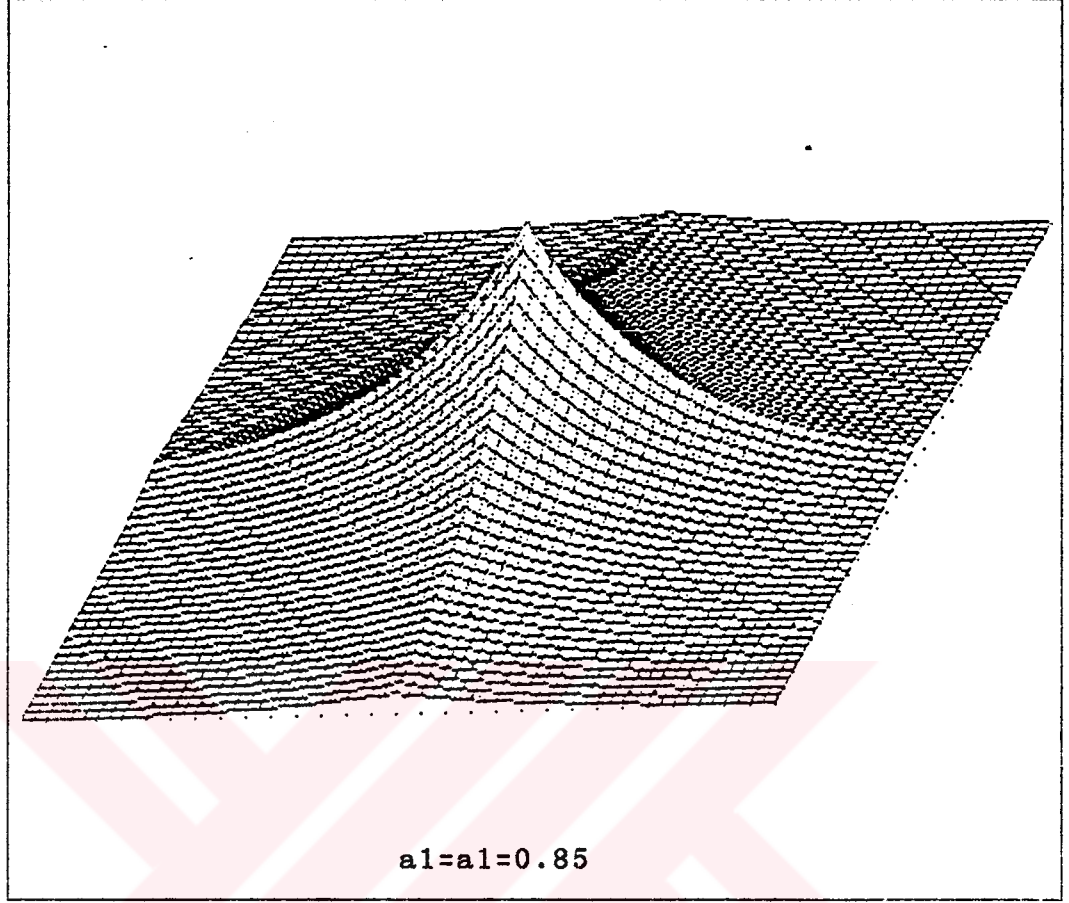
Ayrılabilir model genel olarak resimlere uygun ve de basit olduğu için kullanılır.

3.3.1.2. Ayrılamaz Özilişki Fonksiyonu

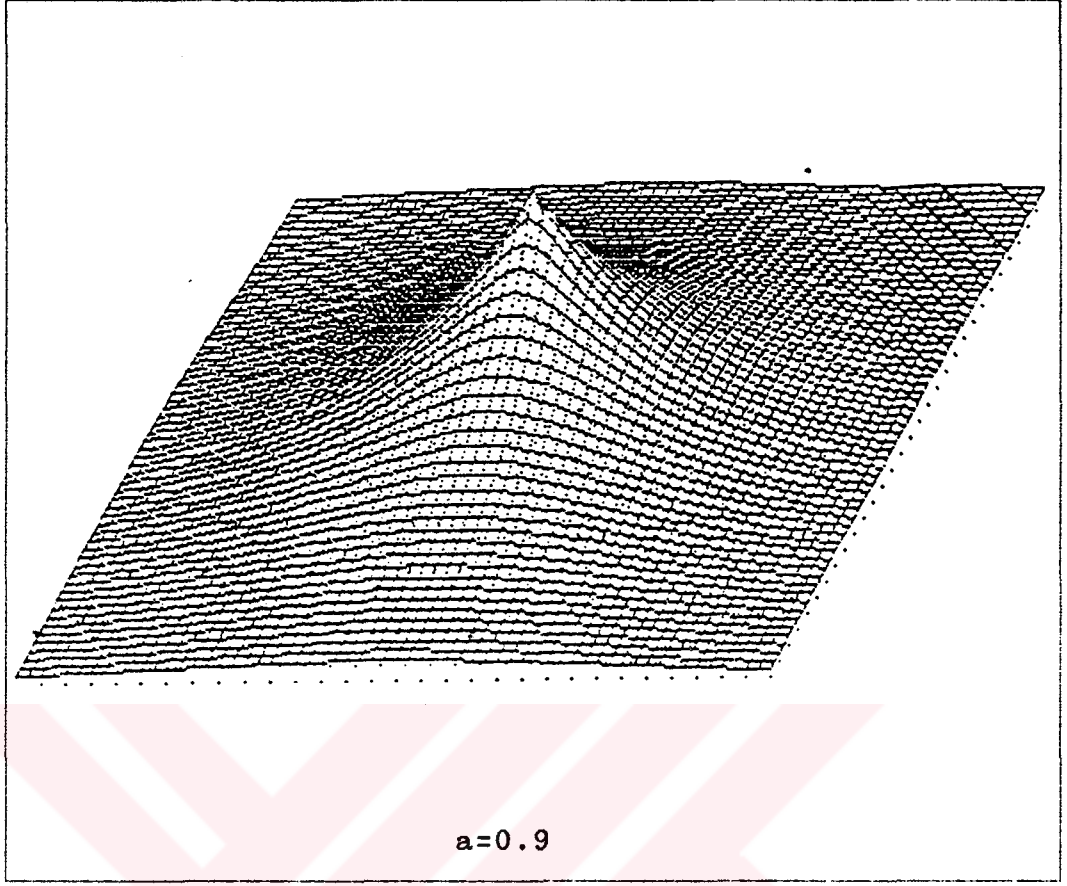
Bu model diğerinin tersine dairesel simetriye sahiptir. Ve şu şekilde verilir,

$$R_x(k,l) = a^{\sqrt{k^2+l^2}} \quad |a| < 1 \quad (3.26)$$

$a=0.9$ için özilişki fonksiyonu Şekil 3.8'de gösterilmiştir. Buradan da görüldüğü gibi ayrılamaz özilişki dairesel simetriktir.



Sekil 3.7. 1.Derece Ayrılabilir Özilişki Fonksiyonu



Şekil 3.8. 1.Derece Ayrılamaz Özilişki Fonksiyonu

BÖLÜM 4. 2-B ARDIŞIL NEDENSEL SÜZGEÇLEME

4.1. Boyut Problemi

1-B ile 2-B arasındaki temel fark boyuttur. 1-B 'tan 2-B 'ta geçerken karşımıza bir çok zorluklar çıkmaktadır. Bunlardan biri daha önce de söylenilen veri miktarıdır. 2-B 'ta veri miktarı çok fazladır. Örnek olarak 5kHz bant genişlikli bir ses işareti örneklenmesi sonunda saniyede 10 bin veri noktası elde edilir. Oysa, aynı sürede bir görüntü işareti için bu değer 7.5 milyonu aşmaktadır. Bu aradaki muazzam fark 2-B 'ta işlemlerin ne kadar zor olacağını göstermektedir.

İkinci önemli konu da, 2-B 'lu sistemlerin matematiksel modellerle incelenebilmesi 1-B 'lu sistemlere göre daha yetersizdir. 1-B 'ta sistemler adi diferansiyel denklemler yardımıyla gösterilebilmesine karşılık, 2-B 'ta kısmi diferansiyel denklemler yardımıyla modellenir. Kısmi diferansiyel denklemleri çözmek daha zordur. Bunun dışında 2-B 'ta cebirin temel teoremi geçerliliğini kaybetmiştir. Yani M . dereceden 1-B 'lu bir polinomun M adet kökü vardır. Ve de faktörize edilebilir. Ancak bu durum 2-B 'ta mümkün değildir. 2-B 'lu bir polinomun sonsuz sayıda kökü vardır. Ve daha küçük dereceli polinomlarla ifade edilemez. Bu teoremin geçersizliği kararlılığın incelenmesinde ve gerçekleştirmede zorluklar çıkarmaktadır. 1-B'ta kök yerlerinden

kararlılık kolayca incelenebilir. 2-B 'ta ise kökler kolay bulunamadığından, yani sonsuz sayıda, kararlılık testlerini uygulamak zorlaşır [6].

2-B 'lu işaretlerin Markov modellenmesi yapılırken ortaya çıkan zorluk boyuttan kaynaklanmaktadır. Durum denklemi gerçekte sonsuz uzunlukta olduğundan dolayı tanımlamak zor olmaktadır.

