

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DİJİTAL GÖRÜNTÜ İŞLEME

Matematik Müh. Murat GÜMÜŞER

**F.B.E Matematik Mühendisliği Anabilim Dalında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı

: Yrd. Doç. Dr. İbrahim EMİROĞLU

İ.Emiroğlu

Prof. Dr. Mehmet Aklatçioğlu

MA

Prof. Dr. Mustafa Sivri

MS

İSTANBUL, 2001

**Y.E. YÜKSEK ÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

106190

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
ŞEKİL LİSTESİ.....	v
ÇİZELGE LİSTESİ.....	vi
ÖNSÖZ.....	vii
ÖZET	viii
ABSTRACT.....	ix
1 GİRİŞ	1
2 DİJİTAL GÖRÜNTÜNÜN TEMELLERİ	4
2.1 Siyah-Beyaz Görüntü.....	4
2.2 Dijital Görüntü.....	5
2.3 Renkli Görüntü	6
2.3.1 Renk Bilgisi	6
2.3.2 Renk Modelleri	9
2.3.2.1 RGB Renk Modeli	9
2.3.2.2 CMY Renk Modeli	10
2.3.2.3 YIQ Renk Modeli	10
2.4 Bitmap dosyaları.....	11
3 GÖRÜNTÜ GELİŞTİRME.....	17
3.1 Nokta İşleme Teknikleri.....	18
3.1.1 Negatif Görüntü.....	18
3.1.2 Gri Ton Değerlerini Dilimleme	19
3.1.3 Gri Ton Ölçeğini Sıkıştırma	20
3.1.4 Görüntü Histogramı	21
3.1.5 Parlaklık Ayarlama	23
3.1.6 Kontrast Geliştirme.....	24
3.1.7 Histogram Eşitleme.....	27
3.1.8 Renkli Görüntülerin Histogramı	28
3.1.9 Eşikleme.....	28
3.2 Maske İşleme Teknikleri.....	30
3.2.1 Görüntü Aritmetiği	31
3.2.2 Lineer Filtreleme	33
3.2.3 Sıralamaya Dayanan Filtreler	35
3.2.4 Kenar Belirleme.....	39
4 MORFOLOJİ.....	43
4.1 Mantıksal İşlemler	43

4.2	Pikseller Arasında İlişkiler	45
4.2.1	Komşuluk	45
4.2.2	Bağılılık	45
4.2.3	Bölge	46
4.3	İnceltme	49
4.4	Morfolojik İşlemlerde Kullanılan Temel Kavramlar	50
4.5	Genleşme ve Aşınma	51
4.6	Açılım ve Kapanım	54
5	SONUÇLAR	56
	KAYNAKLAR	58
	ÖZGEÇMİŞ	59



SİMGE LİSTESİ

$\lfloor X \rfloor$	X'in tam değeri
∇f	Bir f fonksiyonunun gradienti
$ \nabla f $	Fonksiyonun gradientinin magnitudü
Δ	Mantıksal işlem operatörü
$N_4(p)$	Bir p noktasının 4-komşuluğu
$N_D(p)$	Bir p noktasının d-komşuluğu
$N_8(p)$	Bir p noktasının 8-komşuluğu
$A \cap B$	A ve B kümelerinin kesişimi
$N(p_1)$	p_1 noktasının sıfıra eşit olmayan komşularının sayısı
$S(p_1)$	p_1 noktasının komşuluklarında 0-1 geçişlerinin sayısı
$(A)_x$	A'nın x kadar ötelenmesi
A^c	A'nın tümleyeni
$\hat{B}$	B'nin yansıması
$A \oplus B$	A'nın B ile genişmesi
$A \ominus B$	A'nın B ile aşınması
$A \circ B$	A'nın B ile açılımı
$A \bullet B$	A'nın B ile kapanımı

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1	Renk karıştırma işlemleri	8
Şekil 2.2	RGB renk kübü.....	9
Şekil 3.1	Negatif görüntü için dönüşüm fonksiyonu	18
Şekil 3.2	İlgi alanına girmeyen tonların düşük değere çekildiği fonksiyon	19
Şekil 3.3	İlgi alanına girmeyen tonların değişmediği fonksiyon	19
Şekil 3.4	Gri ton ölçeğini sıkıştırmak için kullanılan fonksiyon örneği	20
Şekil 3.5	Okul; örnek bir görüntü	22
Şekil 3.6	Okul görüntüsünün histogramı	22
Şekil 3.7	$c < 0$ için dönüşüm fonksiyonu	24
Şekil 3.8	$c > 0$ için dönüşüm fonksiyonu	24
Şekil 3.9	Karanlık görüntüler için dönüşüm fonksiyonu	25
Şekil 3.10	Aydınlık görüntüler için dönüşüm fonksiyonu	25
Şekil 3.11	Gri görüntüler için dönüşüm fonksiyonu	26
Şekil 3.12	Renkli görüntüler için geliştirme işlemi şeması	28
Şekil 3.13	Eşikleme için dönüşüm fonksiyonu	29
Şekil 3.14	Örnek bir histogram	30
Şekil 3.15	Maske modeli	31
Şekil 3.16	Alçakgeçiren filtre örnekleri	33
Şekil 3.17	Yüksekgeçiren filtre örnekleri	34
Şekil 3.18	High-boost filtre için maske	35
Şekil 3.19	Şekiller; orijinal test görüntüsü	38
Şekil 3.20	Görüntünün (Şekil 3.19) siyah-beyaz beneklerle bozulmuş hali	38
Şekil 3.21	Ortalama filtre uygulamasının sonucu	38
Şekil 3.22	Kesilmiş ortalama filtre uygulamasının sonucu	38
Şekil 3.23	Medyan filtre uygulamasının sonucu	38
Şekil 3.24	Çizgi belirleme maskeleri	39
Şekil 3.25	Prewitt operatörleri	41
Şekil 3.26	Sobel operatörleri	41
Şekil 4.1	Görüntüler üzerinde mantıksal işlemler	44
Şekil 4.2	Piksel komşulukları	45
Şekil 4.3	Pikseller arasında yollar	46
Şekil 4.4	Örnek bir bölge	47
Şekil 4.5	İnceltme işlemi için piksel numaralaması	49
Şekil 4.6	A harfi; örnek bir nesne ve incelmış hali	50
Şekil 4.7	Kare ve artı yapılandırma elemanları	51
Şekil 4.8	Artı yapılandırma elemanı ile genişleme	52
Şekil 4.9	B harfi nesnesinin kare yapılandırma elemanı ile genişmesi	52
Şekil 4.10	Artı yapılandırma elemanı ile aşınma	53
Şekil 4.11	B harfi nesnesinin kare yapılandırma elemanı ile aşınması	53
Şekil 4.12	A harfi nesnesinin iç sınırları	54

ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1	Mantıksal operatörlerin tanımları.....	44
-------------	--	----



ÖNSÖZ

Bu tezin hazırlanmasında büyük yardımları bulunan değerli hocam Yrd. Doç. Dr. İbrahim Emiroğlu'na teşekkür ederim.

Haziran 2001

Murat Gümüşer



ÖZET

Dijital görüntü işleme bir görüntünün dijital formata dönüşümü ve bilgisayarla işlenmesi konularını içerir. Bu çalışma görüntü işleme metotlarına bir giriş niteliğindedir. Görüntü gösterme ve saklama ortamı olarak bitmap görüntüleri ve dosyaları seçilmiştir. Araştırma alanı ağırlıklı olarak gri ton değerli görüntüler ve ikilik görüntüler üzerine odaklanmıştır. Uygulamalar için, verimli çalışmaları ve hesapları hızlı yapabilmeleri nedeniyle C ve C++ bilgisayar programlama dilleri kullanılmıştır. Çalışmada ikilik görüntüler için matematiksel morfolojinin temel kavramları tanıtılmıştır. Gri tonlu görüntüler için kontrast geliştirme, gürültü giderme, çizgi ve kenar belirleme, ve eşikleme algoritmaları incelenmiştir. Histogram eşitleme ve filtreleme gibi çeşitli teknikler test görüntüleri üzerinde uygulanmış ve sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler: Gri ton değerli görüntü, ikilik görüntü, kontrast geliştirme, gürültü giderme, eşikleme.



ABSTRACT

Digital image processing concerns the transformation of an image to a digital format and its processing by digital computers. This work is an introduction to image processing methods. Bitmap images and files were chosen as image representation and storage means. The research area is mainly focused on binary and grey-scale images. For the applications, C and C++ computer programming languages were used because of their efficiency and computational speed. The basic concepts of mathematical morphology for binary images were introduced. For grey-scale images, the algorithms for contrast enhancement, noise reduction, line and edge detection, and thresholding were studied. Various techniques such as histogram equalization and filtering were applied on test images and the results were compared.

Keywords: Grey-scale image, binary image, contrast enhancement, noise reduction, thresholding.



1.GİRİŞ

Görsel bilgilerin sağladığı geniş anlatım ve belgeleme olanakları 1830'lu yıllarda ilk fotoğrafların ortaya çıkmasıyla birlikte görüntüyü elde etme, saklama ve iletme konularında bir çok çalışma yapılmasına neden olmuştur. İlk elektriksel görüntü iletme sistemi 1920'li yılların başında kullanıma giren Bartlane kablolu resim iletme sistemidir. Bu sistem sayesinde Atlantik Okyanusu'nun bir kıyısından diğer kıyısına üç saat içinde bir fotoğraf göndermek olanaklı hale gelmiştir. Bu yıllarda çözümü aranan problemler, parlaklık seviyelerinin dağılımı ve baskı kalitesi ile ilgili problemlerdi. Bartlane sistemi ile, beş farklı parlaklık seviyesinde kodlanmış resimler iletilebiliyordu. Bu kapasite 1930 yılında 15 farklı parlaklık seviyesine yükseldi. Takip eden 35 yıl boyunca iletim ve baskı metotları sürekli olarak gelişti. Buna rağmen görüntü işleme uygulamalarının potansiyeli ancak dijital bilgisayarların uzay programında kullanımıyla gündeme geldi.

Modern anlamda ilk görüntü işleme uygulaması, 1964 yılında ABD California'da Jet Propulsion Laboratory'de, Ay'ın çevresinde dönen bir uydudan gönderilen görüntülerdeki bozulmaların ve elektronik gürültülerin ortaya çıkardığı lekelerin bilgisayar yoluyla ortadan kaldırılmasıdır. Bu teknikler daha sonra Ay'dan, Mars'dan ve bir çok uzay görevinden elde edilen görüntülerin işlenmesinde kullanılan tekniklerin temelini oluşturmuştur. Görüntü işleme yalnız uzay programı ile sınırlı kalmadı. Tıp alanında (röntgen filmlerinin daha çok ayrıntı gösterir hale getirilmesi), arkeolojide (uçaktan veya uydudan çekilen fotoğraflardaki silik detayların ortaya çıkarılması), fizik bilimde (elektron mikroskopundan elde edilen görüntülerin iyileştirilmesi), ve daha bir çok dalda kullanım alanı buldu (Gonzalez ve Woods, 1993).

Dijital görüntü işleme alanı bir görüntüyü dijital hale dönüştürme ve bu görüntü üzerinde bilgisayarla işlem yapma konularını kapsar. Dijital görüntü işleme yöntemlerinin iki ana amacı vardır:

1-İnsanın görüntüyü daha iyi yorumlayabilmesi için görüntünün iyileştirilmesi,

2-Bilgisayarın görüntüyü yorumlayabilmesi için görüntünün analizi.

Görüntü iyileştirme tekniklerinin hedefi görüntüyü özel bir amaç için orijinal görüntüden daha uygun bir biçime getirmektir. Bir görüntü iyileştirme sisteminin hem girdisi hem de çıktısı dijital resimlerdir. Kullanılan teknikler genellikle probleme bağlıdır. Dijital görüntü analizi ise

görüntünün içeriğini tanıma ve tanımlandırma konularını kapsar. Burada girdi dijital bir resimdir. Çıktı ise görüntünün içeriğini anlatmaya yarayan çeşitli semboller veya kodlardır.

Görüntü işleme tekniklerinin, uydudan gönderilen görüntülerin veya röntgen filmlerinin iyileştirilmesi gibi ilk uygulamaları genellikle görüntü iyileştirme kısmına giren işlemlerdir. Bunlardan farklı olarak doküman analizi (otomatik karakter tanıma), üretim ve kontrol amaçlı endüstri makinalarının optik algılayıcılar yardımıyla işlemesi, bilgisayarla parmak izi tanıma gibi uygulamalar (Emiroğlu, 1997) görüntü analizi kısmına girerler.

Bir görüntü işleme uygulamasında ilk adım görüntüyü elde etmektir. Bunun için bir görüntü algılayıcı gerekir. Bu alet bir tarayıcı, dijital fotoğraf makinası veya bir televizyon kamerası olabilir. Eğer aletin çıktısı sayısal değilse bu sinyal bir analog-sayısal çeviriciden geçirilerek sayısallaştırılır. Bundan sonra ön işleme aşaması gelir. Burada amaç, sonraki aşamaların başarı şansını artırmak amacıyla görüntüyü geliştirmektir. Kontrast geliştirme, eşikleme, görüntüyü filtreden geçirme bu kısımda yapılan tipik işlemlerden bazılarıdır. Görüntü iyileştirme uygulamaları genellikle bu aşamanın ötesine geçmez. Ön işleme aşamasında hem girdi, hem de çıktı sayısal görüntülerdir. Bu aşamadan sonra segmentasyon aşaması gelir. Burada görüntü, kendi içinde anlamlı bir birlik oluşturan parçalara ayrılır. Görüntü işleme uygulamalarının en zor kısmı otonom segmentasyondur. Bu aşamanın başarısı, uygulamanın başarısını doğrudan etkiler. Segmentasyon kısmının çıktısı ham piksel verileridir. Uygulamanın özelliklerine göre bu pikseller, belli bir bölgenin bütün pikselleri veya sadece bölgenin sınırlarını gösteren pikseller olabilirler. Her iki durumda da bu pikseller ile bunların ifade ettiği düşünülen nesneler arasında bir ilişki kurmak gerekir. Bu işlem, bundan sonraki iki aşamada gerçekleştirilir: betimleme (veya özellik seçme), tanıma ve yorumlama. Betimleme aşamasında, söz konusu piksellerin taşıdığı ayırt edici özellikler hesaplanır. Bu aşamanın çıktısı, bir piksel kümesinin özelliklerini içeren bir özellik vektörü olabilir. Tanıma ve yorumlama aşamasının tanıma bölümünde bu özellik vektörü, bir özellik vektörleri kümesinin herhangi bir elemanına benzetilmeye çalışılır. Böylece piksel kümeleri sınıflandırılır. Yorumlama bölümünde ise sınıflandırılması yapılan nesnelerin anlamları belirlenir ve çıktı olarak verilir (Gonzalez ve Woods, 1993).

Bu tez çalışması dijital görüntü işleme konusuna giriş niteliğindedir. Çalışmanın amacı, görüntü işlemede kullanılan kavramları tanıtmak ve daha ileri düzeydeki uygulamalara bir temel oluşturmaktır. Bunun için önce ikinci bölümde, işlenecek görüntünün bilgisayarda gösterimi ve saklanmasıyla ilgili konular ayrıntılı olarak anlatılmıştır. Üçüncü bölümde hem insanın

kullanımına, hem de bilgisayarın kullanımına yönelik görüntü geliştirme uygulamaları tartışılmış ve pratikte kullanılan çeşitli yöntemler verilmiştir. Dördüncü bölümde genellikle ikilik görüntülerin segmantasyonu aşamasında kullanılan temel morfolojik yöntemler tanıtılmıştır. Beşinci bölümde ise, çeşitli tekniklerin geliştirilmesi ve uygulanması sonucunda ortaya çıkan durumlar değerlendirilmiştir. Dijital görüntü işleme tekniklerinin bilgisayar uygulamalarında, hızlı ve verimli çalışma özellikleri göz önüne alınarak, C ve C++ bilgisayar programlama dilleri kullanılmıştır (Shammas, 1994) .



2.DİJİTAL GÖRÜNTÜNÜN TEMELLERİ

2.1 Siyah-Beyaz Görüntü

Siyah beyaz bir görüntü, ışık yoğunluğunu gösteren iki boyutlu bir fonksiyon olarak düşünülebilir. Bu $f(x,y)$ fonksiyonunda x ve y düzlemsel koordinatları, fonksiyonun (x,y) noktasındaki değeri resmin o noktadaki parlaklığını gösterir. Siyah beyaz bir resmin (x,y) noktasındaki parlaklığına resmin bu noktadaki gri ton değeri (grey level) adı verilir. Teoride $x, y \in R$ 'dir ve $f(x,y)$ sıfırdan büyük, sonlu bir sayıdır. $f(x,y)$ 'nin iki bileşeni vardır:

- 1) Ortamdaki ışığın miktarı (aydınlanma),
- 2) Nesnenin ışığı yansıtma oranı (yansıtma).

Aydınlanmayı $i(x,y)$, yansıtmayı $r(x,y)$ ile gösterirsek $f(x,y)$ 'yi bunların çarpımı şeklinde yazabiliriz:

$$f(x,y) = i(x,y)r(x,y). \quad (2.1)$$

Burada teorik olarak

$$0 < i(x,y) < \infty \quad (2.2)$$

ve

$$0 < r(x,y) < 1 \quad (2.3)$$

kabul edilir. $i(x,y)$ 'yi ışık kaynağı, $r(x,y)$ 'yi nesnelerin yansıtma özellikleri belirler. $r(x,y)$ için 0 limit değeri ışığın tamamen absorbe edilmesi, 1 limit değeri ışığın tamamen yansıtması anlamına gelir. Aydınlanma ortalama olarak açık bir günde 9000, bulutlu günde 1000, tipik bir ofiste 100 foot-kandeladır. Yansıtma için ortalama olarak bulunan bazı değerler en koyu cisim için 0.01, paslanmaz çelik için 0.65, kar için 0.93'tür (Gonzalez ve Woods, 1993).

