

**BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**LAZER İŐİNİ İLE İŐARETLENEN HAREKETLİ CİSİMLERİ
HEDEFE KİTLENEREK İZLEYEN SİSTEM**

EVİRİM GÜNER

**YÜKSEK LİSANS TEZİ
2016**

**LAZER IŞINI İLE İŞARETLENEN HAREKETLİ CİSİMLERİ
HEDEFE KİTLENEREK İZLEYEN SİSTEM**

**A LOCKING AND TRACKING SYSTEM FOR LASER
POINTED MOVING OBJECTS**

EVİRİM GÜNER

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK – ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2016

“Lazer Işını ile İşaretlenen Hareketli Cisimleri Hedefe Kitlenerek İzleyen Sistem” başlıklı bu çalışma, jürimiz tarafından, 29/01/2016 tarihinde, **ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan (Danışman) : Prof. Dr. Murat Emin AKATA

Üye : Prof. Dr. Hasan OĞUL

Üye : Doç. Dr. Hasan Şakir BİLGE

ONAY

04/02/2016

Prof. Dr. Murat Emin AKATA
Fen Bilimleri Enstitüsü
Müdürü



**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU**

Tarih: 02 / 02 / 2016

Öğrencinin Adı, Soyadı : Evrim GÜNER

Öğrencinin Numarası : 21220061

Anabilim Dalı : Elektrik – Elektronik Mühendisliği Anabilim Dalı

Programı : Elektrik – Elektronik Mühendisliği Tezli Yüksek Lisans Programı

Danışmanın Adı, Soyadı : Prof. Dr. Murat Emin AKATA

Tez Başlığı : Lazer Işını İle İşaretlenen Hareketli Cisimleri Hedefe Kitlenerek İzleyen Sistem

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 137 sayfalık kısmına ilişkin, 02/02/2016 tarihinde şahsım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 6'dır.

Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimedenden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası

Onay

04/02/2016

Öğrenci Danışmanı
Prof. Dr. Murat Emin AKATA

TEŐEKKÜR

Bu tez alıőmasının ortaya ıkmasında byk emeđi olan, tecrbe ve zamanını benimle paylaőarak yol gsterici olan deđerli hocam ve tez danıőmanım Prof. Dr. Emin AKATA'ya;

Tez alıőmalarım sresince her zaman yanımda olan ve beni srekli destekleyen niőanlım Fatma Zehra SZMEN'e;

Hayatım boyunca bana her konuda maddi ve manevi destek veren, bugnlere gelmemi sađlayan aileme; desteđini her zaman hissettiđim abim Emre GNER'e, zor anlarda hep yanımda olan annem Defne GNER'e, bilgi ve yardımlarını hibir zaman esirgemeyerek bana ıőık tutan babam Mh. Mustafa Uđur GNER'e sonsuz teőekkr ederim.

ÖZ

LAZER IŞINI İLE İŞARETLENEN HAREKETLİ CİSİMLERİ HEDEFE KİTLENEREK İZLEYEN SİSTEM

Evrinm GÜNER

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Bu çalışmada, lazer ışını ile işaretlenen hareketli cisimleri hedefe kitlenerek izleyen sistem tasarımı gerçekleştirilmiştir. Sistem, görüntü işleme tekniklerini uygulayan kısmı bilgisayar, kamerayı görüntü izleme algoritmasından gelen bilgiye göre x ve y düzlemlerinde döndürerek, nesnenin görüntü alanının merkezinde kalmasını sağlayan kısmı hedef izleme ünitesi olmak üzere iki bölüm altında incelenmiştir.

Bilgisayar üzerinde, lazer ışınının algılanması, nesnenin bölütlenmesi ve gerçek zamanlı olarak izlenmesi işlemleri yapılır. Lazer ışınının algılanmasında, kullanılan kırmızı lazerin renk özelliklerinden ve lazer ışınının görüntüdeki yüksek parlaklık seviyesinden yararlanır. Lazer ışınının belirlenmesinden sonra, ışının son düştüğü noktada bölgesel tabanlı bölütleme yapılarak izlenecek nesne görüntüden ayrılır. Nesnenin merkezine yerleştirilen bir kare bölgenin, sürekli olarak büyütülmesi ile nesne içerisinde kalan en büyük kare alan bulunur ve bu alan içerisindeki bölgenin H, S, V değerleri ile histogramı elde edilir. Nesne izleme, elde edilen veriler kullanılarak uygulanan CAMSHIFT algoritması ile gerçekleştirilir.

Hedef izleme ünitesi, bilgisayardan gelen bilgiyi kullanarak 2 servo motor yardımı ile sisteme hareket kazandırır. Hedef izleme ünitesi ile bilgisayar arasındaki haberleşme USB veri yolu üzerinden gerçekleştirilir.

Tasarlanan sistem 9 farklı durum altında test edilmiştir. Bu testler sonucunda sistemin, saniyede ortalama 18.9 çerçeve işlediği ve % 95.6'lık izleme başarısı elde ettiği görülmüştür.

ANAHTAR SÖZCÜKLER: Görüntü İşleme, Nesne Takibi, Hareketli Nesne Takibi, Sürekli Uyarlamalı Ortalama Değer Kayma Algoritması (CAMSHIFT)

Danışman: Prof. Dr. Murat Emin AKATA, Başkent Üniversitesi, Elektrik – Elektronik Mühendisliği Bölümü

ABSTRACT

A LOCKING AND TRACKING SYSTEM FOR LASER POINTED MOVING OBJECTS

Evrin GÜNER

Başkent University Institute of Science

Department of Electrical and Electronics Engineering

In this study, a locking and tracking system for laser pointed moving objects has been developed. The system investigated as two parts; the computer performs all the image processing techniques and the target tracking unit performs moving the camera on the x – y axis to keep the object at the center of each frame.