4.2. 1.Dereceden 1-B 'lu Ardışıl Süzgeçleme Yapısı Ve Kalman Denklemleri

Lineer ardışıl tahmin, parametrelerin en küçük karelerle veya Gauss olması durumunda en büyük benzerlikle çözümüne dayanır. Bu parametreler doğrusal formdaki gözlemle ilişkilidir [2]. Şöyleki,

$$Y_n = H_n \theta + V_n \quad (4.1)$$

Burada Y_n ve θ gözlem parametreleri, V_n gürültü vektörü, H_n ise gözlem matrisidir. θ için en küçük kareler çözümü şöyle yazılır,

$$\min_{\theta} \left[\left(\begin{bmatrix} Y_n - H_n \theta \end{bmatrix} \right)^T \left(\begin{bmatrix} Y_n - H_n \theta \end{bmatrix} \right) \right] \quad (4.2)$$

Eğer burada V_n Gauss olursa en büyük benzerlik tahmini de aynı sonucu verecektir.

Bu aynı problemin çözümünü Kalman 1960 'da zamanla değişen parametreler için çözmüştür [2]. Lineer gözlem

ve durum denklemlerini Gauss-Markov model olarak düşünmüştür.

$$x(m) = F(m-1)x(m-1) + w(m-1) \quad (4.3)$$

$$y(m) = H(m)x(m) + v(m) \quad (4.4)$$

Burada,

$x(m)$: $n \times 1$ boyutlu durum vektörü

$y(m)$: $r \times 1$ boyutlu gözlem vektörü

$F(m-1)$: $n \times n$ boyutlu durum geçiş matrisi

$H(m)$: $r \times n$ boyutlu gözlem matrisi

$w(m)$: $n \times 1$ boyutlu durum gürültü vektörü

$v(m)$: $r \times 1$ boyutlu gözlem gürültü vektörü

$w(m)$ ve $v(m)$ sıfır ortalamalı beyaz gürültüler olup, kovaryans matrisleri ise,

$$E\{w(i)w^T(j)\} = Q(i)\delta(i,j) \quad (4.5)$$

$$E\{v(i)v^T(j)\} = R(i)\delta(i,j) \quad (4.6)$$

ve ayrıca $w(m)$, $v(m)$ ve $x(0)$ birbirlerinden ayrı ayrı bağımsızdır.

Kalman en küçük varyans tahminini bulunan zaman kadar bütün gözlemler üzerinden elde etti. Burada diklik prensibi kullanılmıştır. Gözlem gürültüsü Gauss olduğu zaman Kalman süzgeci en küçük varyans hata tahmin edicisini (veya şartlı ortalamayı) verir. Eğer Gauss

olmazsa Kalman süzgeci doğrusal en küçük varyans tahmini verecektir [2].

Kalman süzgeci denklemleri şu şekilde verilir,

Başlangıç koşulu,

$$\bar{x}(0) = E(x) \quad (4.7)$$

$$M(0) = E \left\{ \{x(0) - \bar{x}(0)\} \{x(0) - \bar{x}(0)\}^T \right\} \quad (4.8)$$

Kestirim (zaman güncelleştirme) denklemleri,

$$\bar{x}(m) = F(m-1)\hat{x}(m-1) = E\{x(m)/Y^{(m-1)}\} \quad (4.9)$$

$$M(m) = F(m-1)P(m-1)F^T(m-1) + Q(m-1) \quad (4.10)$$

Süzgeçleme (ölçüm güncelleştirme) denklemleri,

$$\begin{aligned} \hat{x}(m) = \bar{x}(m) + M(m)H^T(m) \left\{ H(m)M(m)H^T(m) + R(m) \right\}^{-1} \\ \cdot \left\{ y(m) - H(m)\bar{x}(m) \right\} = E\{x(m)/Y^m\} \end{aligned} \quad (4.11)$$

$$P(m) = M(m) - M(m)H(m)^T \left\{ H(m)M(m)H(m)^T + R(m) \right\}^{-1} H(m)M(m) \quad (4.12)$$

Burada,

$$M(m) = E \left\{ \{x(m) - \bar{x}(m)\} \{x(m) - \bar{x}(m)\}^T \right\} \quad (4.13)$$

$$P(m) = E \left\{ \{x(m) - \hat{x}(m)\} \{x(m) - \hat{x}(m)\}^T \right\} \quad (4.14)$$

$M(m)$ tahmin kestirim varyansını $P(m)$ ise süzgeç kestirim varyansını ifade eder. Ayrıca gözlem dizisi olan $Y^{(m)}$,

$$Y^{(m)} = [y(0), y(1), \dots, y(m)]^T \quad (4.15)$$

dir.

Bölüm 3 'te anlatılan MAP tahmin edicisi, burada Gauss olma durumunda ölçüm güncelleştirme denklemleriyle aynı olmaktadır. Şöyleki,

$$\max_{x(m)} p(x(m)/Y^m) \longrightarrow \hat{x}(m)$$

Bu aynı zamanda,

$$\left. \frac{\partial}{\partial x(m)} \log p(x(m)/Y^m) \right|_{x(m)=\hat{x}(m)} = 0 \quad (4.16)$$

$$\left. \frac{\partial}{\partial x(m)} \log p(y(m)/x(m)) + \frac{\partial}{\partial x(m)} \log p(x(m)/Y^m) \right|_{x(m)=\hat{x}(m)} = 0 \quad (4.17)$$

Bu yazılan son eşitlikte ölçüm güncelleştirme denklemine eşit olur ki,

$$H^T(m)R^{-1}(m)\{(y(m)-H(m)x(m))-M^{-1}(m)\{x(m)-\hat{x}(m)\}\} = 0$$

(4.18)

olarak bulunur [2].

Şimdi de skaler durum incelenirse, özbağımlı modelle $x(m)$ üretilsin.

$$x(m) = ax(m-1)+w(m) \quad (4.19)$$

Burada $w(m)$ sıfır ortalamalı σ_w^2 varyanslı beyaz Gauss gürültüsüdür. a ise AR katsayısıdır. Gürültülü gözlem ifadesi ise,

$$y(m) = x(m) + v(m) \quad (4.20)$$

Burada $v(m)$ sıfır ortalamalı σ_v^2 varyanslı beyaz Gauss gürültüsüdür. $w(m)$, $x(m)$ ve $v(m)$ ile ilişkisizdir.

Şimdi de en iyi doğrusal nedensel kestirici bulunsun. $\hat{x}(m)$ en iyi lineer kestirici olsun. Yani,

$$\hat{x}(m) = E\{x(m) \mid y(j), j \leq m\} \quad (4.21)$$

Ortalama kare tahmin hatasına P_m denilirse,

$$P_m = E\{(x(m) - \hat{x}(m))^2\} \quad (4.22)$$

Burada, en iyi kestirimci P_m 'yi minimum edecek şekilde

olmalıdır. Gösterilebilir ki, optimum kestirimci ardışıl olarak aşağıdaki biçimdedir [10].

$$\hat{x}(m) = a\hat{x}(m-1) + K_m [y(m) - a\hat{x}(m-1)] \quad (4.23)$$

Bu denklem, Kalman süzgeci olarak bilinir. K_m Kalman kazancını göstermektedir. Burada K_m , Kalman süzgecinde tekrarlamalı olarak bulunabileceği gibi, tahmin hatasını minimum yapacak şekilde bulunabilir [10]. Tahmin hatası için,

$$e(m) = x(m) - \hat{x}(m) \quad (4.24)$$

$$e(m) = ax(m-1) + w(m) - [a\hat{x}(m-1) + K_m \{y(m) - a\hat{x}(m-1)\}] \quad (4.25)$$

$$e(m) = (1 - K_m)a e(m-1) + (1 - K_m)w(m) - Kv(m) \quad (4.26)$$

$$\sigma_e^2 = \frac{(1 - K_m)^2 \sigma_w^2 + K_m^2 \sigma_v^2}{1 - (1 - K_m)^2 a^2} \quad (4.27)$$

Buradan ortalama kare tahmin hatasını en aza indiren K değeri bulunabilir. K_m 'ya göre türev sifıra eşitlenerek sonuçta,

$$K_{\min} = \frac{\sqrt{\sigma_v^4 + \sigma_v^4 + 2\sigma_v^2 \sigma_v^2 + 2a^2 \sigma_v^2 (\sigma_v^2 - \sigma_v^2) + a^4 \sigma_v^4} + a^2 \sigma_v^2 - \sigma_v^2 - \sigma_v^2}{2a^2 \sigma_v^2} \quad (4.28)$$

$a=0.9$, $\sigma_v^2=1$, $\sigma_x^2=1$ için $\sigma_v^2=1-a^2=0.19$ olarak $K_{\min}=0.303$ olarak bulunur.

4.3. 1.Dereceden 2-B Ardışıl Süzgeçleme Yapısı Ve Kalman Denklemleri

1977 'de Woods ve Radewan [3] 2-B Kalman süzgeci incelemişlerdir. Onların modeli NSHP AR modele göredir. 2-B alanla sınırlanmış durumda 2-B ardışıl süzgeci tam olarak tanımlamışlardır. Daha sonra Woods ve Ingle 1981 'de görüntü onarımına uygulanması ve sınır şartlarını incelemişlerdir [11]. Woods 'da [3,11] gösterildiği gibi gerçekte durum uzayı sonsuzdur. Ancak orada durum uzayı sınırlanarak bir yaklaşıklık yapılmıştır. Yapılan yaklaşıklıkla hesaplamalarda indirgeme yapılmaya çalışılmıştır. Çelebi [2] 1990 'da ise daha değişik bir yaklaşımla olayı incelemiştir. Kestirim olarak en iyi tahmin edici formunu kullanmıştır. Ve sonuçta optimum olarak sonuca ulaştıracak $K(m,n)$ Kalman kazancı hesaplanmıştır. İşte burada, bu hesaplama yapılırken gerekli olan kovaryanslarda bir yaklaşıklık kullanmıştır. Bu da çapraz ilişki matrisinin ihmalidir. Bu durum ve denklemleri bir sonraki bölümde verilmiştir.