$f(x,y)$ fonksiyonunun (x,y) noktasındaki değerine λ dersek (2.1), (2.2) ve (2.3) denklemlerinden λ 'nın

$$L_{\min} < \lambda < L_{\max} \quad (2.4)$$

aralığında değiştiğini görürüz. Teoride tek gereklilik $L_{\min}$ 'in pozitif olması ve $L_{\max}$ 'ın sonlu olmasıdır. Uygulamalarda, $L_{\min} = i_{\min} r_{\min}$ ve $L_{\max} = i_{\max} r_{\max}$ yazarsak ve yukarıdaki değerleri göz önüne alırsak, kapalı mekanlar için $L_{\min} \cong 0.005$ ve $L_{\max} \cong 100$ bulunur. $[L_{\min}, L_{\max}]$ aralığına gri ton ölçeği (gray scale) adı verilir. Pratikte bu aralık $[0, L]$ aralığına kaydırılır. Burada $\lambda = 0$ siyah ve $\lambda = L$ beyaz olarak kabul edilir. Aradaki değerler grinin tonlarına karşılık gelir (Gonzalez ve Woods, 1993).

2.2 Dijital Görüntü

Resim üzerinde bilgisayarla işlem yapabilmek için $f(x,y)$ görüntü fonksiyonunun koordinat sisteminin ve genliğinin dijitize edilmesi gerekir. (x,y) düzlemsel koordinat sisteminin sayısallaştırılmasına görüntü örnekleme (image sampling), genliğin sayısallaştırılmasına gri ton ayrıklaştırması (gray level quantization) adı verilir. Bu durumda dijital resim hem düzlemsel kordinatların, hem de parlaklığın ayrık biçimde olduğu bir görüntü fonksiyonudur.

$$f(x,y) \cong \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & \dots & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.5)$$

(2.5) denkleminde $f(x,y)$ fonksiyonu, sürekli bir resmi göstermektedir. Denklemin sağ tarafı, bu resmin yaklaşığıdır ve dijital resim olarak adlandırılır. Bu dijital resmi, satır ve sütun indislerinin resimdeki bir noktanın koordinatlarını, bu noktadaki elemanın da gri ton değerini gösterdiği bir matris olarak düşünebiliriz ($i=0,1,\dots,N-1$; $j=0,1,\dots,M-1$). Bu şekildeki bir görüntü düzleminin elemanlarına piksel adı verilir (Gonzalez ve Woods, 1993).

Sayısallaştırma işleminde N , M değerlerini ve ayrık gri ton sayısını belirlemek gerekir. Gri tonlarının sayısına G diyelim. Bilgisayar uygulamalarında bu değerler genellikle 2'nin pozitif tamsayı kuvvetleri olarak seçilir:

$$N = 2^n, M = 2^m, G = 2^k. \quad (2.6)$$

Burada gri ton değerlerinin $[0, L]$ aralığında ve eşit aralıklı ayrık değerler olduğunu kabul ediyoruz. Resimler bilgisayar ortamında işlem gördüğüne ve saklandığına göre, bir resmi saklamak için $N \times M \times k$ bit hafızaya ihtiyaç vardır. Görüntü matrisi sürekli bir görüntü fonksiyonunun yaklaştığı olduğundan iyi bir benzerlik için matrisin boyutlarının ve gri ton sayısının ne olması gerektiği sorunuyla karşılaşırız. Bu parametreler büyüdükçe dijital resim orijinal resme daha çok yaklaşır. Ancak resim için gereken hafıza miktarı ve görüntü işleme için gereken zaman da artacaktır. İyi bir resmin tanımını yapmak zordur çünkü çeşitli uygulama alanlarının ihtiyaçları birbirinden farklıdır. Siyah beyaz televizyon görüntüsüne denk bir görüntü 512×512 piksel boyutlarında 128 gri tonlu bir resimle karşılanabilir

2.3 Renkli Görüntü

Görüntü işlemede renkli resim kullanımının çeşitli avantajları vardır. Otomatik görüntü analizinde renk, nesnelerin tanımlanmasını ve çevrelerinden izole edilmesini kolaylaştıran güçlü bir araçtır. Ayrıca, insan gözü yaklaşık iki düzine gri tonu ayırt edebilmesine karşılık binlerce renk tonu ve yoğunluğu ayırt edebilir.

2.3.1 Renk Bilgisi

1666'da Sir Isaac Newton güneş ışığını bir prizmadan geçirerek renkli bileşenlerine ayırdı ve mordan kırmızıya uzanan sürekli bir tayf elde etti. Bunu ikinci bir prizmadan geçirerek yeniden beyaz ışık elde etti. Bu tayfta yedi temel rengi (mor, lacivert, mavi, yeşil, sarı, turuncu, kırmızı) belirleyen de Newton'dur. Burada, renk olarak adlandırılan niteliğin ışık demetinde var olan bir nitelik olmadığını, görme organı aracılığıyla ışığın dalgaboyuna bağlı olarak beyinde oluşan bir duyum olduğunu belirtmek gerekir.

Fizikte renk, elektromagnetik ışınım tayfinin insan gözünün algılayabildiği bölgesinde yer alan dalgaboylarıyla ilgilidir. Elektromagnetik ışınım tayfinin görünür bölgesindeki ışınım, ışık olarak adlandırılır. Bu bölge, tayfin $400 - 700 \text{ nm}$ ($1 \text{ nm} = 10^{-9} \text{ m}$) arasındaki çok dar bir dalgaboyu aralığını kaplar. Bu aralıktaki dalgaboyları göz tarafından çeşitli renklerde algılanır. Bir nesnenin yansıttığı ışık, bütün görünür dalgaboylarında görelî olarak dengeliyse

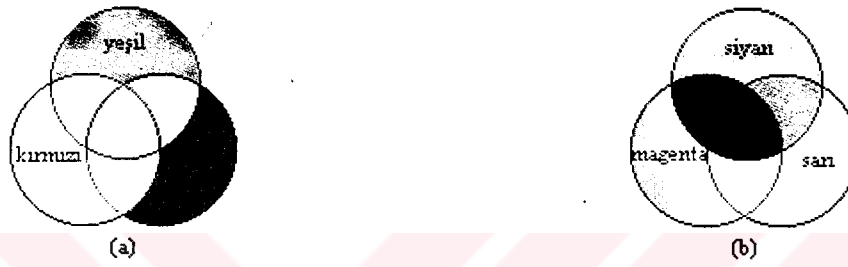
bu nesne beyaz renkte görünür. Görünür tayfin belirli bir bölgesini daha çok yansıtan bir nesne, bu dalgaboyuna denk düşen bir renkte görünür. Örneğin yeşil renkte görünen nesneler 500 - 570 nm dalgaboyu aralığında ışık yansıtip diğer dalgaboylarındaki enerjinin çoğunu absorbe ederler.

Renkler birbirlerinden üç nitelik ile ayırt edilir; bunlar renk tonu (renk türü), doygunluk ve parlaklıktır. Renk tonu, rengi oluşturan ışığın tayftaki yerine, bir başka deyişle baskın dalgaboyuna bağlı bir niteliktir. Günlük dildeki kırmızı, pembe, yeşil, vb terimler tonla ilgilidir. Doymunluk, rengi oluşturan ışıktaki beyaz ışık miktarına ilişkin bir niteliktir. Doymunluk derecesi, eklenen beyaz ışık miktarıyla ters orantılıdır. Tayfin saf renkleri tamamen doymundur. Pembe (kırmızı ve beyaz) veya eflatun (mor ve beyaz) gibi renklerin doymunluğu daha azdır. Ton ve doymunluk birlikte bir rengin renkserlik (chromaticity) olarak adlandırılan niteliğini oluşturlar. Renkler, renkserlikleri ve parlaklıkları ile belirlenebilirler.

Newton'dan bu yana renkleri tanımlama ve sınıflandırma yönünde pek çok çalışma yapılmıştır. Renk ölçümünde kullanılan en yaygın sistem Uluslararası Aydınlatma Komisyonu'nun (CIE - Comission Internationale d'Eclairage) 1931'de kabul ettiğı sistemdir. 1964'de değışikliklerle yeniden kabul edilen CIE sisteminde birincil renkler olarak belirlenen kırmızı, yeşil ve maviye karşılık gelen üç değeri, temel karşılaştırma noktaları olarak alınır. Birincil renkler için standart olarak kırmızı=700 nm, yeşil=546.1 nm, mavi=435.8 nm dalgaboyu değeri kabul edilmiştir. Diğer renkler bu birincil renklerden elde edilir. Burada, tayfta mutlak şekilde kırmızı, yeşil veya mavi olarak adlandırılabilcek renkler olmadığına dikkat etmek gerekir. Bundan dolayı standardizasyon amacıyla üç tane dalgaboyu belirlenmesi, bu üç sabit bileşen yardımıyla tayftaki bütün renklerin elde edilebileceğı anlamına gelmez. Tayfin bütün renklerini üç temel renk ile oluşturmak için dalgaboyunun değışken olması gerekir (Gonzalez ve Woods, 1993).

Herhangi bir renk tonu, birincil renklerden ikisinin uygun oranlarda karıştırılması ile elde edilebilir. Karıştırma işlemi toplamalı ya da çıkarmalı olarak gerçekleştirilir. Toplamalı işlemde tayfin değışik bölgeleri birbirine eklenir. İki birincil renk toplanarak ışığın ikincil renkleri elde edilir: magenta (kırmızı ve mavi), siyan (yeşil ve mavi) ve sarı (kırmızı ve yeşil). Üç birincil rengin uygun oranlarda karışımı beyaz renk verir. Birincil ışık renklerinin toplanarak farklı renk tonları elde edilmesinin bir örneğı renkli televizyondur.

Çıkarmalı işlemde ise tayfin bir bölümünün ortadan kaldırılması ya da soğurulması söz konusudur. Çıkarmalı karıştırma işlemi, renk verici pigmentlerin kullanıldığı renkli baskı işlerinde kullanılır. Buraya kadar olan bölümde renk kelimesinin, ışığın renklerini ifade ettiğine dikkat etmek gerekir. Işığın üç birincil rengi ile renklendirici pigmentlerin üç birincil rengi birbirlerinden farklıdır. Pigmentler söz konusu olduğunda birincil renk, ışığın bir birincil rengini absorbe eden ve diğer iki birincil rengini yansıtan renk olarak tanımlanır. Bundan dolayı pigmentlerin birincil renkleri magenta, siyan ve sarıdır. Pigmentlerin üç birincil renginin uygun oranlarda karıştırılması ile siyah renk elde edilir.



Şekil 2.1 (a) Toplamalı karıştırma işlemi; (b) çıkarmalı karıştırma işlemi

Herhangi bir rengi oluşturmak için gereken kırmızı, yeşil ve mavi miktarlarına tristimulus değerleri adı verilir. Bu değerler sırasıyla X, Y ve Z ile gösterilir. Bir renk, aşağıdaki şekilde tanımlanan trikromatik katsayıları ile belirlenir:

$$x = X / (X + Y + Z) \quad (2.7)$$

$$y = Y / (X + Y + Z) \quad (2.8)$$

$$z = Z / (X + Y + Z) \quad (2.9)$$

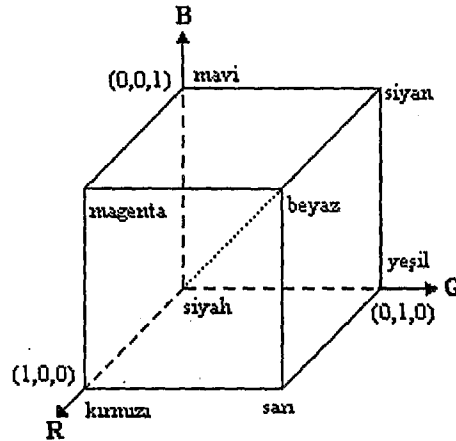
Görünür tayfin her ışık dalgaboyu için, bu dalgaboyuna karşılık gelen rengi üretmek için gereken tristimulus değerleri, bir çok deneysel sonuçtan derlenen eğrilerden veya tablolardan öğrenilebilir.

2.3.2 Renk Modelleri

Renklerin gösteriminde standart sağlamak amacıyla çeşitli renk modelleri oluşturulmuştur. Bir renk modeli, üç boyutlu bir koordinat sisteminin ve bu sistem içinde her bir rengin tek bir nokta ile gösterilebileceği bir alt uzayın belirlenmesidir. Günümüzde kullanılan renk modellerinin çoğunluğu donanıma (renkli monitör ve yazıcı) veya renkli görüntü işlemeye (animasyon için renkli grafik oluşturma) yöneliktir. Donanıma yönelik modeller arasında en çok kullanılanlar renkli monitörlerde ve video kameralarında kullanılan RGB modeli, renkli yazıcılar için CMY modeli ve renkli televizyon yayıncılığında standart olan YIQ modelidir.

2.3.2.1 RGB Renk Modeli

RGB modelinde her renk tayfin kırmızı, yeşil ve mavi birincil bileşenlerinden meydana gelir. Bu modelin temeli kartezyen koordinat sistemidir. Renk altuzayı, Şekil 2.2'de görülen küptür. Orijin noktasında siyah, küpün orijinden en uzak noktasında beyaz bulunur. Birincil renkler, eksenler üzerindeki üç köşededir. Diğer üç köşede ikincil renkler olan magenta, siyan ve sarı bulunur. Gri ton ölçeği, siyahtan beyaza giden doğru üzerinde yer alır. Renkler küpün üzerindeki ve içindeki noktalara karşılık gelir (Gonzalez ve Woods, 1993).



Şekil 2.2 RGB renk küpü

RGB renk modelinde görüntüler, her biri bir adet birincil renge karşılık gelmek üzere, toplam üç adet renk düzleminden oluşur. Resimde (x,y) koordinatlarındaki bir pikselin rengi, her bir renk düzleminin (x,y) koordinatlarındaki değerinin ekranda toplamalı işlemle karışmasıyla

belirtilir. Modelde bütün renk değerleri $[0,1]$ aralığında normalize edilmiştir. Bilgisayar uygulamalarında ise renk değerleri pozitif tamsayılar cinsinden hesaplanır. Bir renk düzlemindeki renk değeri sayısı, bu düzlemdeki renk değerleri için kaç bit yer ayrıldığına bağlıdır. Örneğin kırmızı için 8 bit yer ayrılmışsa her pikselin kırmızı bileşeni 0 ile 255 arasındaki toplam 256 değerden ($2^8=256$) birini alabilir. Pratikte en çok karşılaşılan durum her bileşen için 8 bit yer ayrılmasıdır. Böylece bir piksel hafızada 24 bit yer kaplar.

Renkli monitörler, video kameraları ve dijital televizyonlar RGB renk modelinin kullanıldığı yerlere örnek olarak gösterilebilir. Ayrıca bilgisayarlar da renk bilgilerini bu modele uygun biçimde saklarlar. Modelin dezavantajı, renkli görüntü işleme uygulamaları için elverişli olmamasıdır.

2.3.2.2 CMY Renk Modeli

CMY (cyan magenta yellow) renk modelinde renkler, renklendirici pigmentlerin birincil renkleri olan siyan, magenta ve sarı renklerin çıkarmalı karışımından oluşur. Kağıt üzerine renklendirici pigment bırakan pek çok alet (renkli yazıcı veya kopyalayıcı) CMY renk modelini kullanır. Bu aletler CMY cinsinden veri girişine ihtiyaç duyar veya kendisi RGB'den CMY modeline dönüşüm yapar. Bu dönüşüm basit bir denklemle gerçekleştirilir:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.10)$$

Burada bütün renk değerlerinin $[0, 1]$ aralığında olduğu kabul edilmiştir (Pitas, 1993).

2.3.2.3 YIQ Renk Modeli

YIQ renk modeli renkli televizyon yayıncılığında kullanılır. Aslında bu model veri iletiminde verimlilik sağlamak ve renkli yayınların siyah beyaz televizyonlarda da seyredilebilmesi amacıyla RGB'nin değişik biçimde kodlanmış halidir. Modelin Y bileşeni siyah beyaz televizyon için gereken ışık yoğunluğu verisini içerir. I ve Q bileşenleri, sırasıyla inphase ve

quadrature olarak adlandırılan kromatik bileşenlerdir. RGB'den YIQ modeline dönüşüm aşağıdaki şekilde tanımlanmıştır (Gonzalez ve Woods, 1993):

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.11)$$

Ters dönüşüm formülü:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.620 \\ 1 & -0.272 & -0.647 \\ 1 & -1.108 & 1.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (2.12)$$

YIQ modeli insan gözünün parlaklıktaki değişikliklere, renk tonu ve doygunluktaki değişikliklere oranla daha hassas olmasından yararlanmak için tasarlanmıştır. Bu yüzden veri iletiminde Y bileşeni için ayrılan bant genişliği, I ve Q bileşenleri için ayrılan bant genişliklerinden daha fazladır (Gonzalez ve Woods, 1993).

YIQ renk modelinin görüntü işleme açısından önemi, bu modelde parlaklık ve renk bilgilerinin birbirlerinden bağımsız olmalarıdır. Böylece görüntünün renk içeriğini değiştirmeden sadece parlaklık bileşeni üzerinde işlem yapma olanağı doğar.