The computer performs detection of the laser beam, segmentation of the object to be tracked and the real time tracking of the object. Detection of the laser beam is done by using the color properties of red laser and the high level intensity of laser beam on the image. After the detection of laser beam on the image, the last location of laser beam is used to segment the object from the image. The segmentation is done by using region based segmentation on the last location of laser beam. A small square placed at the center of the object is exaggerated until it passes over the object region to achieve the biggest square bounds the inner region of the object. This square then, used to get the H, S, V values and the histogram. Object tracking is done by using H, S, V values and the histogram with CAMSHIFT algorithm.

The target tracking unit uses 2 servo motors to keep the system moving, based on the information of object location comes from the computer. USB interface is used to establish the communication between the computer and the target tracking unit.

The developed system has been tested for 9 different conditions. The system worked with an average performance of 18.9 frames per second and a success rate of 95.6 %.

KEYWORDS: Image Processing, Object Tracking, Moving Object Tracking, Continuously Adaptive Mean – Shift (CAMSHIFT) Algorithm

Advisor: Prof. Dr. Murat Emin AKATA, Başkent University, Department of Electrical and Electronics Engineering

İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
İÇİNDEKİLER LİSTESİ	iii
ŞEKİLLER LİSTESİ.....	vi
ÇİZELGELER LİSTESİ.....	ix
SİMGELER VE KISALTMALAR LİSTESİ	x
1. GİRİŞ.....	1
2. SAYISAL GÖRÜNTÜ İŞLEME.....	6
2.1. Sayısal Görüntü	6
2.2. RGB Renk Uzayı	9
2.3. HSV Renk Uzayı	10
2.4. RGB ve HSV Renk Uzaylarının Avantaj ve Dezavantajları	11
2.5. RGB – HSV Renk Uzayı Dönüşümleri.....	12
2.5.1. RGB'den HSV'ye dönüşüm algoritması	12
2.5.2. HSV'den RGB'ye dönüşüm algoritması	13
2.6. Morfolojik İşlemler	14
2.6.1. Aşındırma (Erosion).....	15
2.6.2. Yayma (Dilation)	16
2.6.3. Açma (Opening).....	18
2.6.4. Kapama (Closing)	18
2.7. Görüntü Histogramı	19
2.8. Uzamsal Süzme	21
2.9. Görüntü Bulanıklaştırma / Yumuşatma (Blurring / Smoothing).....	23
2.9.1. Doğrusal süzgeç ile yumuşatma	23
2.9.2. Sıra istatistiksel (Order statistic – Doğrusal olmayan) süzgeçler	26
2.10. Pikseller Arası Komşuluk (Pixel Neighborhood)	26
2.10.1. İki boyutlu komşuluk	27
2.10.1.1. 4-komşusu	27
2.10.1.2. 8-komşusu	27
2.10.2. Üç boyutlu komşuluk	28
2.10.2.1. 6-komşusu	28
2.10.2.2. 18-komşusu	29

2.10.2.3. 26-komşusu	29
2.11. Görüntü Bölütleme (Segmentation).....	29
2.11.1. Görüntü bölütlemenin temelleri.....	30
2.11.2. Görüntü eşikleme (Thresholding) ile bölütleme	31
2.11.3. Kenar tabanlı bölütleme.....	33
2.11.4. Bölgesel tabanlı bölütleme	36
2.11.5. Karma bölütleme	38
2.11.6. Harekete dayalı bölütleme.....	38
3. NESNE İZLEME YÖNTEMLERİ.....	40
3.1. Nesne Gösterimi	40
3.2. İzleme İçin Özellik Seçimi.....	41
3.3. Nesne Belirleme	42
3.3.1. Nokta belirleyiciler (Point detectors)	44
3.3.2. Arka plan çıkarımı	44
3.3.3. Bölütleme.....	44
3.3.4. Öğretici ile öğrenme.....	45
3.4. Nesne İzleme	47
3.4.1. Nokta izleme	48
3.4.1.1. Deterministik yöntemler.....	49
3.4.1.2. İstatistiksel yöntemler	50
3.4.2. Çekirdek izleme	52
3.4.2.1. Şablon ve yoğunluk esaslı görünüm modeli	53
3.4.2.2. Çoklu görüş görünüm modelleri	54
3.4.3. Siluet izleme	54
3.4.3.1. Şekil eşleştirme	55
3.4.3.2. Çevre çizgisi izleme	55
4. KULLANILAN YAZILIM VE DONANIMLAR	56
4.1. C++	56
4.2. OpenCV	56
4.3. Arduino IDE	58
4.4. Logitech® HD Pro Webcam C920.....	58
4.5. PhantomX Robot Turret	59
4.5.1. Arbotix – M robot kontrolcüsü	60
4.5.2. Dynamixel AX – 18A servo motor	61
5. YAPILAN ÇALIŞMALAR	63

5.1. Sistemin Genel İncelemesi.....	63
5.2. Bilgisayar.....	65
5.2.1. Lazer izleme	68
5.2.2. Nesne belirleme.....	72
5.2.3. Nesne izleme	75
5.3. Hedef İzleme Ünitesi	85
5.4. Sistemin Başarılı Çalışması İçin Gereken Ön Koşullar.....	89
5.5. Sistemin Performans Analizi	91
5.6. Bulgular ve Tartışma	107
6. SONUÇLAR	111
6.1. Gelecek Çalışma Planları.....	113
KAYNAKLAR LİSTESİ	115