Aşağıdaki ayrılabilir 2-B ardışıl model 1972 'de Habibi tarafından verildi [8].

$$x(m+1,n+1)=a_1 x(m,n+1)+a_2 x(m+1,n)-a_1 a_2 x(m,n)+ \\ \sqrt{(1-a_1^2)(1-a_2^2)} w(m,n) \quad (4.29)$$

Burada a_1 ve a_2 sırayla yatay ve düşey olarak iki nokta-
arasındaki ilişki katsayısıdır. (4.29) denklemindeki
 $w(m,n)$ 'nin varyansı σ_v^2 olarak düşünülürse ve indisler
de göz önüne alınırsa,

$$x(m,n) = a_1 x(m-1,n) + a_2 x(m,n-1) - a_1 a_2 x(m-1,n-1) + w(m,n) \quad (4.30)$$

Gözlem modeli için ise,

$$y(m,n) = Cx(m,n) + Dv(m,n) \quad (4.31)$$

Burada, $w(m,n)$ ile $v(m,n)$ ilişkisiz beyaz gürültülerdir.

4.3.1. Attasi Durum Uzayı Modeli

Vektörel olarak durum ve gözlem denklemleri şu
şekilde verilir [2].

$$x(m,n) = F_1 x(m-1,n) + F_2 x(m,n-1) - F_1 F_2 x(m-1,n-1) + w(m,n) \quad (4.32)$$

$$y(m,n) = Hx(m,n) + v(m,n) \quad (4.33)$$

Burada, F_1 ve F_2 durum geçiş matrisleridir ve değişme özelliği vardır.

$$F_1 F_2 = F_2 F_1 \quad (4.34)$$

Bu model aslında (4.30) 'un genelleştirilmiş halidir ve
Attasi durum-uzayı modeli olarak bilinir [2].

Bu model [2] de incelenmiş ve Kalman denklemleri elde edilmiştir. Yukarıda anlatılan yaklaşıklık sonucu elde edilen denklemler aşağıda verilmiştir [2].

$$\hat{x} = \bar{x}(m,n) + K(m,n)[y(m,n) - H\bar{x}(m,n)] \quad (4.35)$$

Tahmin edici olarak,

$$\bar{x}(m,n) = F_1 \hat{x}(m-1,n) + F_2 \hat{x}(m,n-1) - F_1 F_2 \hat{x}(m-1,n-1) \quad (4.36)$$

olarak kabul edilir. En iyi kestiriciyi (4.35) 'e göre bulmak için kestirim hata varyansını $K(m,n)$ 'ye göre en küçük yapmak gerekir. Yani,

$$\min_K E\{((x(m,n) - \hat{x}(m,n))^T (x(m,n) - \hat{x}(m,n)))\} \quad (4.37)$$

olup, $K(m,n)$ değeri buradan bulunur.

$$P(m,n) = E\{((x(m,n) - \hat{x}(m,n))(x(m,n) - \hat{x}(m,n))^T)\} \quad (4.38)$$

$$M(m,n) = E\{((x(m,n) - \bar{x}(m,n))(x(m,n) - \bar{x}(m,n))^T)\} \quad (4.39)$$

olmak üzere yaklaşıklık altındaki Kalman eşitlikleri aşağıdaki gibidir [2].

Kestirim (zaman güncelleştirme) denklemleri,

$$\bar{x}(m,n) = F_1 \hat{x}(m-1,n) + F_2 \hat{x}(m,n-1) - F_1 F_2 \hat{x}(m-1,n-1) \quad (4.40)$$

$$M(m,n) = F_1 P(m-1,n) F_2^T + F_2 P(m,n-1) F_2^T - F_1 F_2 P(m-1,n-1) F_2^T F_1^T + Q(m,n) \quad (4.41)$$

Süzgeçleme (ölçüm güncelleştirme) denklemleri,

$$\hat{x}(m,n) = \bar{x}(m,n) + K(m,n) [y(m,n) - H\bar{x}(m,n)] \quad (4.42)$$

$$P(m,n) = M(m,n) - M(m,n) H [H M(m,n) H^T + R_v(m,n)]^{-1} H^T M(m,n) \quad (4.43)$$

Burada $K(m,n)$ Kalman kazancıdır ve (4.37) 'ye göre,

$$K(m,n) = M(m,n) H [H M(m,n) H^T + R(m,n)]^{-1} \quad (4.44)$$

dir. Böylece $P(m,n)$ için,

$$P(m,n) = M(m,n) - K(m,n) H^T M(m,n) \quad (4.45)$$

yazılabilir. Burada ayrıca,

$$Q(m,n) = E [w(i,j) w(k,l)^T] \quad (4.46)$$

$$R_v(m,n) = E [v(i,j) v(k,l)^T] \quad (4.47)$$

dir.

4.3.1.1. Skaler Attasi Modelin Kalman Denklemleri

Burada, (4.30) ile verilen modele göre bulanıklık olmaksızın durum ve sözlem denklemleri,

$$x(m,n)=a_1 x(m-1,n)+a_2 x(m,n-1)-a_1 a_2 x(m-1,n-1)+w(m,n) \quad (4.48)$$

$$y(m,n)=x(m,n)+v(m,n) \quad (4.49)$$

Kestirim (zaman güncelleştirme) denklemleri,

$$\bar{x}(m,n)=a_1 \hat{x}(m-1,n)+a_2 \hat{x}(m,n-1)-a_1 a_2 \hat{x}(m-1,n-1) \quad (4.50)$$

$$M(m,n)=a_1^2 P(m-1,n)+a_2^2 P(m,n-1)-a_1^2 a_2^2 P(m-1,n-1)+q \quad (4.51)$$

Burada q sürücü gürültü varyansıdır.

Şüzgeçleme (ölçüm güncelleştirme) denklemleri,

$$K(m,n)=M(m,n)[M(m,n)+r]^{-1} \quad (4.52)$$

$$\hat{x}(m,n)=\bar{x}(m,n)+K(m,n)[y(m,n)-\bar{x}(m,n)] \quad (4.53)$$

$$P(m,n)=M(m,n) r [M(m,n)+r]^{-1} \quad (4.54)$$

denklemleri elde edilir. Burada r beyaz Gauss gürültü varyansıdır.

4.4. Model Parametrelerinin Bulunması

Model parametreleri olan (a, σ_v^2) çeşitli yollar ile bulunabilir. Ortalama kare hatasını minimum yapacak şekilde katsayılar bulunabilir. Ayrıca bilinen bir ilişki fonksiyonu yardımıyla da bulunabilir. Bunlardan kısaca bahsedilecektir.

Yazılan programda, orjinal resime sıfır ortalamalı σ_v^2 (r) varyanslı beyaz Gauss gürültüsü eklendikten sonra resim, sıfır ortalamalı ve $1+\sigma_v^2/\sigma_x^2$ varyanslı hale getirilerek işlemler yapılmıştır.

$$\hat{\sigma}_x^2 = \hat{\sigma}_y^2 - \sigma_v^2 \quad (4.55)$$

olarak orjinal resmin varyansıdır. σ_y^2 gürültülü resmin varyansıdır ve en büyük olabilirlik kestirimi,

$$\hat{\sigma}_y^2 = 1/(MM) \sum_{m,n \in M} \{y(m,n) - \bar{y}(m,n)\}^2 \quad (4.56)$$

$$\hat{m}_x = \hat{m}_y = \bar{y}(m,n) = 1/(MM) \sum_{m,n \in M} y(m,n) \quad (4.57)$$

olarak verilir.

4.4.1. Ayrılabilir Özilişki Fonksiyonuyla Bulma

(3.25) ile verilen özilişki ifadesi örnek özilişki yardımıyla,

$$\hat{R}_x(k,l) = 1/(MM) \sum_{i=0}^{M-1-kM-1-l} \sum_{j=0}^{M-1-kM-1-l} x(i,j)x(i+k,j+l) \quad (4.58)$$

olmak üzere ve $\sigma_x^2=1$ seçildiğinden,

$$\hat{a}_1 = \hat{R}_x(1,0) \quad \hat{a}_2 = \hat{R}_x(0,1) \quad (4.59)$$

$$q = \hat{\sigma}_v^2 = (1 - \hat{a}_1^2)(1 - \hat{a}_2^2) \quad (4.60)$$

olarak hesaplanır.

Asıl amaç gürültülü ifadeden parametrelerin bulunmasıdır. Gürültülü gözlem ifadesinin örnek özilişkisi yazılırsa,

$$\hat{R}_y(k,l) = 1/(MM) \sum_{i=0}^{M-1-k} \sum_{j=0}^{M-1-l} y(i,j)y(i+k,j+l) \quad (4.61)$$

$y(i,j)$ 'nin değerini yerine yazarsak beyaz toplamsal gürültülü durum için,

$$R_y(k,l) = R_x(k,l) \quad k,l \neq 0 \quad (4.62)$$

$$R_y(0,0) = R_x(0,0) + \sigma_v^2 \quad k=l=0 \quad (4.63)$$

olarak yazılır.

Bu denklemlerden yararlanarak gürültülü ifadeden parametreler şöyle bulunur,

$$\hat{a}_1 = \hat{R}_y(1,0) / (\hat{R}_y(0,0) - \sigma_v^2) \quad , \quad \hat{a}_2 = \hat{R}_y(0,1) / (\hat{R}_y(0,0) - \sigma_v^2) \quad (4.64)$$

olarak hesaplanabilir.

4.4.1.1. Simülasyon Sonuçları

Bilgisayar programı sonucunda, (4.59) ve (4.60) ile verilen denklemlere göre parametreler orjinal resim üzerinden hesaplandığında,

$a_1=0.967$ $a_2=0.935$ ve $\sigma_v^2=0.008$ olarak bulunmuştur.