2.4 Bitmap Dosyaları

Herhangi bir şekilde (dijital fotoğraf makinası ile, tarayıcıyla, vs.) elde edilen dijital resmi saklamak gerekecektir. Bilgisayar ortamında bu iş için manyetik veya dijital kayıt ortamlarına yazılan dosyalar kullanılır. Bir resmin görüntü verileri ve resim hakkındaki genişlik, yükseklik gibi bilgiler bu dosyalara çeşitli şekillerde yazılabilir. Bitmap dosyaları dijital resim saklamak için yaygın olarak kullanılmaktadır. Bir bitmap dosyasında dört farklı bölüm bulunabilir. Bunlar bitmap dosya başlığı, bitmap bilgi başlığı, renk paleti ve görüntü verisi bölümleridir. Renk paleti haricindeki bölümler her bitmap dosyasında bulunmak zorundadır. Resimde renk paletinin bulunup bulunmadığı, bitmap bilgi başlığı bölümünden anlaşılır.

Aşağıdaki tanımlamalarda BYTE değişken tipi işaretli bir baytlık değişken, WORD değişken tipi işaretli iki baytlık değişken, DWORD değişken tipi işaretli dört baytlık değişken, LONG değişken tipi işaretli dört baytlık değişken tanımlamaktadır. Bu tipler C programlama dilinin standart veri tipleri olmadıklarından, eğer ayrı bir yerde tanımlanmamışlarsa, aşağıdaki şekilde tanımlanmaları gerekir:

```
typedef unsigned char    BYTE;
typedef unsigned short   WORD;
typedef unsigned long    DWORD;
typedef long             LONG;
```

Bitmap dosya başlığı bölümünde dosyanın yapısı ile ilgili bilgiler bulunur. Bu bölüm aşağıdaki şekilde tanımlanmıştır (Lindley, 1995):

```
typedef struct _BITMAPFILEHEADER {
    WORD    bfType;
    DWORD    bfSize;
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD    bfOffBits;
} BITMAPFILEHEADER;
```

bfType : Dosya tipi. BM olmalıdır.

bfSize : Dosyanın bayt cinsinden büyüklüğü.

bfReserved1 : Kullanılmayan alan. 0 (sıfır) olmalıdır.

bfReserved2 : Kullanılmayan alan. 0 (sıfır) olmalıdır.

bfOffBits : Resim verisinin ilk baytının kayıt numarası. Dosyadaki ilk kayıt (bayt) numarası sıfır kabul edilmektedir.

Bitmap bilgi başlığı bölümünde resmin boyutları ve renk formatı ile ilgili bilgiler bulunur. Bu bölüm aşağıdaki şekilde tanımlanmıştır (Lindley, 1995):

```
typedef struct _BITMAPINFOHEADER {
    DWORD    biSize;
    LONG     biWidth;
    LONG     biHeight;
    WORD     biPlanes;
    WORD     biBitCount;
    DWORD    biCompression;
    DWORD    biSizeImage;
    LONG     biXPelsPerMeter;
    LONG     biYPelsPerMeter;
    DWORD    biClrUsed;
```

```

        DWORD    biClrImportant;
    } BITMAPINFOHEADER;

```

biSize : Bitmap bilgi başlığı bölümünün bayt cinsinden büyüklüğü. Renk paletinin bulunduğu durumlarda renk paleti, bitmap bilgi başlığı kısmından sonra gelir. Bu durumda **biSize** değişkeni yoluyla renk paletinin dosya içindeki başlangıç konumu bulunabilir. Ayrıca bu değişken versiyon kontrolü yapmaya yarar. Örneğin bilgi başlığı bölümü OS/2 1.x bitmap dosyalarında 12 bayt, Windows 3.1 bitmap dosyalarında 40 bayt yer tutar.

biWidth : Resmin piksel cinsinden genişliği.

biHeight : Resmin piksel cinsinden yüksekliği. Eğer pozitif ise resim aşağıdan yukarı şekildedir ve orijini sol alt köşededir. Eğer negatif ise resim yukarıdan aşağı şekildedir ve orijini sol üst köşededir.

biPlanes : Çıkış aygıtındaki renk düzlemi sayısı. 1 olmalıdır.

biBitCount : Bir pikselin kaç bitten oluştuğunu gösteren sayı. 1, 4, 8, 16 veya 24 olabilir. Bu sayı yoluyla resimde en fazla kaç tane renk kullanılabileceği hesaplanabilir. Bir resimde maksimum $2^{\text{biBitCount}}$ renk kullanılabilir. Maksimum renk sayısı ve bilgi başlığı bölümünün **biClrUsed** değişkeni, renk paletinin durumu hakkında bilgi verir. Bir pikselin 1, 4, 8 bitten oluştuğu durumlarda renk paletinin bulunması zorunludur. Bu durumda her bir piksel, renk paletindeki bir RGBQUAD yapısının indeks numarasını gösterir. Kırmızı, yeşil, mavi renk değerleri bu indeks numarasındaki RGBQUAD yapısında bulunur. Renk paleti yoksa resim verisindeki sayılar doğrudan renk değerlerini gösterir.

biBitCount 1'e eşitse bir bit bir pikseli gösterir. 0'ın (sıfır) temsil ettiği renk değerleri renk paletinin ilk kaydında, 1'in temsil ettiği renk değerleri renk paletinin ikinci kaydında tanımlanmıştır. Dosyanın resim verisi bölümündeki her bir bayta 8 adet piksel yazılır. **biBitCount** 4 ise dört bit bir pikseli oluşturur ve resimde maksimum 16 renk bulunur. Renk paleti de 16 adet RGBQUAD yapısından oluşur. Resim verisindeki her bir bayta 2 adet piksel yazılır. **biBitCount** 8'e eşitse resimde 256 renk kullanılabilir. Resim verisindeki her bir bayt bir pikseli gösterir. Bu piksel de 256 adet RGBQUAD yapısından oluşan renk paletinin kayıtlarından birini gösterir.

biBitCount 16 ise resimde en çok 65536 adet renk kullanılabilir. Renk paleti kısmında üç adet dört baytlık sayı bulunur ancak bu sayılar RGBQUAD yapısında değildir. Bu sayılar kırmızı, yeşil, mavi bileşenleri tanımlamak amacıyla resim verisi bölümündeki 16 bitlik piksellere

uygulanacak maskeleri gösterir. Bir pikselin hangi bitlerinin hangi rengi temsil ettiği bu sayılardan anlaşılır. Resim verisindeki iki bayt bir pikseli oluşturur.

biBitCount 24 ise resimde 2^{24} adet renk kullanılabilir. Resim verisi bölümündeki birer baytlık her üç değer doğrudan mavi, yeşil, kırmızı bileşenleri gösterdiğinden renk paletine ihtiyaç yoktur. Üç bayttan oluşan bir pikselin ilk baytı mavi, ikinci baytı yeşil, üçüncü baytı kırmızı renk bileşenlerini gösterir.

biCompression : Sıkıştırma tipi. Bir pikselin 4 veya 8 bitten oluştuğu resimlerde sıkıştırma yapmak mümkündür fakat bitmap dosyalarında genellikle sıkıştırma kullanılmaz. Sıkıştırılmamış resim dosyasında biCompression değişkeninin değeri 0'dır.

biSizeImage : Resim verisi kısmının bayt cinsinden büyüklüğü. Sıkıştırılmamış resimler için 0 olabilir.

biXPelsPerMeter : Çıkış aygıtının piksel/metre cinsinden yatay çözünürlüğü. Değeri 0 olabilir.

biYPelsPerMeter : Çıkış aygıtının piksel/metre cinsinden dikey çözünürlüğü. Değeri 0 olabilir.

biClrUsed : Resimde fiili olarak kullanılan renk sayısı. Bu sayı 0 olursa resimde $1 \ll biBitCount$ ile tanımlanan maksimum sayıda renk kullanılır. Eğer biClrUsed sıfırdan farklıysa ve biBitCount 16'dan küçükse, biClrUsed değişkeni resmi göstermek için gereken gerçek renk sayısını gösterir. Eğer biClrUsed sıfırdan farklıysa ve biBitCount 16'ya eşitse veya 16'dan büyükse, biClrUsed değişkeni Windows renk paletinin performansını optimize etmek için kullanılan renk paletinin büyüklüğünü gösterir. Uygulamalarda biClrUsed değişkeninin değeri genellikle sıfırdır.

biClrImportant: Resmin gösterimi için önemli kabul edilen renk sayısı. 0 ise bütün renkler önemlidir.

Bitmap dosyasında renk paleti varsa, bitmap bilgi başlığı bölümünün arkasında bulunur. Renk paletinin her bir kaydı aşağıdaki şekilde tanımlanan bir yapıdan oluşur:

```
typedef struct _RGBQUAD {
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserved;
} RGBQUAD;
```


Burada rgbBlue, rgbGreen, rgbRed değişkenleri sırasıyla mavi, yeşil, kırmızı renk bileşenlerine karşılık gelir. rgbReserved 0 olmalıdır. Renk paletindeki RGBQUAD yapıları önem sırasına göre sıralanmışlardır.

Dosyadaki son kısım resim verisi kısmıdır. Burada doğrudan resmi oluşturan renk değerleri veya renk paletindeki renk değerlerini temsil eden sayılar bulunur. Resim, dosyaya satır satır yazılmıştır. Resmin satırlarındaki piksellerin dosyaya yazılma sırası soldan sağdır. Gerçek renk (24 bit) resimlerde bir pikseli oluşturan renk değerlerinin yazılma sırası BGR'dır (mavi, yeşil, kırmızı). Eğer bilgi başlığı kısmındaki biHeight üyesi pozitif ise dosyanın resim verisi kısmındaki ilk satır, resmin en alt satırıdır. biHeight negatif ise dosyanın resim verisi kısmındaki ilk satır, resmin en üst satırıdır (Lindley, 1995).

Bir piksel 1 bitten oluşuyorsa resim verisi bölümündeki her baytta 8 piksel bulunur. Böyle bir resim dosyaya yazılırken önce her satır sekizer piksellik parçalara ayrılır. Her bir parça ikilik sistemde bir sayı olarak kabul edilir ve onluk sistemdeki karşılığı 0 ile 255 arasında olan bu sayı dosyaya yazılır. Resmi yeniden oluşturmak için, dosyadan okunan bayları ikilik sisteme çevirmek ve bitleri resim matrisinin uygun yerlerine yerleştirmek gerekir. Bir piksel dört bitten oluşuyorsa resim verisinin her baytında 2 piksel bulunur. Bu durumda resmi oluşturmak için dosyadan okunan sayının onaltılık sistemde karşılığı bulunur. 16^1 'in çarpanı sol taraftaki pikselin, 16^0 'ın çarpanı sağ taraftaki pikselin renklerinin renk paletindeki yerini gösterir.

Bir pikseli oluşturan bit sayısı kaç olursa olsun, dosyaya yazılan her satırdaki bayt sayısı 4 ile tam olarak bölünmelidir. Eğer bir satırda dördün tam katı olmayan sayıda bayt varsa bu satıra ek sıfırlar ilave edilerek bayt sayısının dördün tam katı olması sağlanır. Resim verisi için bilgisayar belleğinde yer ayırırken ve pikselleri oluşturan değerleri dosyadan okurken bu özellik dikkate alınmalıdır. Örnek olarak her biri 9 piksel genişliğinde ve 2 piksel yüksekliğinde üç farklı resmi göz önüne alalım. Bu resimlerde 1 piksel sırayla 1, 8 ve 24 bit yer kaplasın. Piksel başına 1 bitlik resimde 9 pikselin 8 adeti bir bayta sığdırılır. Resmin en sağında kalan tek bir piksel için ek bir bayt daha gerekir. Son piksel de bu baytın en soldaki bitine yazılır. Bu baytın daha sağ taraftaki bitlerinin hepsi sıfır olmalıdır. Böylece 9 piksellik bir satır iki bayta sığar. Dosyanın satır gerekliliklerini sağlamak için bu sayı dördün tam katı olmalıdır. Bunun için satır sonuna her biri sıfır değerini içeren iki bayt daha eklenir. Böylece dosyanın resim verisi kısmındaki her satır 4 bayt uzunluğunda olur. Resim verisinin dosyada kapladığı toplam yer, her satır için 4 bayt ve toplam 2 satır için, $4 \cdot 2 = 8$ bayttır. Piksel başına

8 bitlik resimde bir satır için gereken 9 baytlık yere dosya formatına uyması için 3 bayt daha ilave edilir ve dosyadaki bir satır 12 bayt yer kaplar. Bu durumda resim verisinin büyüklüğü $12 \cdot 2 = 24$ bayttır. Piksel başına 24 bitlik resimde bir satır için gereken $9 \cdot 3 = 27$ baytlık yere dosya formatına uyması için 1 bayt daha ilave edilir ve dosyadaki bir satır 28 bayt yer kaplar. Bu durumda resim verisinin büyüklüğü $28 \cdot 2 = 56$ bayttır. Bu hesaplamalar önemlidir çünkü bitmap dosyalarında büyük çoğunlukla kullanılan sıkıştırılmamış resim formatı için, dosyadaki resim verisi kısmının büyüklüğünü veren `biSizeImage` elemanının değeri sıfır olabilir. Bu yüzden bellekte resime yer ayırmak için bu değişkenin içerdiği değere güvenmek doğru değildir (Lindley, 1995).



3.GÖRÜNTÜ GELİŞTİRME

Görüntü geliştirme tekniklerinin amacı, bir görüntüyü özel bir uygulama alanı için orijinalinden daha uygun bir hale getirmektir. Görüntü geliştirme, insana yönelik uygulamalar için asıl amaç, bilgisayara yönelik uygulamalar için bir ara basamaktır. Çok koyu veya çok açık görüntülerin parlaklık dağılımının düzenlenmesi, insana yönelik uygulamalara örnek olarak gösterilebilir. Bilgisayara yönelik uygulama örnekleri ise, genellikle segmentasyon aşamasına hazırlık olarak kabul edilen eşikleme, görüntü üzerinden maske geçirme gibi işlemlerdir.

Görüntü düzleminde gerçekleştirilen geliştirme işlemleri, nokta işlemleri ve maske işlemleri olarak iki genel kategoride incelenebilir. Görüntü düzlemi terimi bir görüntüyü oluşturan pikseller kümesini ifade eder. Görüntü düzlemi metotları doğrudan bu pikseller üzerinde işlem yapan metotlardır. Görüntü düzlemindeki görüntü işleme fonksiyonları aşağıdaki şekilde gösterilebilir (Gonzalez ve Woods, 1993):

$$g(x, y) = T[f(x, y)] \quad (3.1)$$

Burada $f(x, y)$ giriş görüntüsü, $g(x, y)$ işlenmiş görüntü, T ise f üzerinde (x, y) 'nin bir komşuluğunda tanımlı bir operatördür. T operatörü bir dizi görüntü üzerinde de tanımlanabilir; böylece iki görüntüyü toplamak veya bir dizi görüntünün ortalamasını almak gibi işlemler yapılabilir.

(x, y) 'nin bir komşuluğunu belirlemek için en çok kullanılan yöntem, merkezi (x, y) olan kare şeklinde bir bölge tanımlamaktır. T 'nin en basit olduğu durum, bu bölgenin boyutlarının 1×1 olduğu durumdur. Burada, T herhangi bir şekilde belirlendikten sonra, g 'nin alacağı değer f 'nin (x, y) 'deki değerine bağlıdır. Bu durumda T 'ye gri ton değeri dönüşüm (veya eşleme) fonksiyonu adı verilir. Görüntünün herhangi bir noktasındaki geliştirme sadece bu noktadaki gri ton değerine bağlı olduğundan, bu kategorideki tekniklere nokta işleme teknikleri adı verilir. (x, y) 'nin bir komşuluğunu belirlemek için daha büyük bölgeler de tanımlanabilir. Pratikte en çok, (x, y) noktasını merkez alan ve kenarları tek sayı olan kare şeklinde bölgeler kullanılır. Bu bölgelere maske (veya filtre, pencere) adı verilir. Sonuçta elde edilen görüntünün bir (x, y) noktasındaki değeri, giriş görüntüsünde merkezi (x, y) olan bölgedeki

pikseller ve bu bölge üzerine uygulanan maskenin katsayıları yardımıyla belirlenir. Bu kategorideki tekniklere maske işleme teknikleri adı verilir (Gonzalez ve Woods, 1993).

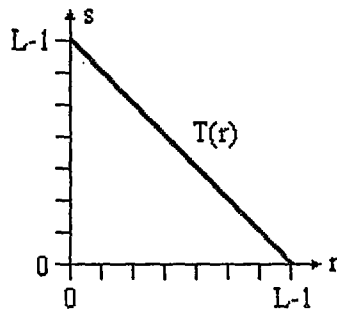
3.1 Nokta İşleme Teknikleri

Nokta işleme tekniklerinde sonuçta elde edilen görüntünün bir (x,y) noktasındaki değeri, orijinal görüntünün aynı (x,y) noktasındaki değeri bağlıdır. Bu kategorideki teknikler, sadece grinin tonlarından oluşan görüntüler için, $s=T(r)$ bağıntısıyla gösterilebilir. Burada r orijinal görüntüdeki bir piksel değeri, s de geliştirme işleminden sonraki piksel değeridir. Görüntüde $[0,L-1]$ arasında toplam L adet parlaklık değeri olduğu kabul edilirse, T fonksiyonu $[0,L-1]$ aralığındaki değerleri yine $[0,L-1]$ aralığına dönüştüren, tek değerli bir fonksiyondur.

Uygulamalarda r 'nin $[0,L-1]$ aralığında alacağı değerler için s değerleri önceden hesaplanır ve bu değerler bir tabloya aktarılır. Görüntü geliştirme işlemi sırasında orijinal görüntü taranır, her bir pikselin gri ton değerine karşılık gelen yeni değer tablodan bulunur ve yeni görüntüde yerine yazılır. Nokta işleme tekniklerinde bir piksel sadece bir defa kullanıldığı için yeni piksel değerini orijinal görüntüdeki yerine yazmak mümkündür.