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1 Sayısal Görüntü ([26]'dan Türkçeleştirilerek).....	6
Şekil 2.2 Piksellerin Matris Gösterimi	7
Şekil 2.3 Bit Sayısı, Detay İlişkisi. (a) 24-bit renkli, (b) 8-bit renkli, (c) 8-bit gri seviye, (d) 1-bit siyah/beyaz [27]	8
Şekil 2.4 Renkli Görüntünün Oluşumu ([26]'dan Türkçeleştirilerek)	9
Şekil 2.5 RGB Renk Küpü ([26]'dan Türkçeleştirilerek)	10
Şekil 2.6 HSV Renk Konisi [29]	11
Şekil 2.7 Miss, Hit ve Fit (C: Miss, B: Hit, A: Fit) [26].....	15
Şekil 2.8 Aşındırma örneği. Orijinal görüntü (a), yapı elemanı(b), aşındırılacak noktalar (c), işlenmiş görüntü (d) [34]	16
Şekil 2.9 Yapışmış nesnelerin ayrılması(a), çıkıntıların yok edilmesi(b) [26]	16
Şekil 2.10 Yayma örneği. Orijinal görüntü (a), yapı elemanı (b), genişletilecek noktalar (c), işlenmiş görüntü (d) [34]	17
Şekil 2.11 Kopuklukların onarılması(a), girintilerin yok edilmesi(b) [26].....	17
Şekil 2.12 Açma işlemi. Orijinal şekil (a), açma sonrası (b) [35]	18
Şekil 2.13 Kapama işlemi. Orijinal şekil (a), kapama sonrası (b) [35]	18
Şekil 2.14 Görüntü Histogramı	19
Şekil 2.15 Düşük değerlerde toplanan histogram grafiği	20
Şekil 2.16 Yüksek değerlerde toplanan histogram grafiği.....	20
Şekil 2.17 Orta değerlerde toplanan histogram grafiği	20
Şekil 2.18 Histogram eşitleme uygulanmış bir görüntü (Şekil 2.15'ye uygulanmıştır).....	21
Şekil 2.19 Uzamsal süzme ve komşuluk (Neighbourhood)	21
Şekil 2.20 3 x 3 komşuluk ile uzamsal süzme teknikleri	23
Şekil 2.21 3 x 3 yumuşatma (ortalama alan) süzgeç maskesi	24
Şekil 2.22 3 x 3 ağırlıklı (weighted) ortalama alan süzgeç maskesi.....	24
Şekil 2.23 Görüntü yumuşatma. Orijinal görüntü (a), yumuşatılmış görüntü (b) ...	25
Şekil 2.24 Gürültülü orijinal görüntü (a), 3 x 3 ortalama alma süzgeci uygulanmış görüntü (b), 3 x 3 orta değer süzgeci uygulanmış görüntü (c).....	26
Şekil 2.25 4-komşuluğun bağlantı yönleri	27
Şekil 2.26 p pikselinin 4-komşuluğu $N_4(p)$	27
Şekil 2.27 8-komşuluğun bağlantı yönleri	28
Şekil 2.28 p pikselinin 8-komşuluğu $N_8(p)$	28
Şekil 2.29 6-komşuluğun bağlantı yönleri	28
Şekil 2.30 18-komşuluğun bağlantı yönleri	29
Şekil 2.31 26-komşuluğun bağlantı yönleri	29
Şekil 2.32 Görüntü eşikleme örneği. Eşikleme öncesi görüntü (a), eşikleme sonrası elde edilen siyah beyaz görüntü (b).....	32
Şekil 2.33 Görüntü eşikleme uygulanan histogram grafiği. Arka plan, nesne ve sınıflandırılmayan pikseller	32
Şekil 2.34 Laplace operatörü maskesi.....	34
Şekil 2.35 Çizgi belirleme maskeleri, yatay, +45°, dikey ve -45°	35
Şekil 2.36 Roberts bayırgı işleci maskeleri	35
Şekil 2.37 Prewitt ve Sobel işleçleri	36