(4.64) 'e göre, yani gürültülü resim üzerinden hesaplandığında ise (giriş işaret-gürültü oranı $SNRI=3dB$ iken) $a_1=0.968$ $a_2=0.937$ olarak bulunmuştur.

Sonuçlar birbirlerine oldukça yakın bulunmuştur. Süzgecin verimini incelerken aşağıda verilen eşitliklerden yararlanılmıştır [2].

$$SNRI (dB) = 10 \log \frac{\sum_{i,j \in M} \{x(i,j) - \bar{x}\}^2}{\sum_{i,j \in M} \{x(i,j) - y(i,j)\}^2} \quad (4.65)$$

$$SNRO (dB) = 10 \log \frac{\sum_{i,j \in M} \{x(i,j) - \bar{x}\}^2}{\sum_{i,j \in M} \{x(i,j) - \hat{x}(i,j)\}^2} \quad (4.66)$$

$$\text{Süzgeç Kazancı} = G = 10 \log \frac{\sum_{i,j \in M} \{x(i,j) - y(i,j)\}^2}{\sum_{i,j \in M} \{x(i,j) - \hat{x}(i,j)\}^2} \quad (4.67)$$

Şekil 4.1 'de orjinal LENA resmi verilmiştir.

Şekil 4.2, Şekil 4.3 ve Şekil 4.4 'te değişik işaret-gürültü oranlarında LENA resmi ve onarılmış hali görülmektedir. Tablo 4.1 'de değişik işaret-gürültü oranlarında sonuçlar verilmiştir. Görüldüğü gibi yüksek giriş işaret-gürültü oranlarında sonuçlar daha iyi olmaktadır.

Tablo 4.1. Birinci metotta farklı SNR'de giriş ve çıkış.

SNRI (dB)	SNRO (dB)	G (dB)
3	7.77	4.75
10	15.83	5.82
20	27.35	7.302

4.4.2. Gürültülü Gözlemden AR Parametrelerinin Uyarlanabilir Olarak Güncelleştirilmesi

Bir önceki bölümde model parametreleri tüm resim üzerinden bulunmuştu. Yani parametreler tüm resim için tek değerdir. Bunun bazı sakıncaları vardır. Resim yöresel olarak çok değişik özelliklere sahip olabilir. Bu gerçek zamanda işlem yapmaya uygun değildir. Ancak resmi küçük bloklara bölerek, **her nokta için o bloğun içinde parametreler bulunur.** Bu bloğun boyutunun uygun seçilmesi önemlidir. Şimdi, sırayla bu yöntem aşağıda anlatılacaktır [12].

Bir resmin AR modeli,

$$x(m,n) = \sum_{k,l \in D} a_{kl}(m,n)x(m-k,n-l) + w(m,n) \quad (4.88)$$

$$y(m,n) = x(m,n) + v(m,n) \quad (4.89)$$

ile verilsin. Burada D bölgesi $L \times L$ destek bölgesidir.

$y(m,n)$ gürültülü gözlemdir. $w(m,n)$ ve $v(m,n)$ sırayla σ_w^2 σ_v^2 varyanslı sıfır ortalamalı ilişkisiz beyaz Gauss gürültüleridir.

a_{kl} için yansız kestirimler,

$$\hat{\Theta} = \left[\sum_w y_1(m,n) y_1^T(m,n) - N I \sigma_v^2 \right]^{-1} \left[\sum_w y_1(m,n) y(m,n) \right] \quad (4.70)$$

Burada,

$$\hat{\Theta} = [a_{-L-L}, \dots, a_{-LL}, a_{-L+1L}, \dots, a_{0-L}, \dots, a_{0-1}]$$

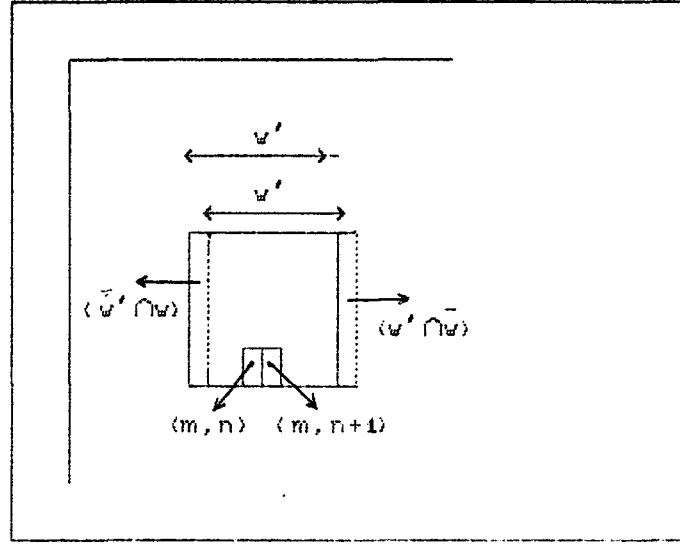
$$y_1(m,n) = [y(m-L, n-L), \dots, y(m-L, n+L), \dots, y(m, n-L), \dots, y(m, n-1)]$$

ve N ise W penceresi içindeki nokta sayısıdır. Her W verisi üzerinden zamanla değişen bir şekilde katsayılar bulunacaktır. $a_{k,l}(m, n+1)$ noktasına gelindiğinde ise W' penceresi üzerinden yeni katsayı hesaplanacaktır. Bu durum Şekil 4.5 'de gösterilmiştir. Yani pencere tüm resim boyunca kaydırılmaktadır. Burada, pencerenin boyutu W , resmin durağanlık varsayımını bozmayacak şekilde yeterince küçük, gürültünün etkisini yeterince

azaltabilecek kadar da büyük olmalıdır.

Şimdi de $a_{kl}(m, n+1)$ 'i $a_{kl}(m, n)$ den bulmak istenirse

$$\hat{\Theta}(m, n) = P(m, n) b(m, n) \quad (4.71)$$



Şekil 4.5. Kayan Veri Penceresi

olarak yazılabilir. Burada,

$$P(m,n) = \left[\sum y_1(i,j) y_1^T(i,j) - N I \sigma_v^2 \right]^{-1} \quad (4.72)$$

$$b(m,n) = \left[\sum y_1(i,j) y(i,j) \right] \quad (4.73)$$

$$\hat{\theta}(m,n+1) = P(m,n+1) b(m,n+1) \quad (4.74)$$

olarak yazılır. Burada $P(m,n+1)$ ve $b(m,n+1)$ toplamları W' penceresi üzerindendir. Şekil 4.5 'de W ve W' pencereleri gösterilmiştir. $(m,n+1)$ noktasını (m,n) 'den yararlanarak bulmak hesap indirgemesi açısından çok önemlidir. Şekil 4.5 'den görüleceği gibi,

$$\begin{aligned}
b(m,n+1) &= b(m,n) + \sum_{(i,j) \in v' \cap \bar{w}} y_1(i,j) y(i,j) \\
&\quad - \sum_{(i,j) \in \bar{v}' \cap w} y_1(i,j) y(i,j)
\end{aligned} \tag{4.76}$$

Şekil 4.5 'de toplanan ve çıkarılan bölgeler gösterilmektedir. $P(m,n+1)$ için de,

$$P^{-1}(m,n) = \sum_w y_1(i,j) y_1^T(i,j) - N I \sigma_v^2 \tag{4.76}$$

$$\begin{aligned}
P^{-1}(m,n+1) &= P^{-1}(m,n) + \sum_{(i,j) \in v' \cap \bar{w}} y_1(i,j) y_1^T(i,j) \\
&\quad - \sum_{(i,j) \in \bar{v}' \cap w} y_1(i,j) y_1^T(i,j)
\end{aligned} \tag{4.77}$$

Toplamlarda gösterilen bölgelerin her ikisinde de sadece bir nokta bulunmaktadır. Böylece (4.77) şöyle yazılabilir,

$$P^{-1}(m,n+1) = P^{-1}(m,n) + y_1^T(k,1) y_1(k,1) - y_1^T(o,p) y_1(o,p) \tag{4.78}$$

$$Q^{-1}(m,n) = P^{-1}(m,n) + y_1^T(k,1) y_1(k,1) \tag{4.79}$$

olmak üzere ve de matris tersi alma kuralı uygulayarak sonuçta,

$$P(m,n+1)=Q(m,n)+\frac{\sum_{p=1}^T Q(m,n)y_1(o,p)y_1(o,p)Q(m,n)}{1-\sum_{p=1}^T y_1(o,p)Q(m,n)y_1(o,p)} \quad (4.80)$$

sonucuna kolayca ulaşılır. (4.75) ve (4.80) denklemleri yansız parametre kestirimini verecektir. Her noktanın hesabı yapıldıktan sonra gerekli bilgiler saklanarak bir sonraki nokta için tekrar hesap yapılmayacaktır. Bu şekilde yapılan işlemler sonucunda büyük ölçüde hesaplamalar azalacaktır [12].

Yukarıda anlatılan yolla yapılan parametre kestirimi sonucu, LENA resmine uygulanmış ve diğerinden iyi sonuçlar alınmıştır.

4.4.2.1. Simülasyon Sonuçlar

Son bölümde anlatılan yöntemle göre parametrelerin kestirimi sonucu Şekil 4.6, Şekil 4.7, Şekil 4.8 'de görülen görüntüler elde edilmiştir. Burada $M=15$ için sonuçlardır. Şekil 4.9 ve Şekil 4.10 'da ise sırayla 10 ve 20 için sonuçlar görülmektedir.

Sonuçlar birde Tablo 4.2 ' de verilmiştir. Görüldüğü gibi (Tablo 4.2.a.ve b.) $M=15$ için en iyi durum elde edilmiştir. Dolayısıyla M 'nin etkisi büyük olmaktadır.