3.1.1 Negatif Görüntü

Görüntü negatiflerinin kullanım yerlerine örnek olarak tıbbi görüntülerin gösterimi veya fotoğrafçılık alanında saydam hazırlanması konuları gösterilebilir. Bir görüntünün negatifi, Şekil 3.1'de görülen $s=T(r)$ dönüşüm fonksiyonu ile elde edilir (Gonzalez ve Woods, 1993).



Şekil 3.1 Negatif görüntü için dönüşüm fonksiyonu

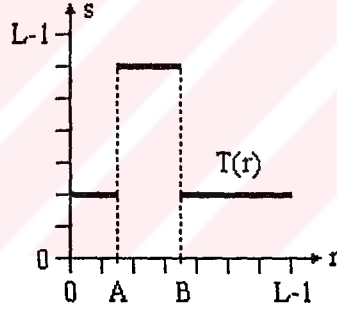
Bu fonksiyon aşağıdaki formülle ifade edilebilir:

$$s = T(r) = L - 1 - r \quad (3.2)$$

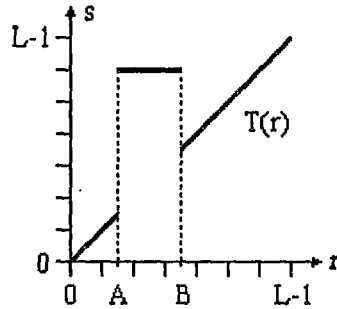
Burada L değeri gri tonların sayısını göstermektedir.

3.1.2 Gri Ton Değerlerini Dilimleme

Bazı uygulamalarda bir görüntüdeki belli bir gri ton değeri aralığını vurgulamak gerekebilir. Bu işlemi gerçekleştirmenin çeşitli yöntemleri vardır ancak bunların çoğu, iki temel yöntemin çeşitlemeleridir. Her iki yöntemde de ilgi alanına giren gri ton değerleri belli bir seviyeye yükseltilir. Birinci yöntemde diğer gri ton değerleri bundan daha düşük bir değere indirilir (Şekil 3.2). İkinci yöntemde ise diğer gri ton değerleri değiştirilmez (Şekil 3.3).



Şekil 3.2 İlgi alanına girmeyen gri tonların düşük bir değere çekildiği fonksiyon



Şekil 3.3 İlgi alanına girmeyen gri tonların değişmediği fonksiyon

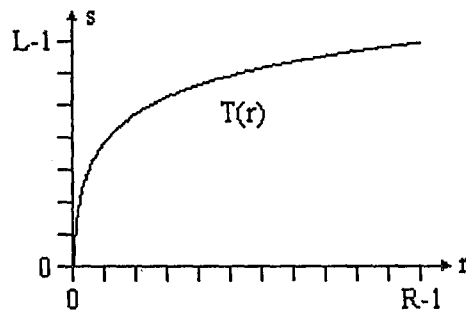
Bu fonksiyonlar sırasıyla aşağıdaki şekilde ifade edilebilir (Gonzalez ve Woods, 1993):

$$s = T(r) = \begin{cases} c_1, & 0 \leq r < A \\ c_2, & A \leq r \leq B \\ c_1, & B < r \leq L-1 \end{cases} \quad (3.3)$$

$$s = T(r) = \begin{cases} r, & 0 \leq r < A \\ c_2, & A \leq r \leq B \\ r, & B < r \leq L-1 \end{cases} \quad (3.4)$$

3.1.3 Gri Ton Ölçeğini Sıkıştırma

Bazı uygulamalarda dijital görüntüdeki gri ton sayısı, görüntünün gösterilmesinde kullanılan bir aygıtın (örneğin ekran kartı) gösterme kapasitesinin üzerinde olabilir. Toplam 256 gri tonundan oluşan bir görüntüyü sadece 16 ton gösterebilen bir aletle ekrana yansıtmak, bu durumun bir örneğidir. Bu durumda görüntüdeki piksellerin parlaklık değeri sınırlarını uygun bir düzeye çekmek gerekir. Bu işlem için basamak fonksiyonu kullanmak mümkündür fakat bu şekilde oluşturulan görüntülerden, ayırt edilebilirlik açısından beklenen sonucun elde edilemediği gözlemlenmiştir. Basamak fonksiyonuna bir seçenek olarak şekildeki fonksiyon önerilebilir.



Şekil 3.4 Gri ton ölçeğini sıkıştırmak için kullanılan fonksiyon örneği

Orijinal görüntüde R adet, yeni görüntüde L adet parlaklık değeri bulunmaktadır. Bu fonksiyon aşağıdaki şekilde ifade edilebilir (Gonzalez ve Woods, 1993):

$$s = T(r) = c \cdot \log(1 + |r|) \quad (3.5)$$

Buradaki c sabiti, formülde r ve s için maksimum değerler yerlerine yazılarak bulunabilir. Yukarıda verilen örnekte $s=15$, $r=255$ için $15=c \log(256)$ ve $c=6.228$ bulunur.

3.1.4 Görüntü Histogramı

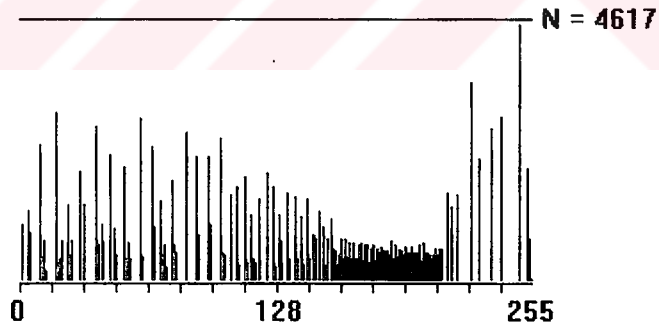
Gri ton değerleri $[0, L-1]$ aralığında değişen bir dijital görüntünün histogramı,

$$p(r_k) = \frac{n_k}{N} \quad (3.6)$$

ayrık fonksiyonudur. Burada r_k k . gri ton değerini, n_k görüntüde bu ton değerinden piksellerin sayısını, N görüntüdeki toplam piksel sayısını gösterir ($k=0, 1, 2, \dots, L-1$) (Gonzalez ve Woods, 1993). Bu şekilde tanımlanmış $p(r_k)$ fonksiyonu, gri ton değerlerinin görüntüde bulunma olasılıklarının hesabını verir. Uygulamalarda histogramın tanımı çeşitli şekillerde basitleştirilebilir. Örneğin r_k değeri hemen her zaman doğrudan k tamsayısı olarak alınır. Ayrıca görüntü histogramı gösteren tablolarda çoğu zaman normalize edilmiş değerler (olasılıklar) yerine her bir gri ton değerinden piksellerin sayısı (frekanslar) gösterilir. Şekil 3.5’de örnek bir görüntü ve Şekil 3.6’da bu görüntünün histogramı görülmektedir.



Şekil 3.5 Okul; örnek bir görüntü.



Şekil 3.6 Okul görüntüsünün histogramı

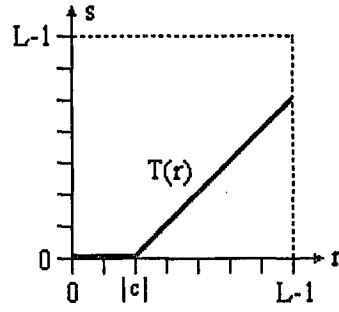
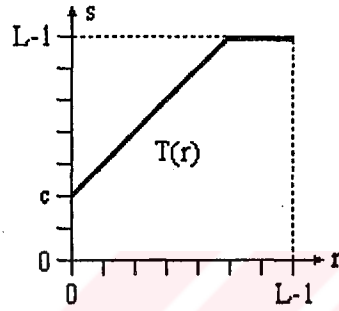
Görüntü histogramı görüntünün genel görünümü hakkında bir bilgi verir. Bu görünüm, histogramda piksellerin yoğunlaştığı bölgelere bakarak çeşitli şekillerde yorumlanabilir. Eğer pikseller histogramın küçük gri ton sayılı değerlerini gösteren bölgelerinde yoğunlaşıyorsa görüntünün genel olarak karanlık olduğu sonucuna varılabilir. Pikseller yüksek gri ton değerlerinde yoğunlaşıyorsa çok aydınlık ve beyazın ağırlıklı olduğu bir görüntü söz konusudur. Pikseller histogramın orta bölgelerinde dar bir alanda yoğunlaşıyorsa görüntüye gri bir ton hakimdir. Bu durumlar, kontrast geliştirme veya histogram işleme tekniklerinin uygulanabilirliği konusunda fikir verir. Ancak, histogramın verdiği bilgilerin sınırlarına dikkat etmek gerekir. Örneğin histogram, görüntünün içeriğine ait herhangi bir bilgi taşımaz. Ayrıca histogramın verdiği bilginin, görüntünün bütününe ait olduğunu unutmamak gerekir. Histogram, her bir gri ton değerinden piksellerin sayısı hakkında bilgi vermesine karşılık, gri ton değerlerinin görüntü düzlemindeki dağılımı konusunda bir bilgi veremez. Bu yüzden görüntünün bütünüyle değil de sadece belli bir bölgesiyle ilgileniyorsak, bu bölgeye ait histogramı hesaplamak gerekecektir (Efford, 2000).

3.1.5 Parlaklık Ayarlama

Gri ton değerlerinin ağırlıklı ortalamasını ve varyansını hesaplayarak görüntünün genel görünümüyle ilgili iki adet özellik elde edebiliriz. Gri ton değerlerinin ağırlıklı ortalaması görüntünün ortalama parlaklığının, varyans da kontrastın bir ölçüsüdür. Görüntünün parlaklığını ayarlamak için görüntünün piksellerinin gri ton değerlerine sabit bir c sayısı eklenebilir. Bu durumda dönüşüm fonksiyonu

$$s = T(r) = r + c \quad (3.7)$$

şeklindedir (Lindley, 1995). Gri ton değerleri $[0, L-1]$ aralığındaki dijital görüntü için $c < 0$ durumunda $s < 0$ bulunuyorsa $s = 0$, $c > 0$ durumunda $s > L-1$ bulunuyorsa $s = L-1$ kabul edilir. Bu iki durum için dönüşüm fonksiyonlarının grafiği aşağıda görülmektedir:

Şekil 3.7 $c < 0$ içi dönüşüm fonksiyonuŞekil 3.8 $c > 0$ için dönüşüm fonksiyonu

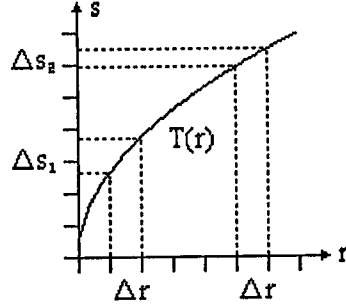
Görüntü histogramı $c < 0$ durumunda şeklini değiştirmeden sol tarafa doğru, $c > 0$ durumunda ise sağ tarafa doğru kayar. $[0, L-1]$ sınırlarının aşıldığı hallerde, 0 veya $L-1$ gri ton değerlerinde birikme olur.

3.1.6 Kontrast Geliştirme

Sözlük anlamı ‘karşıtlık’ olan kontrast sözcüğü, görüntünün ayırt edilebilirliğini ifade etmek için kullanılır. İnsan gözünün gri ton değerlerini ayırt etme kapasitesi ortalama olarak 30 adet gri ton değeriyle sınırlıdır. Bir görüntüdeki en düşük ve en yüksek gri ton değerleri arasındaki farkın bu sınırdan düşük olması durumunda, bu görüntüdeki detayların insan gözü tarafından yeterince algılanmadığı söylenebilir. Bu tipteki görüntülerin histogramlarında, gri ton ölçeğinin dar bir aralığında çok sayıda pikselin yoğunlaştığı görülür. Bu görüntülere düşük kontrastlı görüntüler adı verilir (Lindley, 1995).

İnsanın incelemesine uygun olmayan düşük kontrastlı görüntüler çeşitli dönüşüm fonksiyonları yoluyla insan için daha uygun duruma getirilebilirler. Farklı türdeki düşük kontrastlı görüntüler için farklı fonksiyonlar kullanmak gerekir. Koyu tonların ağırlıklı olduğu

bir görüntüyü geliřtirmek için gri ton deęerlerini açık tonları gösteren yüksek deęerlere doęru kaydirmek gerekir. řekil 3.9'da görölen türden bir fonksiyon, istenen özellięe sahiptir.



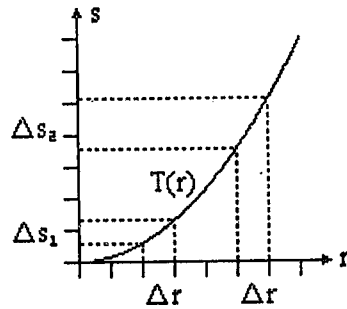
řekil 3.9 Karanlık görüntüler için dönüşüm fonksiyonu

Bu fonksiyonda $\Delta s_1 > \Delta r$ ve $\Delta s_2 < \Delta r$ eşitsizlikleri sağlanmaktadır. Bu yüzden orijinal görüntüde düşük gri tonları gösteren deęerler, işlenmiş görüntüde daha geniş bir aralıęa yayılır. Bu özellięe sahip çeřitli fonksiyonlar bulunabilir; örneęin

$$s = T(r) = \sqrt{L-1}\sqrt{r} \quad (3.8)$$

Bu işlemin yan etkisi, yüksek gri ton deęerlerinin orijinal görüntüden daha dar bir alana sıkışması sonucunda açık tonlarla gösterilen detaylarda kayıplar oluşmasıdır (Efford, 2000).

Açık tonların aęırlıklı olduęu bir görüntü de benzer řekilde geliřtirilebilir. Ancak burada, yukarıda tartışılan durumun tersine, alçak deęerleri sıkıştırıp yüksek deęerleri daha geniş bir aralıęa yayacak bir fonksiyon gereklidir. řekil 3.10'da bu türden bir fonksiyon görölmektedir.

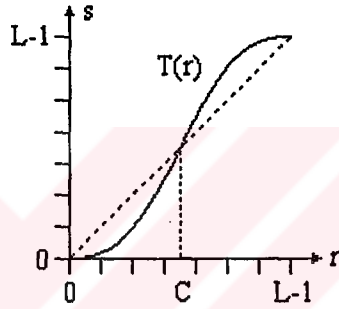


řekil 3.10 Aydınlık görüntüler için dönüşüm fonksiyonu

Bu fonksiyonda $\Delta s_1 < \Delta r$ ve $\Delta s_2 > \Delta r$ eşitsizlikleri sağlanmaktadır. Sıfır değerini tekrar sıfıra, $L-1$ değerini tekrar $L-1$ 'e eşlemesine dikkat etmek koşuluyla herhangi bir üstel fonksiyon bu işi görecektir; örneğin

$$s = T(r) = r^2 (L-1)^{-1} \quad (3.9)$$

Piksel değerlerinin, gri ton ölçeğinin orta kısımlarında yoğunlaştığı görüntülerde kontrastı artırmak için açık tonları daha çok açmak, koyu tonları daha çok koyulaştırmak gerekir. Bu sonucu elde etmek için ihtiyaç duyulan fonksiyon, Şekil 3.11'de grafiği verilen türde bir fonksiyondur (Efford, 2000).



Şekil 3.11 Gri görüntüler için dönüşüm fonksiyonu

Sabit bir C sayısından küçük olan değerler koyu ton bölgelerine, C sayısından büyük olan değerler açık ton bölgelerine doğru itilmektedir. Aşağıda verilen fonksiyon, bu özellikleri sağlamaktadır.

$$s = T(r) = r - \sin[2\pi r (L-1)^{-1}] \quad (3.10)$$

Bu fonksiyonda C değeri $(L-1)/2$ 'ye eşittir, yani sabit nokta gri ton ölçeğinin ortasında yer alır. K değeri fonksiyonun $s=r$ refereans doğrusundan ne kadar saptığını göstermektedir. Pozitif K değerleri için kontrast artmakta, negatif K değerleri için kontrast azalmaktadır. $|K|$ 'nın belli bir değerden büyük olması durumunda $T(r) < 0$ veya $T(r) > L-1$ sonuçları elde edileceğinden K 'nın değerini uygun bir aralıkta sınırlamak gerekir (Lindley, 1995).

3.1.7 Histogram Eşitleme

Kontrast geliştirme bölümünde tartışılan tekniklerin etkinliği, gri ton değerlerinin histogramda yoğunlaştığı bölgelere bağlıdır. Kontrastı artırmak için bu yoğunlaşma bölgelerinden bağımsız yöntemler de mevcuttur. Bunlardan biri kontrast yaymadır. Bu yöntem $f(x,y)$ görüntüsündeki piksellerin gri ton değerlerinin

$$a \leq f(x,y) \leq b \quad (3.11)$$

şeklinde sınırlı olduğu görüntülerde uygulanabilir. Bu koşullarda

$$s = T(r) = (L-1)(r-a)(b-a)^{-1} \quad (3.12)$$

fonksiyonu, $[a,b]$ aralığındaki gri ton değerlerini $[0,L-1]$ aralığına eşit aralıklarla yaymaktadır. Burada L değeri toplam gri ton sayısını göstermektedir. Sonuçta gözle görülür bir etki elde etmek için $b-a$ değerinin L 'ye göre küçük olması (örneğin yaklaşık olarak $L/6$) gerekir.