Şekil 2.38 Bölgesel bölütleme, tohumların seçilmesi (a), küçük grupların oluşturulması (b), küçük grupların büyütülerek büyük bölgeleri oluşturması (c) ...	37
Şekil 2.39 Su bölümü çizgisi yöntemi ([44]'den Türkçeleştirilerek)	38
Şekil 3.1 Nesne gösterimleri. Kütle merkezi (a), çoklu noktalar (b), dikdörtgen şekil (c), eliptik şekil (d), parça temelli çoklu şekil (e), nesne iskeleti (f), çevre çizgisi kontrol noktaları (g), tüm çevre çizgisi (h), nesne silueti (i)	41
Şekil 3.2 Destek vektör makinesi.....	46
Şekil 3.3 İzleme yöntemlerinin sınıflandırılması ([45]'den Türkçeleştirilerek)	47
Şekil 3.4 Çok noktalı izleme	49
Şekil 3.5 Hareket sabitleri. Yakınlık (a), maksimum hız (b), küçük hız değişimi (c), alışılmış hareket (d), katılık (e)	50
Şekil 3.6 Kalman süzgeci çalışma şeması.....	51
Şekil 3.7 Kalman süzgeci örneği	52
Şekil 3.8 Ortalama değer kayması tekrarları. $t-1$ anında hesaplanmış nesne pozisyonu (a), bir önceki nesne pozisyonu başlangıç değeri alınan t anındaki çerçeve (b), ortalama değer kayması tekrarları ile pozisyon güncelleme (c, d, e), t anındaki son pozisyon (f)	54
Şekil 3.9 Çevre çizgisi değişimi örnekleri	55
Şekil 4.1 Logitech® HD Pro C920 web kamera.....	59
Şekil 4.2 PhantomX Robot Turret.....	60
Şekil 4.3 Arbotix-M Robot Kontrolcüsü	60
Şekil 4.4 Dynamixel aktüatör hepsi bir arada yapı.....	61
Şekil 4.5 Dynamixel AX - 18A aktüatör.....	62
Şekil 5.1 Sistemin blok şema gösterimi	64
Şekil 5.2 Sistemin nesne izleme sırasında görünümü	64
Şekil 5.3 Bilgisayar blok şema gösterimi	65
Şekil 5.4 Bilgisayar ekran görüntüsü	65
Şekil 5.5 Algoritma başlangıcı. Görüntü yakalama kısmı	67
Şekil 5.6 Lazer izleme bölümü akış diyagramı	68
Şekil 5.7 Kırmızı lazer ışınının görünümü. Yanlış pozlandırma süresi ile görünüm (a), doğru pozlandırma süresi ile görünüm (b), (b)'nin gri seviye gösterimi (c).....	69
Şekil 5.8 Lazer izleme bölümünde, kameranın pozlandırma süresi ve kazanç değerlerinin değiştirilmesi ile görüntünün parlaklık seviyesinin düşürülmesi	69
Şekil 5.9 Parlaklığı düşürülmüş görüntü üzerinde lazer ışınının belirlenmesi	71
Şekil 5.10 Çoklu renk modu gösterimi. Orijinal görüntü (a), çoklu renk modu ile takip edilmek üzere renk bilgisi çıkarılacak alan seçimi (b), seçilen alanın HSV gösterimi (c).....	72
Şekil 5.11 Nesne belirleme bölümü akış diyagramı	73
Şekil 5.12 Cismin merkezine yerleştirilen 10 x 10 piksellik karenin büyütülerek renk ortalaması hesaplanması. Kare cismin içinde (a), kare cismin dışında (b) ...	74
Şekil 5.13 Nesne izleme bölüme akış diyagramı	76
Şekil 5.14 Nesne olarak seçilen bölgenin HSV renk uzayında gösterimi (a), nesnenin histogram gösterimi (b)	77
Şekil 5.15 Geriye izdüşüm. Renkli mozaik görüntü (a), (a) görüntüsünün hue kanalı (b), (b)'nin histogramı (c), histogramın görüntü içerisinde benzerlik sağladığı noktalar, geriye izdüşüm (d) [77].....	78

Şekil 5.16 Geriye izdüşüm örneği. Orijinal görüntü(a), geriye izdüşüm görüntüsü (b).....	79
Şekil 5.17 Geriye izdüşüm görüntüsü üzerinde birleşik elemanların en büyüğünün bulunması	80
Şekil 5.18 Meanshift örneği [78]	81
Şekil 5.19 Meanshift ile izlemede, cismin boyutunun değişimine bağlı problemin oluşumu [78]	82
Şekil 5.20 CAMSHIFT ile izlemede, cismin boyutunun değişimine bağlı problemin çözümü [78]	82
Şekil 5.21 Nesnenin CAMSHIFT uygulaması ile izlenmesi. Mavi dörtgen arama bölgesini, kırmızı dörtgen ise izlenen nesneyi göstermektedir.....	83
Şekil 5.22 640 x 480 piksellik bir çerçevede, sistemin hareket yönünün belirlenmesi için kullanılan bölgeler	85
Şekil 5.23 Hedef izleme ünitesi blok şema gösterimi.....	86
Şekil 5.24 Hedef izleme ünitesi görünümü	87
Şekil 5.25 Hedef izleme ünitesi akış diyagramı	87
Şekil 5.26 Tek renkli veya hafif renk geçişleri olan arka plan önünde rastgele hareketler ile sallanan tenis topunun izlenmesi	92
Şekil 5.27 İzleme ekranı, izlenecek olan nesnenin içerisinde kalan en büyük kare alan ve bu alan oluşturulan histogramı	93
Şekil 5.28 Ortalama hızı 1.85 m/sn olan topun izlenmesi	94
Şekil 5.29 Masa üzerinde sektirilen topun izlenmesi	95
Şekil 5.30 Ortalama hızı 1.20 m/sn olan topun izlenmesi	97
Şekil 5.31 Masa üzerinde sektirilen topun izlenmesi	98
Şekil 5.32 Karışık renkli arka plan önünde tenis topunun izlenmesi	100
Şekil 5.33 Karışık renkli arka plan önünde oyuncağın izlenmesi	102
Şekil 5.34 Çoklu renk modu izlenecek alanın seçimi (a), seçilen alanın H,S,V görüntüsü (b) ve histogramı (c)	104
Şekil 5.35 Karışık renkli arka plan önünde bir kitabın çoklu renk modu kullanılarak izlenmesi	105
Şekil 5.36 Çoklu renk modu ile seçilen alanın H,S,V görüntüsü (a) ve histogramı (b).....	106
Şekil 5.37 Karışık renkli arka plan önünde bir kutunun çoklu renk modu kullanılarak izlenmesi	106

ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 Bit Sayısı, Ton Sayısı İlişkisi.....	7
Çizelge 3.1 Nesne belirleme kategorileri	43
Çizelge 3.2 İzleme kategorileri [45]	47
Çizelge 6.1 Sistemin performans analizi sonuçları	113

SİMGELER VE KISALTMALAR LİSTESİ

HSI	Renk, doygunluk ve yoğunluk renk uzayı (Hue, Saturation, Intensity)
RGB	Kırmızı, yeşil ve mavi renk uzayı (Red, Green, Blue)
CAMSHIFT	Sürekli uyarlamalı ortalama değer kayması (Continuously Adaptive Mean-Shift)
GPU	Grafik işlem birimi (Graphics processing unit)
LBP	Yerel ikili örüntü (Local binary pattern)
USB	Evrensel seri veri yolu (Universal serial bus)
PWM	Sinyal genişlik modülasyonu (Pulse width modulation)
GB	Gigabyte (1024 Megabyte)
RAM	Rastgele erişimli bellek (Random access memory)
HSV	Renk, doygunluk ve parlaklık renk uzayı (Hue, Saturation, Value)
bpp	Piksel başına düşen bit sayısı (Bits per pixel)
T	Eşik değeri (Threshold value)
IDE	Bütünleşmiş geliştirme ortamı (Integrated development environment)
HD	Yüksek çözünürlük (High definition)
IO	Giriş / Çıkış (Input / Output)
TTL	Transistor – transistor mantığı (Transistor – transistor logic)
bps	Saniyede bir bit (Bits per second)
Mbps	Saniyede bir milyon bit (Mega bits per second)
PID	Oransal – integral – türevsel denetleyici (Proportional, Integral, Derivative)
KLT	Kanade – Lucas – Tomasi özellik izleyici
SIFT	Ölçek bağımsız özellik dönüştürücü (Scale invariant feature transform)
MGE	Değiştirilmiş açgözlü değiştirme (Modified greedy exchange)
GOA	Açgözlü en iyi atama (Greedy optimal assignment)
JPDAF	Birleşik olasılıksal veri ilişkilendirme süzgeci (Joint probability data association filtering)
PMHT	Olasılıksal çok hipotezli izleme (Probabilistic multiple hypothesis tracking)
SVM	Destek vektör makinesi (Support vector machine)
CPU	Merkezi işlem birimi (Central Processing Unit)
SURF	Hızlandırılmış dayanıklı öznitelikler (Speeded up robust features)

1. GİRİŞ

Günümüzde görüntü işleme, teknolojinin hızlı gelişimi ve görüntülerin kısa sürede işlenebilir hale gelmesi ile hayatımızın daha geniş alanlarında yer bulmaya başlamıştır. Görüntü iyileştirme, görüntü sıkıştırma, tıbbi tanı, uydu ile uzaktan algılama, endüstriyel ürün denetimi, retina, yüz ve parmak izi tanıma, plaka okuma ve nesne takibi gibi birbirinden farklı birçok alanda yaygın olarak kullanılmaktadır [1].

Birçok güvenlik uygulamasında kameraların ilgilenilen objeyi otomatik olarak algılayarak izlemesi istenir. Şüpheli şahısların teşhisi ve izlenmesi, trafikteki araç sayısının sayılması, araçların akış hızlarının ölçülerek trafik yoğunluğunun otomatik olarak algılanması, aşırı hız, kaza benzeri durumların algılanarak bildirilmesi, insansız gözetim, insansız araç navigasyonu gibi uygulamalarda nesne izleme kullanılmaktadır [1].

Nesne izleme ile ilgili yapılan araştırmalar incelendiğinde; Cahn von Seelen ve Bajcsy, [2] yaptıkları çalışmada uyarlanabilir korelasyon metodu kullanarak ölçeği değişen nesnelere izlemişlerdir. Söz konusu çalışmada, birbirini izleyen görüntüler arasındaki ölçek değişimlerine bakılarak elde edilen parametrelere göre, aranan nesne şablonu üzerinde değişiklikler yapılmıştır. Liu vd., [3] yaptıkları çalışmada Kalman kestirimi ve çoklu özellik birleşimi temelli CAMSHIFT (Sürekli uyarlamalı ortalama değer kayması – Continuously Adaptive Mean-Shift) [4] kullanarak gerçek zamanlı nesne izleme yapmışlardır. Hidayatullah ve Konik, [5] yaptıkları çalışmada CAMSHIFT ile çok renkli nesnelere izlemişlerdir. Bu çalışmada, nesne üzerindeki dominant renkler farklı birer nesneymiş gibi kabul edilerek, ayrı ayrı izleme yapılmıştır. Exner vd., [6] yaptıkları çalışmada birden fazla histogramın toplamını kullanarak kompleks nesnelere CAMSHIFT ile izlemişlerdir. Bu işlem GPU (Grafik İşleme Ünitesi – Graphics Processing Unit) üzerinde yürütülmüş ve sistemin gerçek zamanlı olması sağlanmıştır. Ellis vd., [7] yaptıkları çalışmada görüntü korelasyonunun kalite testi, görüntü tanıma, görüntü arama ve görüntü izlemede kullanılabilir özelliğini kullanmışlardır. Görüntüye Fourier dönüşümü uygulayarak frekans uzayına geçmiş ve görüntü korelasyonu ile nesne izleme yapmışlardır. Jog ve Halbe, [8] yaptıkları çalışmada video akışı üzerinde kullanıcı tarafından çizilen bir çizgiyi geçen nesne olduğunda nesneyi izlemeye başlayan bir güvenlik