Yüksek işaret-gürültü oranlarında (10dB-20dB) iyileşme veya kazanç çok iyi olmaktadır. Ancak düşük SNR '

de ise çok iyi sonuç alınamamaktadır. Bu yüzden daha değişik modeller (2.derece gibi) ve parametre kestirimi-
ne ihtiyaç vardır.

Tablo 4.2. İkinci metotta farklı SNR'de giriş ve çıkış.

a) M=15 için giriş-çıkış

b) SNRI=10dB için farklı M'de giriş-çıkış

M=15 (a)

SNRI (dB)	SNRO (dB)	G (dB)
3	9.54	6.51
10	18.56	8.54
20	30.72	10.56

SNRI=10dB (b)

M	SNRO (dB)	G (dB)
10	15.27	5.26
15	18.56	8.54
20	14.67	4.65

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu çalışmada, toplamsal Gauss gürültülü görüntülerin Kalman süzgecine uygulanması ve değişik işaret-gürültü oranlarında sistemin verimi üzerinde durulmuştur. Bir görüntünün modellenmesinden yola çıkılarak ve de gözlem modeli oluşturulup Kalman denklemleri elde edilmiştir. Model parametrelerinin bulunmasında iki yol izlenmiştir. Birincisinde tüm resmi ele alarak özilişki yoluyla parametreler bulunmuştur. Diğerinde ise, resmi küçük bloklara ayrılarak resim üzerinde kaydırılan bloklar içinde parametreler uyumlu olarak bulunmuştur. Bu yöntemler Turbo C programlama dili ile yazılan bir bilgisayar programı ile gerçekleştirildi.

Programda kullanılan LENA resmine sıfır ortalamalı toplamsal beyaz Gauss gürültüsü eklendikten sonra, resim sıfır ortalamalı $1+\sigma_v^2/\sigma_x^2$ varyanslı hale getirilerek işlemler yapılmıştır. Toplamsal beyaz Gauss üretimi için Box-Muller metodu kullanılmıştır [13].

Birinci metoda göre bulunan sonuçlar Şekil 4.2, Şekil 4.3 ve Şekil 4.4 'te verilmişti. Birde sonuçlar Tablo 4.1. 'de verilmiştir. Bu tabloya göre iyi kazanç elde edilmesine rağmen onarım sonucu resme gürültü

eklenmiş olduğu görülmektedir. Bu bozulmalar kenarlar-
da daha çok olmaktadır. Sanki resim düzleşmiştir. Bu
ise resmin derinliğini kaybettirmektedir. Özellikle yö-
resel olarak hızlı değişen görüntülerde bu etkidaha faz-
la görülecektir. Bu nedenle bu tür görüntüler için
uyumlu olarak katsayı hesabı yapılması gerekmektedir.
Bir sonraki yöntemle elde edilen bu nedenle daha iyi
sonuçlar verecektir. Burada önemli bir konu da SNR 'nın
durumudur. Eğer SNR düşükse daha karmaşık model ve pa-
rametre kestirimi kullanma zorunluluğu ortaya çıkacaktır.

İkinci metoda göre bulunan sonuçlar ise Şekil 4.6,
Şekil 4.7 ve Şekil 4.8 'de gösterilmiştir. Tablo 4.2.'
de farklı durumlar gösterilmiştir. Diğerine göre oldukça
iyi sonuçlar elde edilmiştir. Çünkü ikinci metodun as-
lında bir miktar kenarları da dikkate aldığı söylenebi-
lir. Farklı M değerleri için de Şekil 4.9 ve Şekil 4.10
'da görüntüler gösterilmiştir.

Pencere boyutunun seçimi de burada önemlidir.
Tablo 4.2.b.'de SNR=10dB de üç değişik M değeri için
sonuçlar verilmiştir.

5.2. Öneriler

İşaret işlemede karşılaşılan en temel problemler-
lerden biri düşük orandaki işaret-gürültü oranı (SNR)
olmasıdır. Bu çalışmada giriş SNR 3dB 'den küçük olması
durumunda iyi sonuç alınamamıştır.

Yüksek dereceden modelleme yapılarak daha düşük

SNR 'nlarında iyi sonuçlar alınabilir. Ancak yüksek dereceden modelleme işlemsel yönden çok fazla külfet getirecektir. Ayrıca kenara hassasiyeti iyi olan süzgecintasarımı çok iyi sonuç verecektir. Yani kenar sez-meli olarak yapılması durumunda onarım sonucu eklenen gürültüyü büyük ölçüde azaltacaktır.

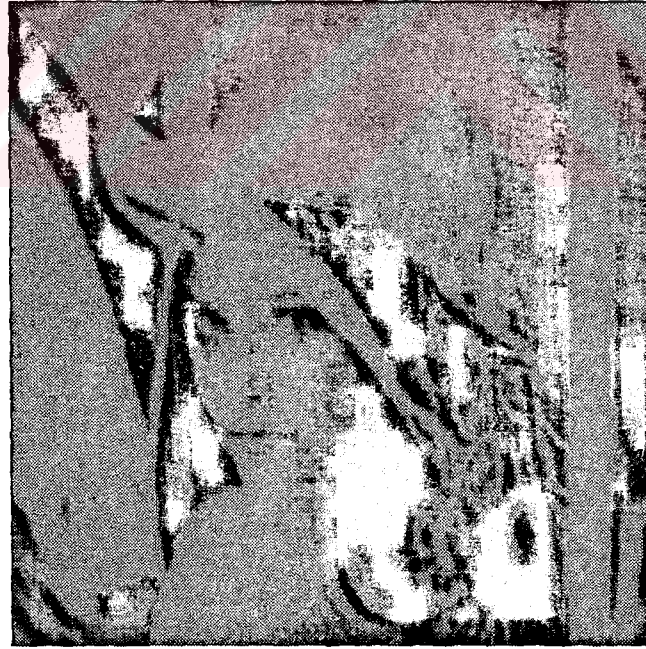




Şekil 4.1. Orjinal LENA resmi



(a)



(b)

Şekil 4.2. Birinci metodla onarım
a) $SNRI=3dB$ olan gürültülü LENA N(0,0.5)
b) $SNRO=7.77dB$ Kazanç= $G=4.75dB$



(a)



(b)

Şekil 4.3. Birinci metoda göre onarım
 a) $SNRI=10\text{dB}$ olan gürültülü LENA $N(0,0.01)$
 b) $SNRO=15.8\text{dB}$ Kazanç= $G=5.82$



(a)



(b)

Şekil 4.4. Birinci metoda göre onarım

a) $SNRI=20dB$ olan gürültülü LENA $N(0,0.001)$

b) $SNRO=27.35dB$ Kazanç= $G=7.3dB$



(a)



(b)

Şekil 4.6. $M=15$ için (ikinci metod)
 a) $SNRI=3dB$ olan gürültülü LENA $N(0,0.5)$
 b) $SNRO=9.5dB$ Kazanç= $G=6.5dB$



(a)



(b)

Şekil 4.7. $M=15$ için (ikinci metod)

a) $SNRI=10\text{dB}$ olan gürültülü LENA $N(0,0.01)$

b) $SNRO=18.5\text{dB}$ Kazanç= $G=8.5\text{dB}$



(a)



(b)

Şekil 4.8. $M=15$ için (ikinci metod)

a) $SNRI=20dB$ olan gürültülü LENA $N(0,0.001)$

b) $SNRO=30.7dB$ Kazanç= $G=10.66dB$



(a)



(b)

Şekil 4.9. $M=10$ için (ikinci metod)
 a) $SNRI=10\text{dB}$ olan gürültülü LENA $N(0,0.01)$
 b) $SNRO=15.27\text{dB}$ Kazanç= $G=5.26\text{dB}$



(a)



(b)

Şekil 4.10. $M=20$ için (ikinci metod)

a) $SNRI=10\text{dB}$ olan gürültülü LENA $N(0,0.01)$

b) $SNRO=14.67\text{dB}$ Kazanç= $G=4.65\text{dB}$

KAYNAKLAR

- [1] AZIMI-SADRAJI, M.R., WONG, D.W., Two Dimensional Block Kalman Filtering for Image Restoration, IEEE Trans. Acoust. Speech. Signal Processing, Vol. ASSP-35, No.12, pp. 1736-1749, (1987).
- [2] ÇELEBİ, M.E., Robust Recursive Estimation With Applications to Image Processing, Ph.D. Thesis, Polytechnic University, (1990).
- [3] WOODS, J.W., Radewan, C.H., Kalman Filtering in Two Dimensions, IEEE Trans. Inform. Theory, Vol. IT-23, NO.4, pp. 473-482, (1977).
- [4] THERRIEN, C.W. et al., Statistical Model Based Algorithms for Image Analysis, Proc. of IEEE Vol. 74, pp. 532-551, (1986).
- [5] HAYKIN, S., Adaptive Filter Theory, Englewood Cliff, New Jersey, Prentice-Hall, (1986).
- [6] LIM, J.S., Two Dimensional Signal and Image Processing, Englewood Cliff, New Jersey, Prentice-Hall, (1990).
- [7] EKSTROM, M.P., Realizable Wiener Filtering in Two Dimensions, IEEE Trans. ASSP, Vol. 30, pp. 31-40, (1982).
- [8] HABIBI, A., Two Dimensional Bayesian Estimation of

Images, Proc. IEEE, Vol. 60, pp. 878-883, (1972).

- [9] KATAYAMA, T., Restoration of Noisy Images Using a Two Dimensional Linear Model, IEEE Trans. SMC, Vol. 9, pp.711-717, (1979).
- [10] PAPOULIS, A., Probability, Random Variables, and Stochastic Processes, Mc.Graw Hill, (1984).
- [11] WOODS, J.W., INGLE, V.K., Kalman Filtering in Two Dimesions:Further Results, IEEE Trans. ASSP ,Vol. 29, pp. 187-197, (1981).
- [12] TEKALP, A.M., KAUFMAN, H., WOODS, J.W., Fast Recursive Estimation of the Parameters of a Space Varying Autoregressive Image Model, IEEE Trans. ASSP, Vol.33, No.2, pp. 469-472, (1985).
- [13] PRESS, W.H.et al., Numerical Recipes in C, Cambridge University Press, (1988).