Histogram eşitleme yöntemi de, kontrast yayma yöntemi gibi, gri ton değerlerini tekrar dağıtmayı hedefleyen bir yöntemdir. Histogram eşitleme yönteminde piksel sayısının fazla olduğu gri ton değerlerine daha geniş bir aralık, piksel sayısının az olduğu gri ton değerlerine daha dar bir aralık ayrılır. Bu yöntemde kullanılan dönüşüm fonksiyonu aşağıdaki biçimde tanımlanmıştır (Gonzalez ve Woods, 1993):

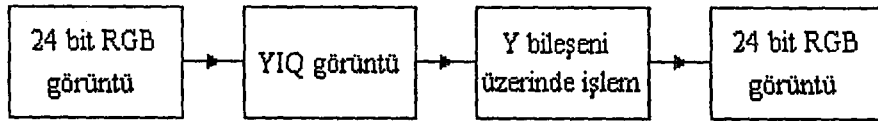
$$s = T(r) = \frac{(L-1)}{N} \sum_{k=0}^r n_k \quad (3.13)$$

Burada L toplam gri ton sayısını, N görüntüdeki toplam piksel sayısını, n_k k . gri ton değerinden piksel sayısını göstermektedir. Bu fonksiyon sonucunda piksel sayısı fazla olan gri ton değerleri için kontrast artar, az olanlar için kontrast azalır. Histogram eşitleme yöntemi histogram yayma yönteminden genellikle daha üstün sonuçlar verir. Histogram eşitleme yöntemi, gri ton değerleri bütün ölçeğe yayılmış görüntüler için kullanılabilir. Ancak bu yöntemin, daha çok sayıda bulunan gri ton değerinin daha önemli olduğunu kabul eden istatistiksel bir yöntem olduğuna dikkat etmek gerekir. İlgilendiğimiz görüntüde böyle bir durum yoksa yöntem sonucunda elde edilen görüntü kullanılmaz olacaktır. Ayrıca histogram

eşitleme yönteminin, histogramdaki gri ton değerlerinin dar bir aralığa sıkıştığı görüntülerde her zaman istenen sonucu vermediği gözlenmiştir.

3.1.8 Renkli Görüntülerin Histogramı

Gerçek renk (true color) olarak isimlendirilen görüntülerde her bir piksel üç ayrı bileşenden oluşur. Bu bileşenler, pikselin rengini oluşturan kırmızı, yeşil ve mavi temel renklerin ağırlıklarını gösterir. Bu görüntüde her bir bileşenin ayrı bir histogramı vardır. Ancak RGB renk modelindeki görüntü ve bu görüntünün histogramı, görüntü geliştirme tekniklerini uygulamaya elverişli değildir. Örneğin histogram eşitleme yöntemi her bir histograma ayrı ayrı uygulanırsa sonuçta elde edilecek görüntüdeki renkler, orijinal görüntü ile tutarlı olmayacaktır. Bu yüzden renkli görüntüyü görüntü işleme tekniklerine uygun bir modele dönüştürmek gerekir. Örneğin YIQ renk modelinde parlaklık (ışık yoğunluğu) bilgisi renk bilgilerinden ayrı bir bileşende saklandığı için bu model, önceki alt bölümlerde tartışılan görüntü işleme tekniklerinin uygulanmasında elverişlidir. Bunun için önce görüntü RGB'den YIQ renk modeline dönüştürülür ve parlaklığı gösteren Y bileşeni üzerinde görüntü geliştirme teknikleri uygulanır. Son olarak görüntü, YIQ renk modelinden RGB'ye ters dönüşüm ile ekranda gösterilmeye hazır hale gelir. Bu işlemler Şekil 3.12 'de şematik olarak gösterilmiştir.

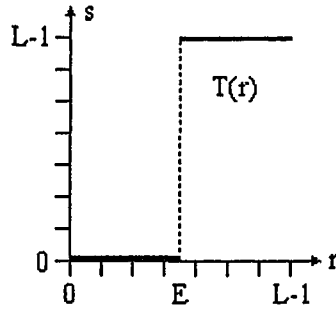


Şekil 3.12 Renkli görüntüler için geliştirme işlemi şeması

3.1.9 Eşikleme

Görüntü geliştirme söz konusu olduğunda eşikleme işleminin amacı, gri ton ölçeğinde L adet değer bulunan bir görüntüyü ele alıp, bu görüntüden gri ton ölçeği sadece iki adet değer içeren bir görüntü elde etmektir. Bir çok uygulamada $[0, L-1]$ aralığındaki L adet değerden $\{0; L-1\}$ veya $\{0; 1\}$ değerlerine geçilir. Hangi değerlerin seçileceği, uygulamanın ihtiyaçlarına bağlıdır. Eşikleme yöntemi global olarak uygulanırken gri ton ölçeği üzerinde bir eşik değeri saptanır ve bu eşik değerinden küçük olan değerler yerine sıfır, büyük olanlar yerine $L-1$

(veya uygulamaya göre 1) yerleştirilir. Bu değerlerin çeşitli anlamları olabilir. Örneğin bir uygulamada $\{0; 1\}$ değerleri, siyah için 0 ve beyaz için 1 olmak üzere, doğrudan renk değerlerini temsil ederken bir başka uygulamada 0 değeri arka planı ve 1 değeri ön planı temsil edebilir. Bu açıdan bakıldığında eşikleme işleminin amacı, görüntüde ilgi alanına giren kısımları diğer kısımlardan ayırmaktır. Aşağıdaki şekilde $[0, L-1]$ aralığındaki değerleri $\{0; L-1\}$ değerlerine gönderen bir dönüşüm fonksiyonu görülmektedir (Efford, 2000).



Şekil 3.13 Eşikleme için dönüşüm fonksiyonu

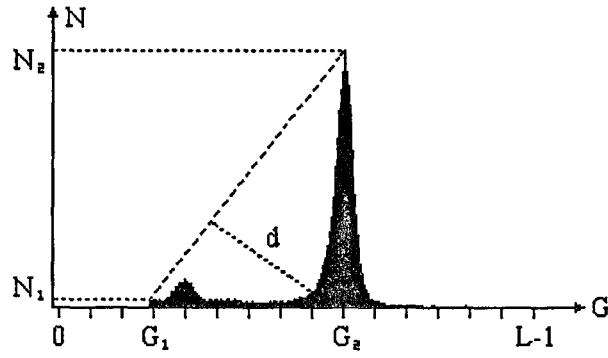
Bu fonksiyon aşağıdaki şekilde ifade edilebilir:

$$s = T(r) = \begin{cases} 0; & r \leq E \\ L-1; & r > E \end{cases} \quad (3.14)$$

Eşikleme işleminde kullanılacak olan E değerini saptamanın çeşitli yolları vardır. Bir gözlemci görüntü histogramını kontrol ederek bu değeri belirleyebilir. Ayrıca histogramın ağırlıklı olarak iki ayrı kümeden oluştuğu durumlar için kullanılacak yöntemler de mevcuttur. En sık kullanılan yöntemde, başlangıç olarak keyfi bir E_0 eşik değeri seçilir. Sonra gri ton değeri bu değerden küçük olan piksellerin gri tonlarının ortalaması (M_1) ve büyük olan piksellerin gri tonlarının ortalaması (M_2) bulunur. Yeni eşik değeri, M_1 ve M_2 değerlerinin ortalamasıdır:

$$E_1 = (M_1 + M_2) / 2 \quad (3.15)$$

Bu işleme $E_k = E_{k-1}$ eşitliği sağlanıncaya kadar devam edilir. Diğer bir yöntemde, histogramda en çok sayıda pikseli gösteren noktadan diğer kümenin bulunduğu tarafta en uçtaki gri ton değerini ve bu değerden kaç tane piksel bulunduğunu gösteren noktaya bir doğru çizilir ve bu aralıktaki histogram değerlerinin bu doğruya uzaklığı hesaplanır.



Şekil 3.14 Örnek bir histogram

Şekil 3.14'da örnek bir histogram görülmektedir. Burada G eksenini gri ton değerlerini, N eksenini piksel sayılarını göstermektedir. Bu histogramın ait olduğu görüntüde, yüksek parlaklık değerine sahip bir fon üzerinde koyu tonlarla gösterilen nesneler olduğu düşünülmektedir. Histogramda G_2 gri ton değerinde N_2 adet piksel bulunmaktadır ve bu sayı maksimum değerdir. G_1 gri ton değeri diğer kümenin en uç elemanını göstermektedir ve bu değerden N_1 adet piksel bulunmaktadır. Bu durumda (G_1, N_1) ve (G_2, N_2) noktalarından geçen doğrunun denklemi

$$(N_2 - N_1)G + (G_1 - G_2)N + G_2N_1 - G_1N_2 = 0 \quad (3.16)$$

ile verilir. Histogram değerlerinin, $k=0, 1, \dots, L-1$ olmak üzere h_k ile gösterildiğini kabul edersek, G_1 ve G_2 gri ton değerleri arasında kalan noktaların bu doğruya uzaklıkları

$$d_i = \frac{|(N_2 - N_1)i + (G_1 - G_2)h_i + G_2N_1 - G_1N_2|}{\sqrt{(N_1 - N_2)^2 + (G_1 - G_2)^2}} \quad (3.17)$$

ile hesaplanır. Burada i değişkeni G_1 ve G_2 arasında değişmektedir. Sonuçta en büyük d_i değerini veren gri ton değeri, eşikleme noktası olarak belirlenir.

3.2 Maske İşleme Teknikleri

Maske işleme tekniklerinde sonuçta elde edilen görüntünün bir (x,y) noktasındaki değeri, orijinal görüntüde (x,y) 'nin bir komşuluğundaki değerlere bağlıdır. Bu komşuluk genellikle

(x,y) noktasını merkez alan 3×3, 5×5, ... büyüklüğünde kare şeklinde bölgelerden (maske) oluşur. Şekilde 3×3 büyüklüğünde bir maske görülmektedir.

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

Şekil 3.15 Maske modeli

Maskenin katsayıları $w_0, \dots, w_8$ ile gösterilmiştir. Uygulama sırasında maske, w_4 katsayısı (x,y) noktasındaki piksele karşılık gelecek şekilde orijinal görüntünün üzerine yerleştirilir. Orijinal görüntüde maskenin altında kalan piksellerin gri ton değerleri $z_0, \dots, z_8$ ile gösterilirse bu maskeden elde edilecek değer

$$S = \sum_{i=0}^8 w_i z_i \quad (3.18)$$

ile hesaplanır. Elde edilen bu değer yeni görüntünün (x,y) noktasına yazılır. Benzer şekilde bütün görüntü taranır. Görüntünün kenarlarındaki piksellerde, maskenin sadece sınırların dışına taşmayan katsayılarıyla hesap yapılır veya kenar pikselleri hesaplanmadan, oldukları gibi yeni görüntüye geçirilirler. Maske işleme tekniklerinde, görüntünün herhangi bir pikseli hesaplama sırasında bir defadan fazla kullanıldığı için, yeni değeri orijinal görüntüde yerine yazmak mümkün değildir (Efford, 2000; Gonzalez ve Woods, 1993).

3.2.1 Görüntü Aritmetiği

Dijital görüntüler üzerinde aritmetiksel işlemler tanımlamak olanaklıdır. Bu işlemlerin hesaplamalarında görüntü piksellerinin gri ton değerleri kullanılır. Aynı boyutlarda iki görüntü üzerinde aritmetiksel bir işlem yapmak için, iki görüntünün de birbirine karşılık gelen bütün (x,y) noktalarındaki piksellerinin gri ton değerleri operand olarak kullanılır ve elde edilen değer yeni görüntünün (x,y) noktasına yazılır. Boyutları aynı olmayan görüntülerde aritmetiksel işlemler sadece iki görüntünün üst üste gelen kısımları için tanımlıdır.

Gri ton değerleri $[0, L-1]$ aralığında değişen iki görüntü toplanırsa yeni görüntüdeki gri ton değerleri $[0, 2(L-1)]$ aralığında olacaktır. Bu değerleri $[0, L-1]$ aralığına çekmenin bir yolunu

bulmak gerekir. Bunun için toplama işlemi sonucunda elde edilecek görüntü aşağıdaki şekilde tanımlanır (Efford, 2000):

$$g(x, y) = af_1(x, y) + (1 - a)f_2(x, y) \quad (3.19)$$

Burada f_1 ve f_2 giriş görüntüleri, a sabit bir sayı ($0 < a < 1$), ve g de sonuçta elde edilen görüntüdür. a 'nın çeşitli değerleri için görüntülerden herhangi birinin daha baskın olması sağlanabilir. $a=0.5$ değeri için iki görüntünün aritmetiksel ortalaması hesaplanmış olur ve yapılan işlem görüntü ortalama adını alır. Görüntü ortalamasının kullanım alanı, lekelerden temizlenmiş bir görüntü elde etmektir. Bu işlem, değişmeyen bir sahnenin arka arkaya dijital görüntülerinin alınabildiği durumlarda uygulanabilir. Bu yöntemle, görüntü alma aygıtının görüntüye eklediği lekeler ortadan kaldırılabilir. Toplam K adet görüntü girişi için sonuç görüntüsü

$$g(x, y) = \frac{1}{K} \sum_{i=1}^K f_i(x, y) \quad (3.20)$$

ile tanımlanır (Gonzalez ve Woods, 1993).

Bir görüntüden bir başka görüntünün çıkarıldığı durumda da elde edilen değerlerin, istenen sınırların dışında olması durumuyla karşılaşılır. Bu sefer, gri ton değerleri $[0, L-1]$ aralığında değişen iki görüntüden gri ton değerleri $[-(L-1), L-1]$ aralığında değişen yeni bir görüntü elde edilir. Burada negatif değerler yerine sıfır değeri yazılır veya yeni görüntü

$$g(x, y) = |f_1(x, y) - f_2(x, y)| \quad (3.21)$$

ile tanımlanır. Görüntü çıkarma işlemi de genellikle aynı sahnenin arka arkaya alınmış görüntüleri üzerinde uygulanır. Burada amaç, değişmeyen arka planı ortadan kaldırmaktır. Örneğin bu sayede, birbirini takip eden iki görüntüdeki hareketli nesneleri belirlemek mümkündür (Efford, 2000).

3.2.2 Lineer Filtreleme

Dijital görüntüler söz konusu olduğunda görüntü frekansı terimi, görüntü düzleminde piksellerin gri ton değerlerinin değişimini ifade eder. Frekans, görüntünün bir pikselinden diğerine geçildiğinde parlaklık değerindeki değişimin ölçüsüdür. Gri ton değerlerinin küçük farklarla değiştiği bir görüntüde alçak frekanslar ağırlıktadır. Ton farkı büyük bölgeler veya ince detaylar içeren görüntülerde yüksek frekanslı bileşenler bulunur. Çeşitli maskeler yardımıyla bir görüntüdeki yüksek veya alçak frekansları öne çıkarmak veya bastırmak mümkündür.

Yüksek frekansları geçiren filtreler görüntünün alçak frekanslı bileşenlerini zayıflatır veya ortadan kaldırırken yüksek frekanslı bölgeleri öne çıkarır. Yüksek frekanslı bileşenler genellikle bölge kenarlarını veya diğer keskin detayları işaret ettiğinden, yüksekgeçiren filtre görüntüyü keskinleştirir. Görüntüdeki lekeler veya gürültüler (görüntünün parçası olmayan benekler veya noktalar) genellikle çevrelerinden büyük oranda farklı gri ton değerlerine sahip olduklarından, yüksekgeçiren filtre sonucunda daha da belirgin hale gelirler. Alçak frekansları geçiren filtreler ise, görüntünün yüksek frekanslı bileşenlerini zayıflatırken alçak frekanslı bileşenlerinin değişmeden kalmasını sağlar. Alçakgeçiren filtrenin etkisi ile görüntüdeki lekeler daha az belirgin hale gelir ancak, yüksek frekans içeriği de zayıfladığı için görüntüdeki ince detaylar silikleşir ve görüntü bulanıklaşır (Efford, 2000).

Alçakgeçiren bir filtrenin bütün katsayıları pozitifdir. Aşağıda Şekil 3.16'da iki adet örnek filtre görülmektedir.

$$\frac{1}{9} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{10} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Şekil 3.16 Alçakgeçiren filtre örnekleri

Sol taraftaki filtre, ortalama filtre olarak adlandırılır. Bu filtre görüntü üzerinde piksellerin gri ton değerlerinin değişmediği bir bölgeye yerleştirildiğinde merkezdeki gri ton değeri değişmeden kalır. Aynı filtre, gri ton değerlerinde büyük değişiklikler olan bir bölgeye yerleştirildiğinde, merkez pikseline çevresindeki piksellerin gri ton değerlerinin ortalaması atanacağından merkez ile çevre pikseller arasındaki fark azalacaktır. Bu yüzden bu tür filtreler

görüntüdeki gürültüleri ortadan kaldırmak için kullanılırlar ancak yan etki olarak görüntüyü bulanıklaştırırlar (Gonzalez ve Woods, 1993).

Yüksekgeçiren bir filtre, merkezde büyük, pozitif bir katsayı ve bunun etrafında negatif katsayılar ile belirlenir. Aşağıda Şekil 3.17'de iki adet örnek filtre görülmektedir.

$$\frac{1}{9} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Şekil 3.17 Yüksekgeçiren filtre örnekleri

Sol taraftaki filtre, temel yüksekgeçiren filtre olarak adlandırılır (Gonzalez ve Woods, 1993). Bu filtrenin katsayılarının toplamı sıfırdır. Bu yüzden filtre, görüntü üzerinde piksellerin gri ton değerlerinin değişmediği bir bölgeye yerleştirildiğinde sıfır sonucunu verir. Aynı bölgeye sağ taraftaki filtre yerleştirildiğinde, gri ton değeri değişmeden kalır. Her iki filtre de gri ton değerlerinde büyük değişiklikler olan bir bölgeye yerleştirildiklerinde, ortadaki büyük katsayı yüzünden değişikliğin olduğu kesim daha fazla ön plana çıkar. Filtrelerin negatif katsayılarından dolayı görüntünün bazı bölgelerinde negatif sonuçlar elde edilebilir. Bu durumda negatif sonuç yerine sıfır değeri veya sonucun mutlak değeri alınabilir.