uygulamasını geliřtirmişlerdir. Bu uygulamada, nesne izleme işleminin CAMSHIFT algoritması kullanılarak yapılmıştır. Rami vd., [9] yaptıkları çalışmada video akışındaki her bir görüntünün ve izlenecek nesnenin LBP (Yerel ikili örüntü - Local Binary Pattern) histogram örüntüsünü çıkarmış ve bu örüntüler arasındaki uzaklığa bakarak histogramlar arasındaki benzerliği bulmuşlardır. Bu şekilde, nesneyi görüntü akışı içerisinde izlemeyi başarmışlardır. Nebhay ve Pflugfelder, [10] yaptıkları çalışmada optik akış kestirimi yaparak önemli noktaların birbirini takip eden çerçevelerde izlemesini yapmışlardır. İzleme işleminin başarılı olduğu durumlarda sisteme yeni önemli noktalar eklenerek, sistemin nesnenin değişimlerini öğrenmesi sağlanmıştır.

Bu tez çalışmasında, birden fazla hareketli cismin serbestçe hareket ettiği üç boyutlu bir ortamda, bu cisimlerden lazer ışını ile anlık olarak işaretlenmiş olanına kilitlenen bir hedef izleme sistemi geliştirilmiştir. Nesne izleme, x ve y düzlemlerindeki bakış açısı iki servo motor ile ayarlanabilen taşıyıcı sisteme yerleştirilmiş bir video kameranın, hedef olarak seçilen cismin görüntü alanının merkezinde kalacak şekilde hareket ettirilmesiyle sağlanmıştır.

Bir video kaynağı üzerinde izleme işlemi yapılırken, görme alanı içerisinde izlenmesi amaçlanan belirli nesne veya nesnelere vardır. Bu nesnelere, her bir görüntü çerçevesi için arka plandan izole edilmesi gerekir. İzolasyon işlemi tamamlandıktan sonra nesnenin hareketi incelenebilir. Bu işlemin, tanıma ve modelleme olmak üzere iki ana bileşeni vardır [11].

Tanıma, bir çerçeve içerisinde ilgili nesnenin video akışındaki takip eden çerçevelerde de bulunmasıdır. Momentler ve renk histogramları, aranan nesnelere tanınmasında kullanılan araçlardır. Modelleme, izleme algoritmalarının nesnenin gerçek pozisyonu ile ilgili sağladığı ölçümlerdeki gürültüyü tespit etmeye yarar ve gürültüye karşı önlem olarak izlenen nesnenin gerçek yerinin belirlenmesini kolaylaştırır [11].

Yapılan çalışmada, izlenecek nesnenin seçilmesi ve izolasyon işlemlerinin yapılabilmesi için nesnenin ilk olarak lazer işaretleyici tarafından işaretlenerek sisteme gösterilmesi gerekir. İşaretleme anlık olarak yapılır, lazer ışını yok olduğunda sistem ışının çarptığı son nesneyi hedef olarak seçer. Hedef olarak

seçilen nesne, renk özelliklerine bakılarak arka plandan izole edilir ve nesnenin içinde kalacak en büyük kare çerçeve hesaplanır. Bu çerçeve içerisindeki renk, parlaklık ve ışığın doyum miktarına göre izleme işleminde kullanılacak değerler elde edilir.

Lazer işaretleyici kullanılarak hedef belirleme, lazer ışınının farklı aydınlatma koşulları altında, farklı renkli nesnelere üzerinde sistem tarafından ayırt edilebilmesine bağlıdır. Bu gibi durumlarda lazer ışınının saptanması konusunda literatür çalışmaları mevcuttur; Meško ve Toth, [12] yaptıkları çalışmada gerçek zamanlı görüntü işleme teknikleri kullanımı ve kamera ayarlarının yüksek parlaklıktaki lazere göre ayarlanması ile kırmızı ve yeşil lazer ışınını projeksiyondan yansıyan görüntü üzerinde tespit etmişlerdir. Le vd., [13] yaptıkları çalışmada, lazer ışınının karakteristik özelliklerini kullanarak ışını görüntünün diğer kısımlarından ayırmışlardır. HSI (Renk, doygunluk ve yoğunluk renk uzayı - Hue, Saturation, Intensity) renk uzayı kullanılarak yapılan çalışmalarının RGB (Kırmızı, yeşil ve mavi renk uzayı - Red, Green, Blue) uzayı kullanılarak yapılan çalışmalara göre daha başarılı olduğunu ve %82 oranında doğru tespit yaptığını savunmuşlardır. Krstinić vd., [14] yaptıkları çalışmada, değiştirilmiş dairesel Hough dönüşümü ve Lucas-Kanade [15] hareket analizi yöntemlerini kullanarak lazer ışınını tespit etmişlerdir.