EK A

UYGULANAN YONTEMLERIN PROGRAM LISTESI

```
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <alloc.h>
```

```
#define M1 259200
#define IA1 7141
#define IC1 54773
#define RM1 (1.0/M1)
#define M2 134456
#define IA2 8121
#define IC2 28411
#define RM2 (1.0/M2)
#define M3 243000
#define IA3 4661
#define IC3 51349
#define pi 3.141592
```

```
float ran1(int*);
float gasdev(int*);
```

```
void gur_ekle(void);
void lena_oku(void);
void ger_oziliski(void);
void AR_kat(void);
void gen_oziliski(void);
void blok_oziliski(void);
void cizim(void);
void kestirim1(void);
void kestirim2(void);
void kal_oi2(void);
void orj_lena(void);
void giris(void);
void secim(void);
void surgeol(void);
void surgec2(void);
```

```
float snri;
float snro;
```

```
float g;
```

```

int ref;
float Ry[2][2];
float yy[30][30];
float a1,a2,q,bq;
int idum=-2,M;
float ortal,varyan,ortyd,varyd,varydx;
float rr,r;

main()

{
char kr;
clrscr();
giris();
do{
secim();
}while(1);
}

void giris()
{
int gd=DETECT,gm;
char *konu[3]={"TOPLAMSAL GAUSS GUPULTULU",
"GORUNTULERIN KALMAN SUZGECI ILE",
"ONARIMI"};

char *ad1="Yasar Becerikli";
char *ad2="M.E.Celebi";
clrscr();
initgraph(&gd,&gm,"");
if(graphresult()!=grOk) {
printf("grafik karti hazir degil\n");
printf("%s\n",grapherrormsg(graphresult()));
}

setbkcolor(15);
setcolor(GREEN);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
setviewport(getmaxx()/6,getmaxy()/6,getmaxx()/4,
getmaxy()/4,0);
rectangle(0,0,getmaxx()/2+80,getmaxy()/2+80);
settextstyle(1,0,2);

moveto(getmaxx()/4+35,getmaxy()/4-70);
settextjustify(1,1);
outtext(konu[0]);

moveto(getmaxx()/4+40,getmaxy()/4-40);
settextjustify(1,1);
outtext(konu[1]);

moveto(getmaxx()/4+25,getmaxy()/4-10);
settextjustify(1,1);
outtext(konu[2]);

settextstyle(1,0,2);

```

```

moveto(getmaxx()/4+150,getmaxy()/2+60);
outtext(ad1);

settextstyle(1,0,2);
moveto(getmaxx()/4-100,getmaxy()/2+60);
outtext(ad2);
getch();
cleardevice();
closegraph();
}

/*Secim*/
void secim()
{
    int sayi;
    clrscr();
    textcolor(15);
    textbackground(1);
    gotoxy(25,5) ;cprintf(" 1. GENEL AR KATSAYI
                                KESTIMIYLE ONARIM        ");
    gotoxy(25,7) ;cprintf(" 2. UYUMLU AR KATSAYI
                                KESTIRIMIYLE ONARIM        ");
    gotoxy(25,9) ;cprintf(" 3. ORIJINAL RESIM            ");
    gotoxy(25,11);cprintf(" 4. CIKIS                    ");
    gotoxy(25,20);cprintf("SECIMINIZ(1-4):                ");
    textbackground(MAGENTA);
    gotoxy(40,20);scanf("%d",&sayi);
    ref=sayi;
    switch(sayi)
    {
        case 1:suzgeci1();break;
        case 2:suzgeci2();break;
        case 3:orj_lena();break;
        case 4:exit(0);
    }
}

/*Orjinal Lena*/
void orj_lena()
{
    textcolor(1);
    clrscr();
    gotoxy(0,10);
    puts("Lutfen bekleyiniz.");
    lena_oku();
    ger_oziliski();
    AR_kat();
    cizim();
}

/*Yontem1*/
float g;
void suzgeci1()
{
    textcolor(1);
    clrscr();

```

```

puts("Gozlem gurultu varyansini giriniz.(r)");
scanf("%f",&g);
puts("Lutfen bekleyiniz.");
r=g;
gur_ekle();
gen_oziliski();
kestirim1();
kal_ciz();
}

/*Yontem2*/
void suzgeo2()
{
int mr;
textcolor(1);
clrscr();
puts("Gozlem gurultu varyansini giriniz.(r)");
scanf("%f",&g);
puts("Pencere boyutunu giriniz.(M)(3-30 arasi)");
scanf("%d",&mr);
puts("Lutfen bekleyiniz.");
r=g;
M=mr;
gur_ekle();
kestirim2();
kal_ciz();
}

void lena_oku()
{
register int i,j;
float va=0,ort=0;
int temp,tem;
FILE *fp,*fs;
if((fp=fopen("lena.img","rb"))==NULL) {
printf("Dosya acilamadi");
exit(0);
}
for(j=0;j<256;j++){
for(i=0;i<256;i++){
temp=fgetc(fp);
ort +=(float)temp;
}
}
ortal=(float)ort/65536;

if((fs=fopen("lena.img","rb"))==NULL) {
printf("Dosya acilamadi");
exit(0);
}

for(j=0;j<256;j++){
for(i=0;i<256;i++){
tem=fgetc(fs);
va+=(float)(tem-ortal)*(tem-ortal);
}
}

```

```

    }
    varyan=(float)va/65536;
    fclose(fp);
    fclose(fs);
}

float Rx[2][2];
float imm[2][256];

void ger_oziliski()
{
    register int i,j,jj,k,l,mm,nn;
    float sq,Rxx,temp1,temp2;
    FILE *fp;

    sq=(varyan);
    if((fp=fopen("lena.img","rb"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }

    for(k=0;k<2;k++){
        for(l=0;l<2;l++){
            Rxx=0.0;
            for(mm=0;mm<2;mm++){
                for(nn=0;nn<256;nn++){
                    temp1=fgetc(fp);
                    imm[mm][nn]=temp1;
                }
            }
            for(i=0;i<(256-k);i++){
                for(j=0;j<(256-l);j++){
                    Rxx+=(float)(imm[0][j]-ortal)*(imm[k][j+l]-ortal)/sq;
                }
            }
            for(jj=0;jj<256;jj++){
                imm[0][jj]=imm[1][jj];
                temp2=fgetc(fp);
                imm[1][jj]=temp2;
            }
        }
        fseek(fp,0,SEEK_SET);
        Rx[k][l]=(float)Rxx/(256.0*256.0);
    }
}

fclose(fp);
}

/*AR Kat*/
void AR_kat()
{
    a1=(float)Rx[1][0]/Rx[0][0];
    a2=(float)Rx[0][1]/Rx[0][0];
    q=(float)(1-a1*a1-a2*a2+a1*a1*a2*a2)*Rx[0][0];
}

```

```

void cizim()
{
    register int i,j,t;
    char *dname;
    char *drn,*drm,oh;

    char varr[10],ortt[10],ar1[10],ar2[10];
    int gd=DETECT,temp,gm;
    FILE *fp;

    struct palettetype pal;

    if((fp=fopen("lena.img","rb"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }
    clrscr();
    initgraph(&gd,&gm,"");
    if(graphresult()!=grOk){ printf("grafik karti hazir degil\n");
    getch();}

    getpalette(&pal);
    for(i=0;i<pal.size;i++)
        setrgbpalette(pal.colors[i],i*4,i*4,i*4);

    for(j=0;j<256;j++){
        for(i=0;i<256;i++){
            temp=fgetc(fp);
            t=(int)(temp/16);
            putpixel(i+(getmaxx()/2-128),j+(getmaxy()/2-128),t);
        }
    }

    settextstyle(0,0,1);
    govt( varyan,8,varr);
    govt(ortal,8,ortt);
    govt(a1,8,ar1);
    govt(a2,8,ar2);
    outtextxy(getmaxx()/2-128,getmaxy()/2+135,"Ortalama :");
    outtextxy(getmaxx()/2-128,getmaxy()/2+145,"Varyans :");
    outtextxy((getmaxx()/2-128)+100,getmaxy()/2+135,ortt);
    outtextxy((getmaxx()/2-128)+100,getmaxy()/2+145,varr);

    outtextxy(getmaxx()/2-128,getmaxy()/2+155,"a1      :");
    outtextxy(getmaxx()/2-128,getmaxy()/2+165,"a2      :");
    outtextxy((getmaxx()/2-128)+100,getmaxy()/2+155,ar1);
    outtextxy((getmaxx()/2-128)+100,getmaxy()/2+165,ar2);

    getch();
    closegraph();
    fclose(fp);
}

/*Gurultu Ekleme*/
void gur_ekle()
{

```

```

register int i,j,jj,k,l,mm,nn;
int yd1;
float sq,yd,yd2;
float tmp;
float gr,sr,ort=0.0,vry=0.0;
float oryd=0.0,vryd=0.0;
FILE *fl,*fg,*fgr,*fgs;

sr=sqrt(r);

if((fl=fopen("lena.img","rb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
if((fg=fopen("gurultu.dat","w"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((fgs=fopen("gur.asc","wb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        gr=sr*gasdev(&idum);
        ort+=(float)(gr/65536);
        vry+=(float)((gr*gr)/65536);
        yd=(float)(fgetc(fl)+gr);
        yd1=(int)yd;
        if(yd1<0) yd1=0;
        if(yd1>255) yd1=255;
        fprintf(fg,"%f\n",yd);
        fprintf(fgs,"%c",yd1);
        oryd+=(float)(yd/65536);
    }
}

fclose(fg);
fclose(fl);
fclose(fgs);

if((fg=fopen("gurultu.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fg,"%f\n",&yd);
        vryd+=(float)((yd-oryd)*(yd-oryd)/65536);}}
fclose(fg);

ortyd=oryd;
varyd=vryd;
varydx=(float)(varyd-r);
printf("ortgur=%f      vrygur=%f\n",ort,vry);
printf("ortydx=%f      vrydx=%f\n",ortyd,vryd);