Orijinal görüntüden, bu görüntünün alçakgeçiren bir filtreyle işlenmiş hali çıkarılırsa sonuç görüntüsünde yüksek frekanslı bileşenler göreceli olarak daha fazla görülür hale gelir. Buna karşılık görüntüdeki gürültüler fazla ortaya çıkmaz. Bu işlem

$$Highpass = Original - Lowpass \quad (3.22)$$

şeklinde yazılabilir. Bu işlemde orijinal görüntü bir A sayısı ile çarpılırsa, yüksek-frekans-vurgulu veya high-boost filtrenin tanımı elde edilir:

$$Highboost = (A)(Original) - Lowpass \quad (3.23)$$

Burada eşitliğin sağ tarafına orijinal görüntü eklenip çıkarılırsa ve (3.22) bağıntısı göz önüne alınırsa

$$Highboost = (A - 1)(Original) + Highpass \quad (3.24)$$

elde edilir. $A=1$ değeri, standart yüksekgeçiren filtreyi verir. $A>1$ için, orijinal görüntü belli bir oranda yüksekgeçiren filtre ile işlenmiş haline eklenir ve bu durum, yüksekgeçiren filtre sonucunda kaybolmuş alçak frekanslı bileşenleri tekrar görüntüye kazandırır. Sonuç olarak elde edilen görüntü, orijinal görüntüye benzeyen ancak görüntüdeki çeşitli bölgelerin kenarlarının daha çok ortada olduğu yeni bir görüntüdür. High-boost filtre bir maske şeklinde tanımlanabilir:

$$\frac{1}{9} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Şekil 3.18 High-boost filtre için maske

Burada w katsayısı

$$w = 9A - 1 \quad (3.25)$$

ile hesaplanır ($A \geq 1$). A değeri 1'e yaklaştıkça görüntünün keskinliği artar ve $A=1$ için standart yüksekgeçiren filtre elde edilir (Gonzalez ve Woods, 1993).

3.2.3 Sıralamaya Dayanan Filtreler

Bir filtre tanımlamanın tek yolu, bunu bir maske şeklinde ifade etmek değildir. Sıralama işlemini temel alan filtrelerde bir pikselin yeni gri ton değerini belirlemek için bu pikselin bir komşuluğu kullanılır ancak, bu bölge üzerine bir maske yerleştirilmez. Bunun yerine, önce bu komşuluktaki piksellerin gri ton değerleri küçükten büyüğe sıralanır ve sonra, bu listenin içinden bir değer seçilerek yeni gri ton değeri belirlenir.

Bu yöntemle ilgili verilebilecek en basit örnekler, minimum ve maksimum filtrelerdir. Minimum filtre için, sıralanmış listenin ilk (en küçük) elemanı çıktı değeri olarak alınır. Maksimum filtre için, sıralanmış listenin son (en büyük) elemanı çıktı değeri olarak alınır. Minimum filtre, görüntünün göreceli aydınlık kısımlarını ortadan kaldırıp karanlık kısımların

boyut olarak büyümesini sağlarken, maksimum filtre bunun tersi bir etkiye sahiptir. Ayrıca çıktı değeri olarak, listenin maksimum ve minimum değerlerinin arasındaki fark alınabilir. Bu şekilde tanımlanan bir filtre, non-lineer kenar belirleyici etkisi gösterir. Ancak, özellikle büyük komşuluklar söz konusu ise, böyle bir filtre ile belirlenen kenarlar görüntüde doğru yerlere yerleşmeyebilirler (Efford, 2000).

Sıralama işlemi yardımıyla, lineer filtre olarak maske şeklinde tanımlanmış ortalama filtre kavramını geliştirmek olanaklıdır. Sıralanmış listenin başından ve sonundan elemanlar atarak, kalanların ortalamasının hesaplanmasıyla oluşan filtreye kesilmiş (trimmed) ortalama filtre adı verilir. En basit durumda, 3×3 büyüklüğündeki bir bölgenin minimum ve maksimum elemanları dışarıda bırakılır ve geriye kalan yedi adet elemanın ortalaması alınır. Genel olarak, $n \times n$ büyüklüğünde bir bölgedeki piksellerin sıralanmış gri ton değerleri

$$z_1 \leq z_2 \leq \dots \leq z_{n^2} \quad (3.26)$$

ile gösterilirse, listenin her iki ucundan a adet eleman atılmasıyla tanımlanan filtreden elde edilecek sonuç

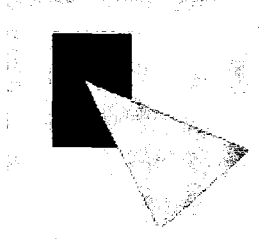
$$S = \frac{1}{n^2 - 2a} \sum_{i=a+1}^{n^2-a} z_i \quad (3.27)$$

ile tanımlanır. Bu filtre, gürültü temizleme işleminde klasik ortalama filtreden daha etkilidir.

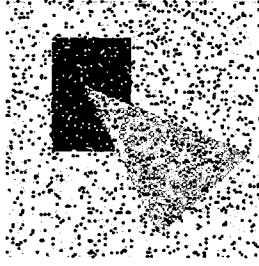
En sık kullanılan sıralama temelli filtre medyan filtredir. Bu filtre için, bir pikselin komşuluğundaki piksellerin gri ton değerlerinin küçükten büyüğe sıralanmasıyla oluşan listenin ortasındaki elemanı, yani medyanı çıktı değeri olarak alınır. 3×3 büyüklüğünde bir komşuluk için medyan değeri listenin beşinci elemanıdır. n tek sayı olmak üzere $n \times n$ büyüklüğünde bir bölge için medyan değerinin yeri $\lfloor n^2/2 \rfloor + 1$ olur. Medyan filtre çeşitli türden gürültüleri ortadan kaldırmak konusunda iyi bir performans gösterir. Gri ton değerleri $[0, L-1]$ aralığında değişen bir görüntüde, gürültüyü oluşturan benekler veya noktalar sıfır veya $L-1$ değerlerini alıyorsa, medyan filtre, görüntünün netliğini kaybettirmeden ve kenarları silikleştirmeden bu gürültüyü ortadan kaldırma konusunda oldukça başarılıdır. Medyan filtrenin işe yarar bir sonuç vermesi için, kullanılan komşuluk içindeki gürültülü piksel sayısı, komşuluktaki toplam piksel sayısının yarısından az olmalıdır (Efford, 2000).

Şekil 3.19’da gürültü filtrelemek için oluşturulmuş örnek bir görüntü görülmektedir. Şekil 3.20’deki görüntü, test görüntüsüne rastgele siyah-beyaz benekler saçılmasıyla oluşturulmuştur. Şekil 3.21, Şekil 3.22 ve Şekil 3.23’de bu gürültülerin çeşitli filtrelerle işlenmiş halleri görülmektedir. Örneklerde de görüldüğü gibi bu tür gürültülerde en iyi sonucu medyan filtre vermektedir.

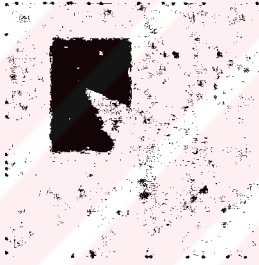




Şekil 3.19 Şekiller; orijinal test görüntüsü



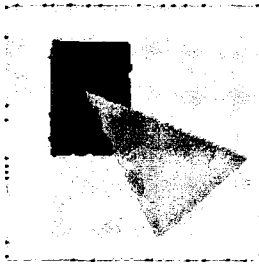
Şekil 3.20 Görüntünün (Şekil 3.19) siyah-beyaz beneklerle bozulmuş hali



Şekil 3.21 Ortalama filtre uygulamasının sonucu



Şekil 3.22 Kesilmiş ortalama filtre uygulamasının sonucu



Şekil 3.23 Medyan filtre uygulamasının sonucu

3.2.4 Kenar Belirleme

Bir görüntünün içeriğini oluşturan nesneler, bu görüntüde arka plandan farklı gri ton değerleriyle belirlenirler. Bu farklılık arka planın sabit bir gri ton değerini taşıdığı durumlarda, bu gri ton değerinde bir süreksizlik şeklinde belli olur. Bir görüntüde bulunabilecek üç temel süreksizlik tipi noktalar, çizgiler ve kenarlardır. Pratikte, görüntüde bu tür süreksizlikleri belirlemede en çok kullanılan yöntem, görüntü üzerinden bir maske geçirmektir. Süreksizlikleri ortaya çıkaran maskeler, çizgiler veya kenarlar çevrelerindeki piksellerden farklı gri ton değerlerine sahip olduklarından, yüksekgeçiren filtre etkisi gösterirler. Yüksekgeçiren filtreler görüntüdeki gürültüleri, özellikle siyah beyaz noktalardan oluşanları, daha da belirginleştirirler. Bu yüzden, benekli-lekeli görüntüler üzerinde çizgi veya kenar arama işlemleri yapmadan önce, bu gürültüleri bir maske ile temizlemek gerekebilir. Ancak bu tür filtrelerin görüntüyü bulanıklaştırdığı ve dolayısıyla kenarları silikleştirdiği unutulmamalıdır. Gürültü temizleme ve süreksizlik belirleme işlemlerinden sonra bu süreksizlikleri vurgulamak amacıyla eşikleme işlemi uygulanabilir.

İzole noktaların belirlenmesinde Şekil 3.17'de sol tarafta görülen temel yüksekgeçiren filtre (katsayı ile çarpmadan) kullanılabilir. Bu durumda maskenin merkezindeki pikselin izole bir nokta olarak tanımlanması için, bu maske sonucunda elde edilen S değerinin

$$|S| > E \quad (3.28)$$

olacak şekilde bir eşik değerinden büyük olması gerekir. Bu maskenin kullanımındaki ana fikir, izole bir noktanın gri ton değerinin, çevresindeki piksellerinkinden büyük oranda farklı olmasıdır (Gonzalez ve Woods, 1993).

Çizgileri belirlemek için aşağıda Şekil 3.24'de görülen maskeler kullanılabilir.

-1	-1	-1
2	2	2
-1	-1	-1
Yatay		
-1	-1	2
-1	2	-1
2	-1	-1
+45°		
-1	2	-1
-1	2	-1
-1	2	-1
Dikey		
2	-1	-1
-1	2	-1
-1	-1	2
-45°		

Şekil 3.24 Çizgi belirleme maskeleri

Örneğin sol taraftaki maske görüntü üzerinden geçirildiğinde, bir piksel kalınlığındaki yatay çizgilere daha çok tepki gösterir. Şekilde görülen maskelerle dört farklı yöndeki çizgiler öne çıkarılabilir. Maskelerin yön konusunda daha hassas olmalarını sağlamak için maskelerin boyutlarını büyütmek gerekir. Görüntüdeki bir noktanın ne tür bir doğru üzerinde olduğunu belirlemek için, bu nokta üzerinde yukarıdaki maskeler hesaplanır. Nokta, en büyük sonuç değerini veren maskenin tanımladığı doğruya aittir. Bütün maskeler birbirlerine yakın sonuçlar veriyorsa bu nokta herhangi bir doğrunun parçası değildir (Gonzalez ve Woods, 1993).

Kenar, aralarında göreceli olarak büyük gri ton değeri farkı bulunan iki bölgeyi birbirinden ayıran sınırdır. Bu tanım bir kenarın, piksellerin parlaklığında lokal bir değişim olduğunu gösterir. Bu yüzden bir çok kenar belirleme yöntemi, lokal türevlerin hesaplanmasına dayanır. Görüntü gradienti, lokal parlaklık değişimleri hakkında yararlı bilgiler verir. Gradientin magnitudü kenar detektörü olarak kullanılabilir.

Bir $f(x,y)$ görüntüsünün (x,y) noktasında gradienti ve bunun magnitudü

$$\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T = [G_x \quad G_y]^T \quad (3.29)$$

$$|\nabla f| = [G_x^2 + G_y^2]^{1/2} \quad (3.30)$$

ile tanımlanır (Gonzalez ve Woods, 1993; Pitas, 1993).

G_x ve G_y 'yi (x,y) noktasında yaklaşık olarak hesaplamamanın çeşitli yolları vardır. En basit yöntem, x ve y yönlerinde

$$G_x \cong f(x+1,y) - f(x,y) \quad (3.31)$$

$$G_y \cong f(x,y+1) - f(x,y) \quad (3.32)$$

farklarını hesaplamaktır. Burada G_x ve G_y 'nin hesabında $f(x,y)$ değeri yerine sırayla $f(x-1,y)$ ve $f(x,y-1)$ de yazılabilir ve böylece komşuluk büyür. Ancak bu formüller gürültüye karşı

hassastırlar ve beklenmeyen yerlerde kenar oluşturlar. Pratikte G_x ve G_y 'nin yaklaşık değerleri maskeler yardımıyla hesaplanır. Bu maskelere gradient operatörleri adı verilir. En çok kullanılan operatörler Prewitt ve Sobel operatörleridir. Bu operatörler aşağıda maskeler şeklinde tanımlanmıştır (Gonzalez ve Woods, 1993; Efford, 200):

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

G_x Maskesi

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

G_y Maskesi

Şekil 3.25 Prewitt operatörleri

-1	-2	-1
0	0	0
1	2	1

G_x Maskesi

-1	0	1
-2	0	2
-1	0	1

G_y Maskesi

Şekil 3.26 Sobel operatörleri

Sobel operatörlerinin aynı zamanda görüntüyü yumuşatma gibi bir etkisi de vardır. Türev alma işlemi gürültüyü geliştirdiği için, bu yumuşatma etkisi Sobel operatörlerini daha avantajlı duruma getirir. Bu ek etkiye rağmen gradient operatörleri en iyi performanslarını, keskin parlaklık geçişli ve az gürültülü görüntülerde gösterirler.

Kenar belirleme yöntemleri gri ton değerli görüntü oluşturlar. Bu görüntü her bir pikselin bir kenara ait olma derecesi hakkında bilgi içerir. Kenar detektörünün çıktısı büyükse lokal bir kenar mevcuttur. Aksi halde bu piksel arka plana aittir. Sadece kenar olarak tespit edilen piksellerin vurgulanması ve diğer piksellerin ortadan kalkması için eşikleme işlemi uygulamak gerekecektir. Bir gözlemci görüntü histogramını inceleyerek global bir eşik değeri belirleyebilir. Ancak bir çok görüntüde kenar detektörünün çıktısında farklı istatistiksel özelliklere sahip bölgeler bulunur. Bundan dolayı global eşikleme, görüntünün bazı bölgelerinde gereğinden daha kalın kenarlar gösterirken, bazı bölgelerdeki kenarları tamamen ortadan kaldırabilir. Bu yüzden, yerel olarak tanımlanabilen bir eşikleme yöntemi kullanmak gereklidir. Yerel yöntemde de eşikleme için (3.14) bağıntısı geçerlidir ancak görüntünün her bir pikseli için, bu pikselin bir komşuluğunda bir E eşik değeri hesaplanır. Bu değeri

belirlemede en çok kullanılan yöntem, (x,y) noktasının komşuluğundaki piksellerin gri ton değerlerinin

$$M = \left(\frac{1}{2a+1} \right)^2 \sum_{i=-a}^a \sum_{j=-a}^a f(x+i, y+j) \quad (3.33)$$

aritmetik ortalamasını hesaplamak ve bunu, eşik değerini belirlemek için kullanmaktır. Bu durumda eşik değeri

$$E = M(1 + p) \quad (3.34)$$

bağıntısıyla belirlenir. Burada p sayısı, eşik değerinin, yerel aritmetik ortalamanın ne kadar üzerinde olacağını gösteren bir sayıdır (Pitas, 1993).



4.MORFOLOJİ

Morfoloji veya biçimbilim kelimesi çoğunlukla biyolojinin, hayvanların ve bitkilerin biçimi ve yapısıyla ilgilenen dalını ifade eder. Dijital görüntü işleme konusu göz önüne alındığında matematiksel morfoloji, bir görüntüdeki nesnelerin geometrik şeklini ve yapısını incelemek için kullanılan bir araçtır. Morfolojik işlemler, kümeler teorisinin kavramları yardımıyla ifade edilirler. İşlemden geçirilecek görüntü bir küme olarak düşünülür. Morfolojik bir işlem, bu görüntü kümesinin, yapılandırma elemanı adı verilen küçük bir küme yardımıyla incelenmesi yoluyla tanımlanır. Yapılandırma elemanı görüntü üzerinde dolaştırılır ve uygulanan morfolojik işlemin tanımına göre, görüntüdeki piksel değerleri ile yapılandırma elemanının değerleri karşılaştırılarak, sonuç görüntüsünde kullanılacak değer elde edilir (Pitas, 1993).