Bilgisayarla görme alanındaki ilk çalışmalar sabit bir noktadan çekilmiş görüntüler üzerinde yapılmıştır. Bu tip görüntüleri sağlayan sabit kameraların görüş alanları, kameranın kurulumu sırasında belirlenir ve daha sonradan bu görüş alanının değiştirilmesi için, kamara açısının veya lensinin dışarıdan destek alarak değiştirilmesi gerekir [16]. Bu zorluğun ortadan kalkması amacıyla hareketli kameralar üzerinde çalışmalara başlanmıştır ve Kim, [17] yaptığı çalışmada X ve Y düzlemlerinde dönüş hareketini yapan ve bu işlem için dışarıdan herhangi bir desteğe ihtiyacı olmayan, pan/tilt kamera tasarımının patentini almıştır. Bilgisayarla görme alanında yapılan araştırmalar, nesle izlemede hareketli kameraların kullanılmasıyla elde edilen görüş alanının, sabit kameraların kısıtlı görüş alanına göre çok üstün olduğunu ortaya çıkarmıştır. Bununla birlikte, bir hedefin fiziksel olarak izlenmesi, o hedefin devinim bulanıklığını (Motion blur) azaltırken, çözünürlüğünü artırır [18]. Bu özellik, sınıflandırma gibi yüksek seviye işlemlerin daha başarılı bir şekilde uygulanmasını sağlar. Bilgisayarla görme alanında aktif

görüntüleme (active vision) adı verilen, sistemin performansını arttırmak amacıyla bakış noktasının hareketli bir kamera ile değiştirilmesi [19], temeline dayanan çalışmalar 1980'lerin sonlarında başlamış ve Aloimonos vd., [20] bu alandaki ilk çalışmaları yapmışlardır. Hareketli kameraların nesne izlemede kullanılmaları ile ilgili çalışmalar özetlenecek olursa; Lim vd., [21] yaptıkları çalışmada otomatik bir güvenlik sisteminde yer alan kameralara pan/tilt özelliği kazandırarak izledikleri objenin görüş alanında kalmasını sağlamışlardır. Kurihara vd., [22] yaptıkları çalışmada optik hareket yakalama sisteminde pan/tilt kamera kullanarak çekim alanını dinamik olarak genişletmişlerdir. Stolle ve Rysdyk, [23] yaptıkları çalışmada insansız otonom bir hava aracının önüne yerleştirdikleri pan/tilt kamera ile araç hareket halindeyken, gözlemlendiği cismin görüş alanı içinde kalmasını sağlamışlardır. Bu şekilde gözleme işlemi aracın hareketlerinden bağımsız hale getirilmiştir.

Bu tez çalışmasında sistemin izleyeceği nesnenin belirlenmesi için gerekli olan lazer işaretlemeye; lazer ışının düştüğü noktanın ortam aydınlatmasından etkilenmeden bulunabilmesi için, kamera pozlandırma süresi ve kazancı değiştirilerek kameranın algıladığı ışık seviyesi belli bir değerin altına çekilir. Bu işlem, görüntünün ortalama aydınlığının, ortamın aydınlığından bağımsız olarak belirli bir değere düşürülmesini sağlar. Yani ortam ne kadar aydınlık olursa olsun, görüntü belirli bir değere karartılmış olur. Karartılmış olan görüntüde parlak lazer ışını belirlenir. Lazer ışınının belirlendiği görüntüde morfolojik işlemler yardımı ile gürültü azaltılır ve belirlenen lazer ışını ön plana çıkarılır. Aynı görüntüye eşikleme ve mantık işlemleri uygulanarak, belirlenen noktanın herhangi bir parlamadan ziyade, belirlenmesi amaçlanan lazer ışını olduğuna karar verilir.

İzlenecek olan nesnenin belirlenmesinden sonra, nesnenin renk histogramı çıkarılarak CAMSHIFT ile izleme yapılır. İzleme sırasında benzer renklerden kaynaklanabilecek gürültü, birbirlerine bağlı elemanların en büyüğünü bulma teknikleri ile ortadan kaldırılarak [24], en büyük nesne olan hedef ortaya çıkarılır. Nesnenin başka cisimler tarafından engellenmesi durumunda kalman süzgeci [25] kestirimleri ile hedefin izlenmesine devam edilir. Bunun yanında, hareketli cismin hareketlerinin yumuşatılması ve sistemin keskin hareketler yapmasını engelleme amacıyla kalman süzgecinin düzeltmeleri kullanılır. İzleme işlemi, görüntü işleme

yazılımı ve motor sürücü kart arasındaki seri haberleşme ile sağlanan x ve y düzlemlerindeki hareket ile daha geniş bir görüş alanı içerisinde yapılmıştır. Servo motorlar aracılığıyla sağlanan bu hareket ayrıca, nesnenin hareketinin izleme başarısına olan etkisini azaltmıştır.

Çalışmanın ikinci bölümünde, sayısal görüntü işlemenin temelleri hakkında bilgi verilmiş ve çalışmada kullanılan yöntemlerin anlatımı yapılmıştır.

Üçüncü bölümde, nesne izlemenin temel adımları ve nesne izleme yöntemleri üzerinde durulmuştur.

Dördüncü bölümde, çalışmanın geliştirilmesinde kullanılan yazılım ve donanımlar incelenerek özellikleri belirtilmiştir.

Beşinci bölümde sistemin çalışması ve genel özellikleri, lazer işaretlemede kullanılan algoritma, nesne izlemede kullanılan algoritma ve sistemin hareketini sağlayan algoritma da dahil olmak üzere adım adım anlatıldıktan sonra, sistemin başarılı çalışması için gerekli ön koşullar üzerinde durulmuş ve sistem farklı durumlar altında test edilerek bulgular üzerinde tartışılmıştır.