```

```

    getch();
    sq=sqrt(varydx);
    if((fg=fopen("gurultu.dat","r"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }

    if((fgr=fopen("gur.dat","w"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }
    for(i=0;i<256;i++){
        for(j=0;j<256;j++){
            fscanf(fg,"%f\n",&yd);
            yd2=(float)((yd-ortyd)/sq);
            fprintf(fgr,"%f",yd2);}}

    fclose(fg);
    fclose(fgr);

    rr=(float)(r/varydx);
    printf("Gozlem gurultu var=rr=%f\n",rr);
}

/* Genel oziliski ve AR katsayilari */
void gen_oziliski()
{
    register int i,j,jj,k,l,mm,nn;
    float im[2][256];
    float temp1,temp2;
    float sq,Ryy=0.0;
    FILE *fg;

    rr=(float)(r/varydx);
    sq=varydx;

    if((fg=fopen("gurultu.dat","r"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }
    for(k=0;k<2;k++){
        for(l=0;l<2;l++){
            Ryy=0.0;
            for(mm=0;mm<2;mm++){
                for(nn=0;nn<256;nn++){
                    fscanf(fg,"%f\n",&temp1);
                    im[mm][nn]=temp1;
                }
            }

            for(i=0;i<(256-k);i++){
                for(j=0;j<(256-l);j++){
                    Ryy+=(float)((im[0][j]-ortyd)*(im[k][j+1]-ortyd)/sq);
                }
                for(jj=0;jj<256;jj++){
                    im[0][jj]=im[1][jj];
                }
            }
        }
    }
}

```

```

        fscanf(fg,"%f\n",&temp2);
        im[1][jj]=temp2;
    }
}
rewind(fg);
Ry[k][1]=(float)(Ryy/65536);
}
}
fclose(fg);

a1=(float)(Ry[1][0]/(Ry[0][0]-rr));
a2=(float)(Ry[0][1]/(Ry[0][0]-rr));
q=(float)((1-a1*a1)*(1-a2*a2)*(Ry[0][0]-rr));
}

/* Blok oziliski ve AR katsayı hesabı*/
void blok_oziliski()
{
    register int i,j,k,l;
    float Ry[2][2],ortyy=0.0;

    for(i=0;i<M;i++)
        for(j=0;j<M;j++) ortyy+=yy[i][j];
        for(k=0;k<2;k++){
            for(l=0;l<2;l++){
                Ry[k][1]=0.0;
            }
        }
    for(i=0;i<(M-k);i++)
        for(j=0;j<(M-1);j++)
            Ry[k][1]+=(float)(yy[i][j]*yy[i+k][j+1]);
    Ry[k][1]/=(M*M);
}

a1=(float)(Ry[1][0]/(Ry[0][0]-rr));
a2=(float)(Ry[0][1]/(Ry[0][0]-rr));
q=(float)((1-a1*a1)*(1-a2*a2)*(Ry[0][0]-rr));

if(a1>0.99) a1=0.989;
if(a1<-0.99) a1=-0.989;
if(a2>0.99) a2=0.989;
if(a2<-0.99) a2=-0.989;
q=(float)((1-a1*a1)*(1-a2*a2));
}

void kestirim1()
{
    register int i,ii,iii,j,jj;
    float estx[2][256],ysat[256],ysut[256],ydr;
    float ydl,pp[2][256],kk[2][256];
    int laskk[2][256],leyd;
    float pest,mm,m0,sqt,gr;
    float tem,temp,temp1,hava,havar=0.0,havaort=0.0;
    float sonort=0.0,sonvar=0.0;
    char fname[20];
    FILE *fp,*fs,*f1,*fm,*fkal,*laski;

```

```

m0=1.0;

sqt=sqrt(varydx);

if((fp=fopen("gur.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fp,"%f",&y);
        if(i==0) ysat[j]=y;
        if(j==0) ysut[i]=y;
    }
}
fclose(fp);

if((fp=fopen("gur.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((fs=fopen("sonl.dat","w"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((fm=fopen("mm.dat","w"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((fkal=fopen("kalman.dat","w"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((laski=fopen("lask.asc","wb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

printf("Dosyalara girdi:\n");
for(j=0;j<256;j++){
    estx[0][j]=(float)m0/(m0+rr)*ysat[j];
    pp[0][j]=(float)m0*rr/(m0+rr);
    kk[0][j]=(float)m0/(m0+rr);
    fprintf(fm,"%f",m0);
    fprintf(fkal,"%f",kk[0][j]);
}
for(j=0;j<256;j++) fscanf(fp,"%f",&ysat[j]);
for(i=1;i<256;i++){
    for(j=1;j<256;j++){
        if(j==1) {
            fscanf(fp,"%f",&ysut[i]);
            estx[1][0]=(float)m0/(m0+rr)*ysut[i];
        }
    }
}

```

```

pp[1][0]=(float)m0*rr/(m0+rr);
kk[1][0]=(float)m0/(m0+rr);
fprintf(fm,"%f",m0);
fprintf(fkal,"%f",kk[1][0]);
}
fscanf(fp,"%f",&yd);
pest=(float)(a1*estx[0][j]+a2*estx[1][j-1]
              -a1*a2*estx[0][j-1]);
mm=(float)(a1*a1*pp[0][j]+a2*a2*pp[1][j-1]
            -a1*a1*a2*a2*pp[0][j-1]+q);
estx[1][j]=pest+mm*(yd-pest)/(mm+rr);
pp[1][j]=mm*rr/(mm+rr);
kk[1][j]=mm/(mm+rr);
fprintf(fm,"%f",mm);
fprintf(fkal,"%f",kk[1][j]);
}
for(jj=0;jj<256;jj++){
    laskx[1][jj]=(int)(estx[1][jj]*sqrt+ortyd);
    if(laskx[1][jj]<0) laskx[1][jj]=0;
    if(laskx[1][jj]>255) laskx[1][jj]=255;
    if((i==1) && (jj==0)){
        for(iii=0;iii<256;iii++){
            laskx[0][iii]=(int)(estx[0][iii]*sqrt+ortyd);
            if(laskx[0][iii]<0) laskx[0][iii]=0;
            if(laskx[0][iii]>255) laskx[0][iii]=255;
            for(ii=0;ii<256;ii++){
                fprintf(fs,"%f",estx[0][ii]);
                fprintf(laski,"%o",laskx[0][ii]);}}
            fprintf(fs,"%f",estx[1][jj]);
            fprintf(laski,"%o",laskx[1][jj]);
            pp[0][jj]=pp[1][jj];
            estx[0][jj]=estx[1][jj];
        }
    }
fclose(fs);
fclose(fp);
fclose(fm);
fclose(fkal);
fclose(laski);

if((fl=fopen("lena.img","rb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
if((fs=fopen("son1.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fs,"%f",&tem);
        temp=tem;
        sonort+=(float)tem;
        sonvar+=(float)(tem*tem);
        temp1=(float)(fgetc(fl)-ortyd)/sqrt;
        if(temp==temp1){ printf("iyi kestirim %f\n",temp);
                           getch();}
    }
}

```

```

        hava=(float)(temp1-temp);
        havar+=(float)((hava)*(hava)/65536);
    }
}
fclose(fl);
fclose(fs);
snri=10*log(varydx/r)/log(10);
snro=10*log(1/havar)/log(10);
g=10*log(rr/havar)/log(10);

printf("SNRI=%f   havar=%f   Kazanc=%f\n",snri,havar,g);
printf("SNRO=%f\n",snro);
}
}

void kestirim2()
{
    register int ii,i,j,jj,m,n,k,l,z;
    int tmp,temp2;
    float estx[2][256],ysat[256],ysut[256],yd,pp[2][256],kk[2][256]
    float y[30][256],sr;
    float pest,mm,m0,hava,havar=0.0;
    float tem,temp,temp1;
    float geo,es,ess;
    char fname[20];
    FILE *fp,*fg,*fgr,*fs,*fl,*fsn;

    m0=1.0;

    sr=sqrt(varydx);
    rr=(float)(r/varydx);
    if((fp=fopen("gur.dat","r"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }

    for(i=0;i<M;i++){
        for(j=0;j<256;j++){
            fscanf(fp,"%f",&temp);
            y[i][j]=(float)temp;
        }
    }
    fclose(fp);
    if((fg=fopen("gur.dat","r"))==NULL) {
        printf("Dosya acilamadi");
        exit(0);
    }

    for(i=0;i<256;i++){
        for(j=0;j<256;j++){
            fscanf(fg,"%f",&temp);
            fprintf(fg,"%f",temp);
        }
        if(i==0) ysat[j]=temp;
        if(j==0) ysut[i]=temp;
    }
    }
    fclose(fg);