Morfolojik işlemler genellikle ikilik görüntüler üzerinde tanımlanır. İkilik görüntü, sadece iki adet renk değerine sahip görüntülere verilen isimdir. Bu değerler 0 ve 1 ile gösterilirler. Morfolojik işlemler açısından bakıldığında ikilik görüntüdeki değerler, gerçek anlamda renk veya gri ton değeri olarak düşünülmezler. Bu değerler, arka plan için 0 (sıfır) ve görüntüdeki nesneler için 1 olmak üzere temsili bir anlam taşırlar. Bu düzenleme, eşikleme işlemi sonucunda oluşturulan görüntüleri akla getirmektedir. 0 (sıfır) ve L-1 değerlerinden oluşan bir görüntüde hangi değer arka planı, hangi değer nesneleri veya ilgi alanına giren parçaları gösterdiğine karar vermek gerekir. Örneğin, beyaz renk kağıt üzerine basılmış bir yazıcı çıktısı düşünüldüğünde, L-1 (beyaz) değeri arka planı oluşturduğundan 0 ile, 0 (siyah) değeri nesneleri oluşturduğundan 1 ile gösterilmelidir. Buna karşılık, herhangi bir kenar derektörü ile işleniş, sonra da eşikleme işlemi uygulanmış bir görüntü için, 0 (siyah) değeri arka planı oluşturduğundan 0 ile, L-1 (beyaz) değeri kenarları gösterdiğinden 1 ile ifade edilmelidir. Bu bölümde kullanılan açıklama amaçlı şekillerde arka plan açık renk ile, nesneler koyu renk ile gösterilmektedir.

4.1 Mantıksal İşlemler

Pikseller arasında mantıksal işlemler, görüntü işlemenin bir çok dalında kullanılır. Bu işlemler, sadece ikilik görüntüler üzerinde tanımlıdır. f_1 ve f_2 görüntüleri üzerinde mantıksal bir işlem uygulanmasıyla oluşan g görüntüsü

$$g(x, y) = f_1(x, y) \Delta f_2(x, y) \quad (4.1)$$

veya

$$g(x, y) = \Delta f_1(x, y) \quad (4.2)$$

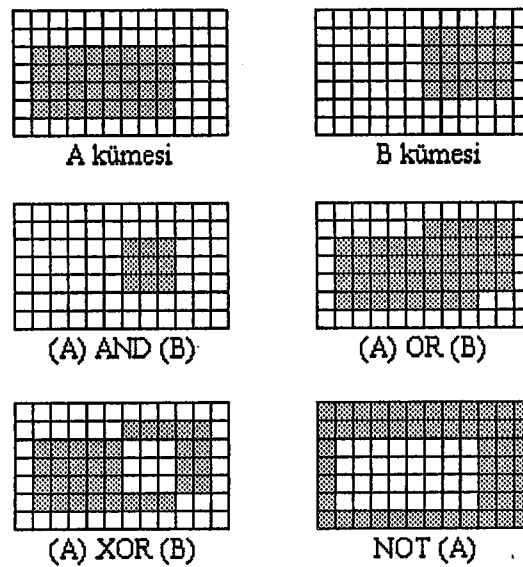
şeklinde tanımlanır. Burada Δ , herhangi bir mantıksal işlem operatörünü göstermektedir. Görüntü işlemede kullanılan temel mantıksal operatörler AND, OR, ve NOT operatörleridir. Bunların yardımıyla daha başka operatörler tanımlanabilir.

p ve q $\{0, 1\}$ değerlerinden birini alabilen pikseller olmak üzere, bazı mantıksal operatörlerin sonuçları aşağıdaki çizelgede verilmiştir.

Çizelge 4.1 Matıksal operatörlerin tanımları

p	q	$p \text{ AND } q$	$p \text{ OR } q$	$p \text{ XOR } q$	$\text{NOT } q$
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	
1	1	1	1	0	

Şekil 4.1'de iki görüntü üzerinde uygulanan bazı mantıksal işlemler görülmektedir.



Şekil 4.1 Görüntüler üzerinde mantıksal işlemler

4.2 Pikseller Arasında İlişkiler

4.2.1 Komşuluk

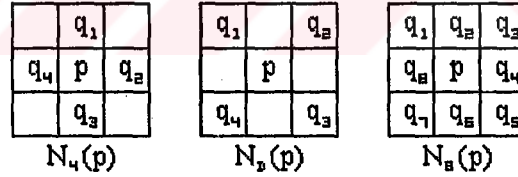
(x, y) koordinatlarındaki bir p pikselinin dört adet dikey ve yatay komşusu vardır. Bu komşuların koordinatları

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1) \quad (4.3)$$

ile verilir. Bu piksel kümesine p 'nin 4-komşuluğu adı verilir ve $N_4(p)$ ile gösterilir. p 'nin diagonal komşularının koordinatları

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) \quad (4.4)$$

ile verilir. Bu piksel kümesine p 'nin D-komşuluğu adı verilir ve $N_D(p)$ ile gösterilir. $N_4(p)$ ve $N_D(p)$ noktalarının hepsinin oluşturduğu kümeye p 'nin 8-komşuluğu adı verilir ve bu komşuluk $N_8(p)$ ile gösterilir. Şekil 4.2'de p pikselinin çeşitli komşulukları görülmektedir.



Şekil 4.2 Piksel komşulukları

4.2.2 Bağlılık

Pikseller arasında bağlılık, bir görüntüde nesnelerin ve bölgelerin saptanmasında kullanılan önemli bir kavramdır. İki pikselin bağlılığından söz edebilmek için önce bu bağlılığın temelini oluşturacak bir komşuluk kararlaştırmak gerekir. Bundan sonra, kararlaştırılan bu komşulukta bulunan iki pikselin gri ton değerlerinin, önceden tanımlanmış bir benzerlik kriterine uygunluğu kontrol edilir. Burada benzerliğin tanımı, iki pikselin gri ton değerlerinin

de, benzerlik kriteri olarak seçilen bir V kümesinin elemanı olmalarıdır. İkilik görüntüde benzerlik kümesi $V=\{1\}$ ile sınırlıdır.

Gri ton değerleri V 'nin elemanı olan p ve q pikselleri, eğer $q \in N_4(p)$ kümesinde ise 4-bağlıdır; $q \in N_8(p)$ kümesinde ise 8-bağlıdır. Bu iki bağıllık çeşidinden başka, m -bağıllık adı verilen karma bir tür de tanımlanabilir. Gri ton değerleri V 'nin elemanı olan p ve q pikselleri, eğer

a) $q \in N_4(p)$ kümesinde ise, veya

b) $q \in N_D(p)$ kümesinde ve $N_4(p) \cap N_4(q)$ kümesi boş ise (yani p ve q 'nin elemanları V kümesinden değilse)

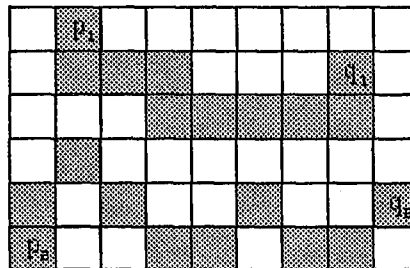
m -bağlıdır. p ve q pikselleri arasında bir bağıllık ilişkisi varsa, p pikseli q pikseline bitişiktir denir. Bir görüntünün S_1 ve S_2 gibi iki alt kümesinin bitişik olması için, S_1 'in herhangi bir pikselinin S_2 'nin herhangi bir pikseline bitişik olması yeterlidir (Gonzalez ve Woods, 1993).

4.2.3 Bölge

Bölgeyi tanımlamak için önce yol kavramını tanımlayalım. $p(x_0, y_0)$ pikselinde $q(x_n, y_n)$ pikseline yol,

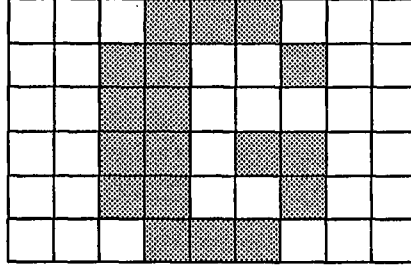
$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (4.5)$$

koordinatlarındaki pikseller kümesidir. Koordinatlar birbirlerinden farklıdır. $1 \leq i \leq n$ olmak üzere (x_i, y_i) pikseli (x_{i-1}, y_{i-1}) pikseline bitişiktir; yolun uzunluğu n 'dir. Kararlaştırılan bağıllık türüne göre 4-, 8-, veya m -yol tanımlanabilir. Şekil 4.3'de, bağıllık türünün yola nasıl etki ettiği görülmektedir. 4-komşuluk tanımlandığında sadece p_1 ve q_1 pikselleri arasında yol varken, 8-komşuluk tanımlandığında hem p_1 'den q_1 'e, hem de p_2 'den q_2 'ye yol vardır.



Şekil 4.3 Pikseller arasında yollar

p ve q'nun, bir görüntünün S altkümesinin iki pikseli olduğunu düşünelim. p'nin q'ye S içinde bağlı olma koşulu, p'den q'ye tamamen S içinde bulunan piksellerden oluşan bir yol bulunmasıdır. Aralarında bu şekilde bir yol bulunan bütün piksellerin oluşturduğu kümeye bölge adı verilir. Bu tanımda arka planda kalan bağlılığın önemi, Şekil 4.4'deki nesne göz önüne alındığında ortaya çıkar. Şekilde, 4-bağlılık tanımlanmış ise üç adet bölge, 8-bağlılık tanımlanmış ise bir adet bölge vardır.



Şekil 4.4 Örnek bir bölge

Bazı uygulamalarda, bir görüntüdeki bölgeleri numaralamak gerekebilir. Bu işleme ayrık bölgelerin etiketlenmesi adı verilir. Etiketleme işlemi, sıralı veya yinelemeli olabilir. İşlemlerin bilgisayarla yapıldığı düşünülürse, kullanılabilecek hafıza miktarının kısıtlı olduğu veya görüntünün tamamının hafızaya alınamayacak kadar büyük olduğu durumlarda sıralı işlemi kullanmak gerekir. Böyle kısıtlamalar yoksa, kendini çağıran fonksiyonları destekleyen bir programlama diliyle, yinelemeli işlemi uygulamak daha kolaydır.

Piksellerin 4-bağlılığı ile tanımlanan bölgeler için sıralı etiketleme algoritması aşağıdaki şekilde verilir. Görüntü solda sağa ve yukarıdan aşağı doğru taranır. Taramanın herhangi bir anında elde bulunan piksel p ile gösterilmektedir. Bu pikselin üstündeki ve solundaki pikseller sırayla r ve t ile gösterilmektedir. r'nin ve t'nin etiketleri sırayla L_r ve L_t ile gösterilmektedir. Burada, $p=0$ için herhangi bir işlem yapılmaz ve sonraki piksele geçilir. $p=1$ için, r'nin ve t'nin değerleri incelenir. Eğer $r=0$ ve $t=0$ ise, etiket matrisinde p'ye karşılık gelen koordinatlara yeni bir etiket değeri atanır. Eğer r ve t'den birisi 0 diğeri 1 ise, değeri 1 olanın etiketi p'nin etiketi olarak atanır. Eğer $r=1$ ve $t=1$ ise, etiketler kontrol edilir. $L_r=L_t$ durumunda bu etiket, p'nin etiketi olarak atanır. Burada işlemin uzamasına neden olan durum, L_r 'nin L_t 'den farklı olduğu durumdur. Bu durumda, etiketlerden herhangi biri p'nin etiketi olarak atanır ve ayrı bir yerde, L_r ve L_t etiketlerinin denk olduğunu gösteren bir kayıt tutulur. Bu yer bir matris olabilir. Örneğin L_r ve L_t 'nin denk olduğunu göstermek için, matrisin (L_r, L_t) ve (L_t, L_r)

elemanlarına 1 değeri yazılabilir. Denk etiketler bu şekilde tutuluyorsa, $L_r=L_t$ durumunda da matrisin (L_r , L_r) elemanına 1 değeri yazmak gerekecektir. Görüntünün bütün pikselleri tarandığında, değeri 1 olan pikseller etiketlenmiştir ancak, bu etiketlerden bazıları birbirine denktir. Etiketleme işlemini tamamlamak için, denkliklerin tutulduğu matris yardımıyla yeni bir etiketleme yapmak ve bu değerleri etiket matrisine yerleştirmek gerekir. 8-bağlılık için de aynı algoritma kullanılır fakat, bir pikselin sol ve üst komşuluklarına ek olarak, üst diagonalindeki iki komşuluğun da hesaba katılmasıyla, toplam dört adet pikselin değerleri ve etiketleri kontrol edilir. Denklik değerlerin, hafızada tutma ve sonra bunları yeniden etiketleme işlemi, 4-bağlılık durumunda olduğu gibidir (Gonzalez ve Woods, 1993).

Yinelemeli etiketleme algoritması temel olarak yığıt adı verilen veri yapısını kullanır. Bu yapıda, bir listeye son eklenen eleman, listeden bir eleman alınmak istendiğinde, listeden ilk çıkan eleman olacaktır. Etiketleme işleminde, görüntü taranırken eğer (x,y) koordinatlarındaki bir pikselin değeri 1 ise, etiket matrisinde (x,y) koordinatlarına yeni bir etiket değeri yazılır, (x,y) koordinatları yığta atılır, orijinal görüntünün (x,y) noktasındaki 1 değeri yerine 0 değeri yazılır ve (x,y)'nin, piksel değeri 1 olan bir komşuluğuna geçilip aynı işlemler tekrarlanır. Böyle bir komşuluk yoksa, yığıttan bir koordinat çifti alınır (yani bir piksel öncesine dönülür) ve işlemler bu noktadan denlenir. Etiketleme sonucunda orijinal görüntü tamamen ortadan kaybolur ve etiket matrisinde, ayırık bölgelerin etiket numaraları bulunur. Bu işlem bazı kaynaklarda 'çayır yangını' olarak adlandırılır. Piksel değeri 1 olan herhangi bir noktadan bir 'ateş' yakılır ve bu ateş, komşuluğundaki bütün bağlı noktalara yayılır. Bu bölgeye bağlı hiç bir piksel kalmadığında 'yangın' söner ve bölgenin pikselleri arka plandan ayrılamaz hale gelir (Pitas, 1993).

Yinelemeli fonksiyonları destekleyen bir programlama dilinde bu işlem basit bir fonksiyon yardımıyla gerçekleştirilebilir. Fonksiyonun yaptığı iş, gerekli değer atamalarını yaptıktan sonra uygun bir komşuluk bulup, bu komşuluğun koordinatlarını parametre olarak kullanıp kendini çağırmasıdır.

Bölge etiketleme işleminde kullanılan yinelemeli yöntem yardımıyla, bir görüntünün çeşitli büyüklükteki nesnelerini ayıklamak olanaklıdır. Bu işlem için etiketleme işlemi yapılmaz; bunun yerine fonksiyonun kaç defa kendini çağırdığı sayılır. Bu sayı aynı zamanda bu bölgenin kaç piksel büyüklüğünde olduğunu vermektedir. Bölgenin büyüklüğü belirlendikten sonra eğer bölge, önceden tanımlanmış bir büyüklük kriterine uygun değilse, başka bir

yinelemeli fonksiyon yardımıyla görüntüden silinebilir. Bu yöntemle büyüklük filtresi adı verilmektedir. İkilik görüntüdeki lekelerin boyutları yaklaşık olarak bilindiğinde ve bu lekelerin karmaşık şekiller aldığı durumlarda büyüklük filtresi, leke temizleme işleminde yardımcı olabilir.

4.3 İnceltme

Bir görüntüde bulunan nesnelerin temel yapısını belirlemek için bu nesneler, bir piksel kalınlıkta kalacak şekilde inceltilebilirler. Bu işlem, bir diski andıran nesnelerde iyi sonuç vermez. İdeal durumda, bir nesnenin inceltilmiş halini gösteren piksellerin, nesnenin ana eksenini üzerinde olması beklenir. İnceltme işlemi, nesnenin sınırlarındaki piksellerin silindiği ardışık işlemler biçiminde tanımlanabilir. Bu silme işlemi sırasında inceltilmiş nesnenin bütünlüğü korunmalıdır (Gonzalez ve Woods, 1993).

Burada tanımlanan ardışık yöntemde her bir iterasyon, iki adımdan oluşur. Bu adımlardan her biri, kendi içinde iki aşamada tamamlanır. Bu aşamalardan birincisinde, silinmeye aday pikseller işaretlenir, ikinci aşamada silme işlemi gerçekleştirilir. Her iki adım da uygulandıktan sonra, görüntüde herhangi bir değişiklik olup olmadığına bakılır. Eğer bir iterasyon sonucunda görüntünün hiç bir pikseli silinmemiş ise inceltme işlemi tamamlanmıştır. Eğer görüntüde bir değişiklik olmuşsa işlemler tekrarlanır.

Yöntemde pikseller Şekil 4.5'de görüldüğü gibi numaralanmıştır.

p_8	p_2	p_3
p_6	p_1	p_4
p_7	p_5	p_5

Şekil 4.5 İnceltme işlemi için piksel numaralaması

Brinci adımın ilk aşamasında görüntüdeki bir piksel şu koşulları sağlıyorsa silinmek üzere işaretlenir: a) $1 < N(p_1) < 7$, b) $S(p_1) = 1$, c) $p_2 p_4 p_6 = 0$, d) $p_4 p_6 p_8 = 0$. Burada $N(p_1)$, p_1 'in sifıra eşit olmayan komşularının sayısıdır; yani

$$N(p_1) = p_2 + p_3 + \dots + p_9 \quad (4.6)$$

$S(p_1)$ ise $p_2, p_3, \dots, p_8, p_9$, p_2 piksel sıralamasında 0-1 geçişlerinin sayısıdır. Bütün görüntü taranıp pikseller işaretlendikten sonra görüntü tekrar taranır ve bu sefer, işaretlenen pikseller silinir. Böylece ilk adım tamamlanır. İkinci adımda piksellerin işaretlenme koşulları a ve b için ilk adımla aynıdır. Bu adımda c ve d koşulları. c) $p_2p_4p_8=0$ ve d) $p_2p_6p_8=0$ olarak değişir. İşaretleme ve silme işlemleri sonucunda bir iterasyon tamamlanır. Şekil 4.6'da örnek bir nesne ve bunu inceltilmiş hali görülmektedir.