```

```

if((fsn=fopen("sonn.asc","wb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
if((fg=fopen("gur.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
if((fs=fopen("son2.dat","w"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
printf("Dosyalara girdi:\n");
for(z=0;z<256;z++){
    estx[0][z]=(float)(m0*ysat[z]/(m0+rr));
    pp[0][z]=(float)(m0*rr/(m0+rr));
    kk[0][z]=(float)(m0/(m0+rr));
    tmp=(int)(estx[0][z]*sr+ortyd);
    if(tmp<0) tmp=0;
    if(tmp>255) tmp=255;
    fprintf(fsn,"%c",tmp);
    fprintf(fs,"%f",estx[0][z]);
}
for(j=0;j<256;j++){ fscanf(fg,"%f",&ysat[j]);
    for(i=1;i<256;i++){
        for(j=1;j<256;j++){
            if(j==1) {
                fscanf(fg,"%f",&ysut[i]);
                estx[1][0]=(float)(m0*ysut[i]/(m0+rr));
                pp[1][0]=(float)(m0*rr/(m0+rr));
                kk[1][0]=(float)(m0/(m0+rr));
            }
            for(m=0;m<M;m++){
                for(n=0;n<M;n++){
                    if(((m+i-1)<256) && ((n+j-1)<256)){
                        yy[m][n]=y[m][j-1+n];
                    }else{ if(((m+i-1)>255) && ((n+j-1)>255)){
                        yy[m][n]=y[M-1][255];
                    }else{ if(((m+i-1)>255)){ yy[m][n]=y[M-1][n];
                    }else{ yy[m][n]=y[m][255];}}}}
                }
            }
        }
    }
    blok_oziliski(); /*AR katsayilari geliyor*/
    fscanf(fg,"%f",&y_d);
    pest=(float)(a1*estx[0][j]+a2*estx[1][j-1]
                -a1*a2*estx[0][j-1]);
    mm=(float)(a1*a1*pp[0][j]+a2*a2*pp[1][j-1]
                -a1*a1*a2*a2*pp[0][j-1]+q);
    estx[1][j]=(float)(pest+mm*(y_d-pest)/(mm+rr));
    pp[1][j]=(float)(mm*rr/(mm+rr));
    kk[1][j]=(float)(mm/(mm+rr));
} /* j dongusu */
for(jj=0;jj<256;jj++){
    if((i==1) && (jj==0)){
        for(ii=0;ii<256;ii++) fprintf(fs,"%f",estx[0][ii]);
        temp=(float)((tmp-ortyd)/sqrt(varydx));
    }
}

```

```

        tmp=(int)(estx[1][jj]*sr+ortyd);
        if(tmp<0) tmp=0;
        if(tmp>255) tmp=255;
        fprintf(fs,"%f",estx[1][jj]);
        fprintf(fsn,"%c",tmp);
        pp[0][jj]=pp[1][jj];
        estx[0][jj]=estx[1][jj];
    }
    for(k=0;k<(M-1);k++)
        for(l=0;l<256;l++) y[k][l]=y[k+1][l];
    fseek(fg,((i-1)*256+M*256)*2,SEEK_SET);
    for(m=0;m<256;m++) {
        fscanf(fg,"%f",&temp);
        y[M-1][m]=temp;}
    fseek(fg,((i+1)*256)*2,SEEK_SET);
} /* i dongusu */

fclose(fs);
fclose(fg);
fclose(fsn);
if((fl=fopen("lena.img","rb"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}

if((fs=fopen("son2.dat","r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fs,"%f",&temp);
        temp1=(float)(fgetc(fl)-ortyd)/sqrt(varydx);
        hava=(float)(temp1-temp);
        havar+=(float)((hava)*(hava)/65536);
    }
}
fclose(fl);
fclose(fs);

g=10*log(rr/havar)/log(10);
snri=10*log(varydx/r)/log(10);
snro=10*log(1/havar)/log(10);
printf("Giris S/N=%fdB\n",snri);
printf("Cikis S/N=%fdB\n",snro);
printf("havar=%f\n",havar);
printf("Kazanc=%fdB\n",g);
}

void kal_ciz()
{
    register int i,j;
    int gd=DETECT,gm,err;
    int maxx,maxy,maxcolor;
    float temp,sq;

```

```

char fname1[20],fname2[20];
FILE *fs,*fp;
struct palettetype pal;
char arr1[10],arr2[10];
char snn1[10],snn2[10],gg[10];

sq=sqrt(varydx);
printf("Gurultulu Resim Dosya adi:");
scanf("%s",fname1);
printf("Onarilmis Resim Dosya adi:");
scanf("%s",fname2);

if((fs=fopen(fname1,"r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
if((fp=fopen(fname2,"r"))==NULL) {
    printf("Dosya acilamadi");
    exit(0);
}
clrscr();
initgraph(&gd,&gm,"");
if(graphresult()!=grOk){
    printf("grafik karti hazir degil\n");
    getch();}

getpalette(&pal);
for(i=0;i<pal.size;i++){
    setrgbpalette(pal.colors[i],i*4,i*4,i*4);

for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fs,"%f",&temp);
        temp=(float)(temp*sq+ortyd);
        if(temp>255) temp=255.0;
        if(temp<0) temp=0.0;
        putpixel(j+getmaxx()/2-256,i+getmaxy()/2-128,(int)(temp/16));
    }
}
for(i=0;i<256;i++){
    for(j=0;j<256;j++){
        fscanf(fp,"%f",&temp);
        temp=(float)(temp*sq+ortyd);
        if(temp>255) temp=255.0;
        if(temp<0) temp=0.0;
        putpixel(j+getmaxx()/2+1,i+getmaxy()/2-128,(int)(temp/16));
    }
}
if(ref==1){
    settextstyle(0,0,1);
    govt(snn1,6,snn1);
    govt(a1,6,arr1);
    govt(a2,6,arr2);
    govt(snn2,6,snn2);
    govt(g,6,gg);
    outtextxy(getmaxx()/2-256,getmaxy()/2+138,"SNRI :dB");
}

```

```

outtextxy(getmaxx()/2-256,getmaxy()/2+148,"a1      :");
outtextxy(getmaxx()/2-256,getmaxy()/2+158,"a2      :");
outtextxy(getmaxx()/2,getmaxy()/2+138,"SNRO      :dB");
outtextxy(getmaxx()/2,getmaxy()/2+148,"KAZANC      :dB");
outtextxy(getmaxx()/2-256+100,getmaxy()/2+138,snn1);
outtextxy(getmaxx()/2-256+100,getmaxy()/2+148,arr1);
outtextxy(getmaxx()/2-256+100,getmaxy()/2+158,arr2);
outtextxy(getmaxx()/2+100,getmaxy()/2+138,snn2);
outtextxy(getmaxx()/2+100,getmaxy()/2+148,gg);
}
if(ref==2){
settextstyle(0,0,1);
govt(snn1,6,snn1);
govt(snn2,6,snn2);
govt(g,6,gg);
outtextxy(getmaxx()/2-256,getmaxy()/2+138,"SNRI      :dB");
outtextxy(getmaxx()/2,getmaxy()/2+138,"SNRO      :dB");
outtextxy(getmaxx()/2,getmaxy()/2+148,"KAZANC      :dB");
outtextxy(getmaxx()/2-256+100,getmaxy()/2+138,snn1);
outtextxy(getmaxx()/2+100,getmaxy()/2+138,snn2);
outtextxy(getmaxx()/2+100,getmaxy()/2+148,gg);
}
getch();
closegraph();
fclose(fs);
fclose(fp);
}

/* Error routine*/
void nrerror(error_text)
char error_text[];
{
void exit();
fprintf(stderr,"run-time error...\n");
fprintf(stderr,"%s\n",error_text);
fprintf(stderr,"....simdi sisteme cikiliyor....\n");
exit(1);
}

/* ran function generates uniform random number */
float ran1(idum)
int *idum;
{
static long ix1,ix2,ix3;
static float rrr[98];
float temp;
static int iff=0;
int j;
void nrerror();

if(*idum < 0 || iff==0) {
    iff=1;
    ix1=(IC1-(*idum)) % M1;
    ix1=(IA1*ix1+IC1) % M1;
    ix2=ix1 % M2;
    ix1=(IA1*ix1+IC1) % M1;

```

```

ix3=ix1 % M3;
for(j=1;j<=97;j++){
    ix1=(IA1*ix1+IC1) % M1;-
    ix2=(IA2*ix2+IC2) % M2;
    rrr[j]=(ix1+ix2*RM2)*RM1;
}
*idum=1;
}
ix1=(IA1*ix1+IC1) % M1;
ix2=(IA2*ix2+IC2) % M2;
ix3=(IA3*ix3+IC3) % M3;
j=1+((97*ix3)/M3);
if(j > 97 || j < 1) nrerror("RAN1: olamaz..");
temp=rrr[j];
rrr[j]=(ix1+ix2*RM2)*RM1;
return temp;
}

/*gauss gurultu uretecı Box-Muller metoduyla*/
float gasdev(idum)
int *idum;
{
    static int iset=0;
    static float gset;
    float fac,g,v1,v2;
    float ran1();

    if(iset==0){
        do{
            v1=2.0*ran1(idum)-1.0;
            v2=2.0*ran1(idum)-1.0;
            g=v1*v1+v2*v2;
        }while(g>=1.0);
        fac=sqrt(-2.0*log(g)/g);
        gset=v1*fac;
        iset=1;
        return v2*fac;
    }else{
        iset=0;
        return gset;
    }
}

```

ÖZGEÇMİŞ

Yaşar Becerikli 1969 yılında Gemlik 'te doğdu. İlk ve orta öğrenimini Gemlik 'te tamamladı. 1987 yılında Yıldız Üniversitesi Kocaeli Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü 'ne girdi. 1991 yılında bu bölümden mezun oldu. Aynı yıl İTÜ Elektronik ve Haberleşme Anabilim Dalı 'nda yüksek lisansa başladı.

Yaşar Becerikli halen, Sakarya Üniversitesi Mühendislik Fakültesi Haberleşme Anabilim Dalı 'nda araştırma görevlisi olarak çalışmaktadır.