Şekil 4.6 A harfi; örnek bir nesne ve inceltmiş hali

4.4 Morfolojik İşlemlerde Kullanılan Temel Kavramlar

Morfolojik işlemlerin tanımlanmasında kümeler üzerinde tanımlanan çeşitli kavramlardan yararlanılır. Aşağıdaki tanımlarda A ve B Z^2 'de iki kümedir. A'nın bir elemanı $a(a_1, a_2)$ olarak, B'nin bir elemanı $b(b_1, b_2)$ olarak düşünülmektedir.

A'nın bir $x(x_1, x_2)$ ile ötelenmesi $(A)_x$ ile gösterilir ve

$$(A)_x = \{c \mid c = a + x, \text{ her } a \in A \text{ için}\} \quad (4.7)$$

ile tanımlanır.

A kümesinin tümleyeni A^c ile gösterilir ve

$$A^c = \{x \mid x \notin A\} \quad (4.8)$$

ile tanımlanır. İkilik görüntüler göz önüne alındığında A'nın tümleyeni ile A'nın mantıksal değili (NOT A) birbirine denktir.

B kümesinin yansıması B ile gösterilir ve

$$\hat{B} = \{x \mid x = -b, \text{ her } b \in B \text{ için}\} \quad (4.9)$$

ile tanımlanır. Burada, B kümesi için bir orijin noktası belirlenmelidir. Morfolojik işlemler kümelerin görüntü içindeki yeriyle değil biçimleriyle ilgili olduğundan, bu orijin noktası uygulamaya uygun herhangi bir şekilde seçilebilir.

A kümesinin B kümesinden farkı A-B ile gösterilir ve

$$A - B = \{x \mid x \in A \text{ ve } x \notin B\} = A \cap B^c \quad (4.10)$$

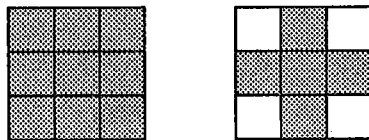
ile tanımlanır (Gonzalez ve Woods, 1993). İkilik görüntüler için

$$A - B = A \text{ AND } (\text{NOT } B) \quad (4.11)$$

yazılabilir.

4.5 Genleşme ve Aşınma

Genleşme ve aşınma işlemleri temel morfolojik işlemlerdir. Bu işlemlerde bir görüntü kümesi, yapılandırma elemanı adı verilen ikinci bir küme yardımıyla işlenir. Yapılandırma elemanı uygulamanın amaçlarına göre çok değişik biçimlerde olabilir ancak, Şekil 4.7'de görülen kare ve artı şeklindeki elemanlar bir çok alanda standart olarak kullanılmaktadır.

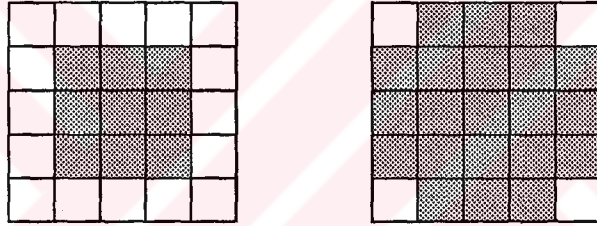


Şekil 4.7 Kare ve artı yapılandırma elemanları

A kümesinin B yapılandırma elemanı ile genişmesi $A \oplus B$ ile gösterilir ve

$$A \oplus B = \{x \mid (B)_x \cap A \neq \emptyset\} \quad (4.12)$$

ile tanımlanır. Bu durumda A'nın B ile genişmesi, B'nin yansıması alındıktan sonra bunun A ile kesişiminin en azından bir eleman verdiği öteleme noktaları kümesi olarak tanımlanır. Genleşme işlemi görüntüdeki nesnelerin yayılmasına neden olur. Nesnelerdeki çıkıntılar daha belirgin hale gelip kalınlaşır, nesnelerin içindeki boşluklar ve girintiler küçülür (Gonzalez ve Woods, 1993). Şekil 4.8'de örnek bir küme ve bu kümenin artı şeklinde yapılandırma elemanı ile genişmiş hali görülmektedir. Şekil 4.9'da ise bir nesne ve bu nesnenin kare şeklinde yapılandırma elemanı ile 3 defa genişmiş hali görülmektedir.



Şekil 4.8 Artı yapılandırma elemanı ile genişleme



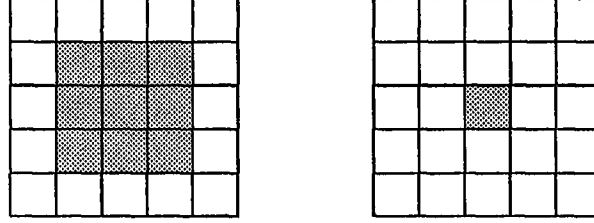
Şekil 4.9 B harfi nesnesinin kare yapılandırma elemanı ile genişmesi

A kümesinin B yapılandırma elemanı ile aşınması $A \ominus B$ ile gösterilir ve

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (4.13)$$

ile tanımlanır. Bu tanıma göre A'nın B ile aşınması, B yapılandırma elemanının tamamen A'nın içinde kalmasını sağlayan öteleme elemanları kümesi olarak tanımlanır. Aşına işlemi görüntüdeki nesnelerin daralmasında neden olur. Nesnelerdeki çıkıntılar kısılır ve nesnelerin içindeki boşluklar büyür. Şekil 4.10'da sol tarafta örnek bir küme ve sağ tarafta bu kümenin

artı yapılandırma elemanı ile aşındırılmış hali görülmektedir. Şekil 4.11’de ise örnek bir nesne ve bu nesnenin kare yapılandırma elemanı ile 3 defa aşınmış hali görülmektedir.



Şekil 4.10 Artı yapılandırma elemanı ile aşınma



Şekil 4.11 B harfi nesnesinin kare yapılandırma elemanı ile aşınması

Genleşme ve aşınma işlemleri, tümleme ve yansımaya göre birbirinin dualidir (Efford, 2000):

$$(A \oplus B)^c = A^c \otimes \hat{B} \quad (4.14)$$

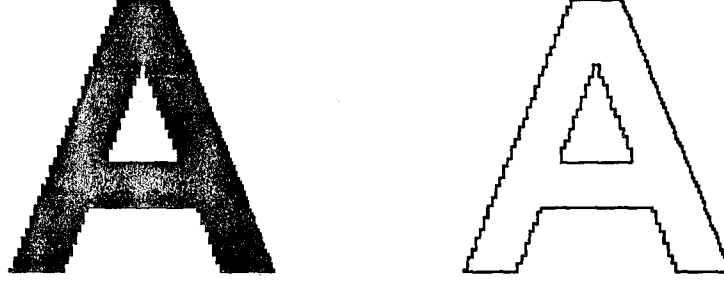
Genleşme ve aşınma işlemleri yardımıyla görüntüdeki nesnelerin sınırlarını belirlemek mümkündür. Aşınma işlemi yardımıyla nesnelerin iç sınırları, genleşme işlemi yardımıyla nesnelerin dış sınırları belirlenir. İç sınırlar, bir nesnenin arka planla ilişkisi olan piksellerinden oluşur. Bu sınırı belirlemek için

$$C = A - (A \oplus B) \quad (4.15)$$

görüntüsü oluşturulur. Bu görüntüdeki nesneler iç sınırlarına indirgenmişlerdir. Dış sınırlar ise, bir nesnenin en dışındaki piksellerle temas eden arka plan piksellerinden oluşur. Bu durumda dış sınırlar nesnenin etrafında bir çerçeve gibidir. Bu sınırları belirlemek için

$$C = (A \oplus B) - A \quad (4.16)$$

görüntüsü oluşturulur. Yapılandırma elemanı olarak 3×3 boyutlarında kare yapılandırma elemanı kullanılır. Şekil 4.12’de örnek bir nesne ve bu nesnenin iç sınırları görülmektedir.



Şekil 4.12 A harfi nesnesinin iç sınırları

4.6 Açılım ve Kapanım

Genel olarak aşınma ve genişleme, geri döndürülemeyen işlemlerdir. Bir aşınma işlemini takip eden genişleme işlemi ile orijinal nesne tekrar oluşturulamaz. Ancak birbirini takip eden bu işlemlerle yeni morfolojik dönüşümler tanımlanabilir.

Bir A kümesinin B yapılandırma elemanı ile açılımı $A \circ B$ ile gösterilir ve

$$A \circ B = (A \oplus B) \oplus B \quad (4.17)$$

ile tanımlanır. Açılım işlemi sonucunda nesnelerdeki ince çizgiler ortadan kalkar, ufak benekler yok olur, nesnelerin keskin ayrıntıları yumuşar. Bu işlem sonucunda yapılandırma elemanından daha ince bağlarla birbirine bağlanmış bir nesneler, birden çok bölgeye ayrılabilir. Açılım işlemi geometrik olarak, bir yapılandırma elemanını bir bölgenin içinde gezdirmeye ve bu elemanın bölge içinde ulaştığı yerlerin sınırlarını çizme işlemine benzetilebilir. Bu durumda bölgenin içine doğru giren detaylar göreceli olarak sağlam kalırken, ufak çıkıntılar ortadan yok olacaktır (Gonzalez ve Woods, 1993).

Bir A kümesinin B yapılandırma elemanı ile kapanımı $A \bullet B$ ile gösterilir ve

$$A \bullet B = (A \oplus B) \ominus B \quad (4.18)$$

ile tanımlanır. Kapanım işlemi sonucunda nesnelerdeki ufak girintiler ve ince uzun 'körfezler' ortadan kalkar, küçük delikler yok olur. Kapanım işlemi geometrik olarak, bir yapılandırma elemanını bir bölgenin dışında gezdirmeye benzetilebilir. Bu durumda bölgenin dışına doğru

taşan detaylar özelliklerini biraz kaybetmelerine rağmen göreceli olarak sağlam kalırlar ancak, bölgenin sınırlarındaki girintiler yok olurlar (Gonzalez ve Woods, 1993).

Açılım ve kapanım işlemleri tümleme ve yansımaya göre birbirinin dualidir (Pitas, 1993):

$$(A \bullet B)^c = (A^c \circ \hat{B}) \quad (4.19)$$

Bu işlemler ayrıca aşağıdaki özelliklere sahiptir (Pitas, 1993):

$$A \circ B \subseteq A \quad (4.20)$$

$$A \subseteq A \bullet B \quad (4.21)$$

$$(A \circ B) \circ B = A \circ B \quad (4.22)$$

$$(A \bullet B) \bullet B = A \bullet B \quad (4.23)$$

Son iki özelliğe göre, aynı yapılandırma elemanı ile arka arkaya uygulanan açılım (veya kapanım) işleminin görüntü üzerinde bir etkisi yoktur.

Açılım işleminin küçük benekleri yok etme ve kapanım işleminin küçük delikleri doldurma özelliği göz önüne alınırsa, bu işlemlerin ikilik görüntülere uygulanması sonucunda gürültü temizleme etkisi elde edilebilir. Açılım işlemini takip eden kapanım işlemi sonucunda elde edilen

$$C = (A \circ B) \bullet B \quad (4.24)$$

C görüntüsü, morfolojik bir filtreyle işleminden geçirilmiştir (Gonzalez ve Woods, 1993).

5.SONUÇLAR

Bilgisayar ortamı söz konusu olduğunda dijital görüntüler için bir çok format kullanılmaktadır. Günümüzde en çok karşılaşılan görüntü formatlarına örnek olarak BMP ve JPG formatları örnek verilebilir. Farklı amaçlar için farklı görüntü formatı kullanma zorunluluğu ve formatlar arasındaki dönüşümlerin karmaşıklığı, görüntü işleme uygulamalarına ek işlem yükü getirmektedir. Bilgisayarın işletim sistemi, dijital görüntüyü ekranda gösterebilmek için piksel başına üç adet renk değerine ihtiyaç duyar. BMP görüntü formatı bu ihtiyaca doğrudan cevap vermektedir ancak bu format hem renkli görüntü işleme uygulamaları için uygun değildir hem de görüntüyü saklamak için büyük miktarda yer gerektirmektedir. Günümüzde renkli görüntü saklamak için en çok kullanılan formatlardan biri JPG formatıdır. Her ne kadar görüntüde bazı kayıplar olsa da JPG'nin sağladığı sıkıştırma oranı, görüntüyü saklamak için harcanan yerden büyük tasarruf sağlar. Ancak bu görüntüyü ekranda göstermek için tekrar piksel başına üç renk değeri içeren bir biçime getirmek gereklidir. Ayrıca bu formattaki görüntü üzerinde doğrudan görüntü işleme uygulaması yapılamaz. Renkli görüntü uygulamaları için kullanılan en basit format, BMP'den lineer bir dönüşümle elde edilen YIQ formatıdır. Renkli nesneler üzerinde işlem yapmak için genellikle HSI formatı kullanılır. Her iki format da renkli görüntü uygulamalarına uygun olmalarında rağmen, görüntü saklama konusunda bir avantajları yoktur ve görüntüyü göstermek için uygun değildir. Görüntü işleme konusunda ilk verilmesi gereken karar, hangi görüntü formatının kullanılacağıdır.

Görüntü geliştirme tekniklerinin amacı, bir görüntüyü özel bir uygulama alanı için orijinalinden daha uygun bir hale getirmektir. Burada kullanılan 'özel bir uygulama alanı' sözcükleri önemlidir çünkü bu sözler, görüntü geliştirme uygulamalarının büyük oranda probleme yönelik olduğunu göstermektedir. Örneğin bir röntgen görüntüsünü geliştirmek için kullanılan teknikler, dünya çevresinde dönen bir meteoroloji uydusundan gönderilen görüntüleri geliştirmek için kullanılan tekniklerden tamamen farklı olabilir. Ayrıca insan açısından bakıldığında, her durumda iyi bir görüntünün tanımını yapmak zordur. Bu koşullar göz önüne alındığında, görüntü geliştirme tekniklerinin deneme-yanılma yöntemine dayalı yapısı daha çok öne çıkmaktadır.

Deneye dayalı bu yapının sonucunda, görüntü geliştirme tekniklerinin verdiği sonuçların ne kadar işe yarayacağı her durumda önceden tahmin edilemez. Buna en iyi örnek, histogram

TEKİRİMSİZ GÖRÜLTÜ KURULU
DOKÜMANTASYON MERKEZİ

eşitleme tekniğidir. Histogram eşitleme tekniği, düşük kontrastlı görüntülerdeki nesnelerin ve detayların ortaya çıkarılmasında çok etkileyici sonuçlar vermesine rağmen, istatistiksel yöntemlere dayandığı için, özellikle piksel sayısı göreceli olarak fazla olan gri ton değerinin görüntünün önemli parçalarını oluşturmadığı durumlarda, başarılı sonuçlar vermemektedir. Beklenmeyen veya uygulamanın ihtiyaçlarını karşılamayan sonuçların alındığı bir başka alan, eşikleme işlemidir. Eşikleme işleminin amacı ilgi alanına giren nesneleri gösteren pikselleri, diğer kısımları gösteren piksellerden ayırmaktır. Bu işlem için tanımlanan algoritmaların verdiği sonuçların bir gözlemci tarafından kontrol edilmesi ve düzeltilmesi gerekebilir.

Eşikleme işleminin hassaslığı, eşiklemenin çoğu zaman segmentasyon işleminin önemli bir adımını oluşturmasında kaynaklanır. Kenar belirleme işlemlerinde de benzer bir zorluk söz konusudur. Görüntünün içindeki bir nesnenin kenarlarının kesin matematiksel tanımı yapılamamaktadır. Bu yüzden kenar detektörleri tarafından bulunan değerlerin yorumlanmasında çeşitli hatalar ortaya çıkmaktadır. Bazı durumlarda otomatik eşikleme işlemi, kenar olması gereken bir kesimi çevresindeki piksellerden ayırmakta yetersiz kalmaktadır.

Morfolojik işlemler, herhangi bir çizgi veya kenar detektörünü temsil eden bir filtreyle işlenmiş ve kaba bir eşikleme işleminden geçirilmiş görüntülerde, istenen bölgeleri vurgulamak amacıyla yaygın olarak kullanılırlar. Matematiksel morfoloji, ikilik görüntüler üzerinde işlem yapmak için kuvvetli araçlar sağlar ancak bu yöntemlerin dezavantajı, işlemlerin çokluğu ve karmaşıklığı yüzünden uygulama süresinin uzamasıdır.

KAYNAKLAR

Efford, N., (2000), Digital Image Processing, Addison-Wesley, Harlow.

Emiroğlu, İ., (1997), Fingerprint Image Enhancement & Recognition, Doktora Tezi, University of Hertfordshire.

Gonzalez, R. C. ve Woods, R. E., (1993), Digital Image Processing, Addison-Wesley, Reading.

Jain, R., Kasturi, R., Schunk, B. G., (1995), Machine Vision, McGraw-Hill, New York.

Lindley, C. A., (1995), Photographic Imaging Techniques in C++, John Wiley & Sons, New York.

Pitas, I., (1993), Digital Image Processing Algorithms, Prentice Hall, Hertfordshire.

Shammas, N. C., (1994), Visual C++ 2 in 21 Days, Sams Publishing, Indianapolis.



ÖZGEÇMİŞ

Doğum tarihi	29.11.1972	
Doğum yeri	İstanbul	
Lise	1983-1991	Saint Benoit Lisesi
Lisans	1992-1997	Yıldız Teknik Üniversitesi Kimya-Metalurji Fak. Matematik Mühendisliği Bölümü
Yüksek Lisans	1998-2001	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Müh. Anabilim Dalı