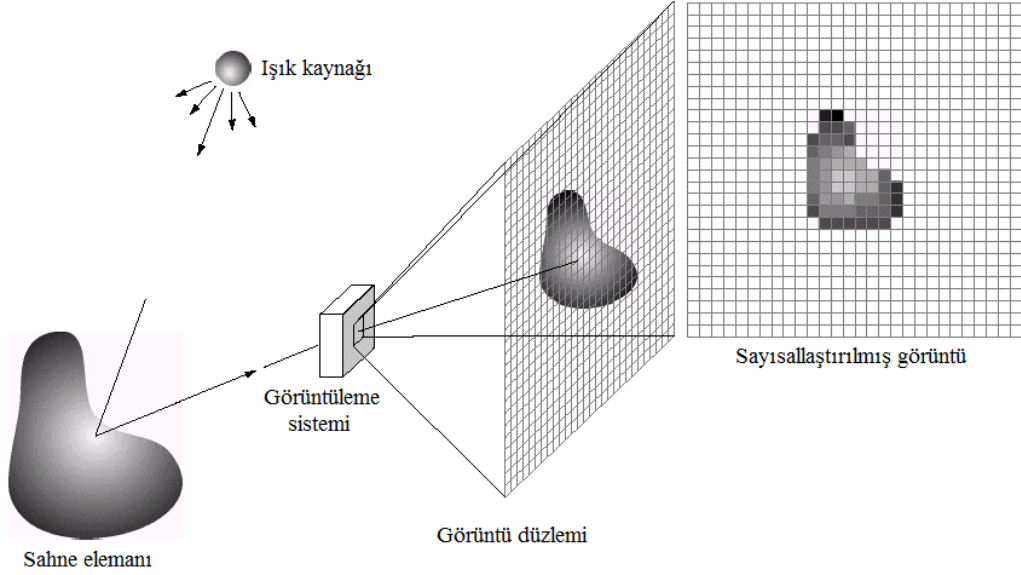
Son bölümde ise, tasarlanan sistemin ve kullanılan yöntemlerin başarısı ve performansı, beşinci bölümde yapılan testlerden elde edilen sonuçlarla irdelenmiş ve gelecek çalışma planlarından bahsedilmiştir.

Bu tez kapsamında yapılan çalışmalar, Microsoft Visual Studio 2013 tümleşik geliştirme ortamında C++ programlama dili ve OpenCV 2.4.9 görüntü işleme kütüphanesi kullanılarak, 8 çekirdekli Intel i7 işlemci, 8 GB RAM ve Nvidia GTX960M ekran kartından oluşan yüksek performanslı bir bilgisayar üzerinde tamamlanmıştır. Görüntü, Logitech® HD Pro Webcam C920 ile alınmıştır. Sistem, Arduino tabanlı PhantomX AX-18A Turret kullanılarak hareketli hale getirilmiş ve hareket algoritması Arduino Software (IDE) 1.6.5 derleyicisi kullanılarak yazılmıştır.

2. SAYISAL GÖRÜNTÜ İŞLEME

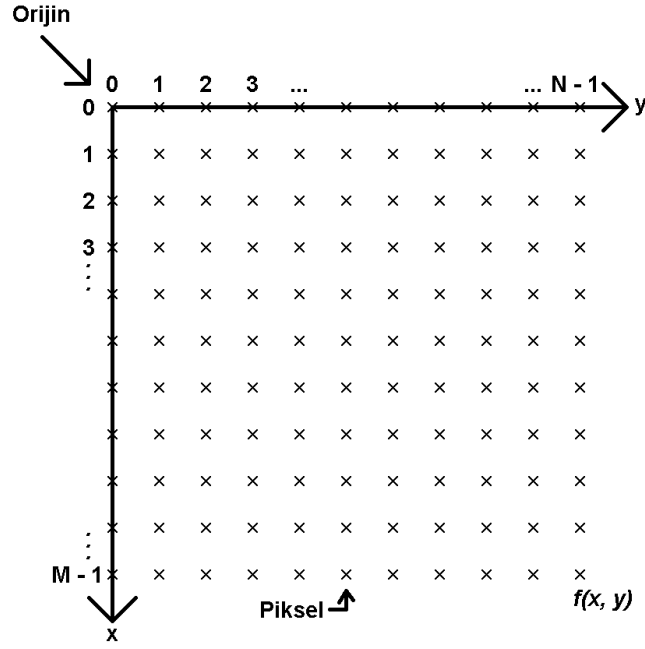
2.1. Sayısal Görüntü

Sayısal görüntü, nesnelere tarafından yansıtılan ışık enerjisinin bir algılayıcı tarafından öngörülen elektromanyetik aralıkta algılanarak sayısal sinyal haline dönüştürülmesiyle oluşur. Matematiksel ifade ile, x ve y uzamsal koordinatlar olmak üzere, $f(x,y)$ ile ifade edilen ve her bir (x,y) koordinatındaki büyüklüğü, görüntünün o noktadaki aydınlanması veya gri seviyesine eşit olan iki boyutlu bir fonksiyona görüntü denir. x , y koordinatlarının ve f fonksiyonunun değerinin, sonlu ve kesikli olduğu görüntülere sayısal görüntü denir (Şekil 2.1) [26].



Şekil 2.1 Sayısal Görüntü ([26]'dan Türkçeleştirilerek)

Sayısal bir görüntü piksellerin m satır n sütun bir matriste saklanması ile oluşur (Şekil 2.2). Gri tonlu bir görüntüde, her bir piksel o pikselin koordinatlarına karşılık gelen gri ton değerini saklar. Bu değer aralıkları genellikle $\{0,1,2,\dots,255\}$ şeklinde ifade edilir. Bunun anlamı, gri tonlu bir görüntünün 256 farklı gri ton değeri bulundurabileceğidir. Bu ifade şeklinde, sıfır değeri siyah renge, 255 değeri ise beyaz renge karşılık gelmektedir. Bu değerler arasında gri tonlar oluşur.

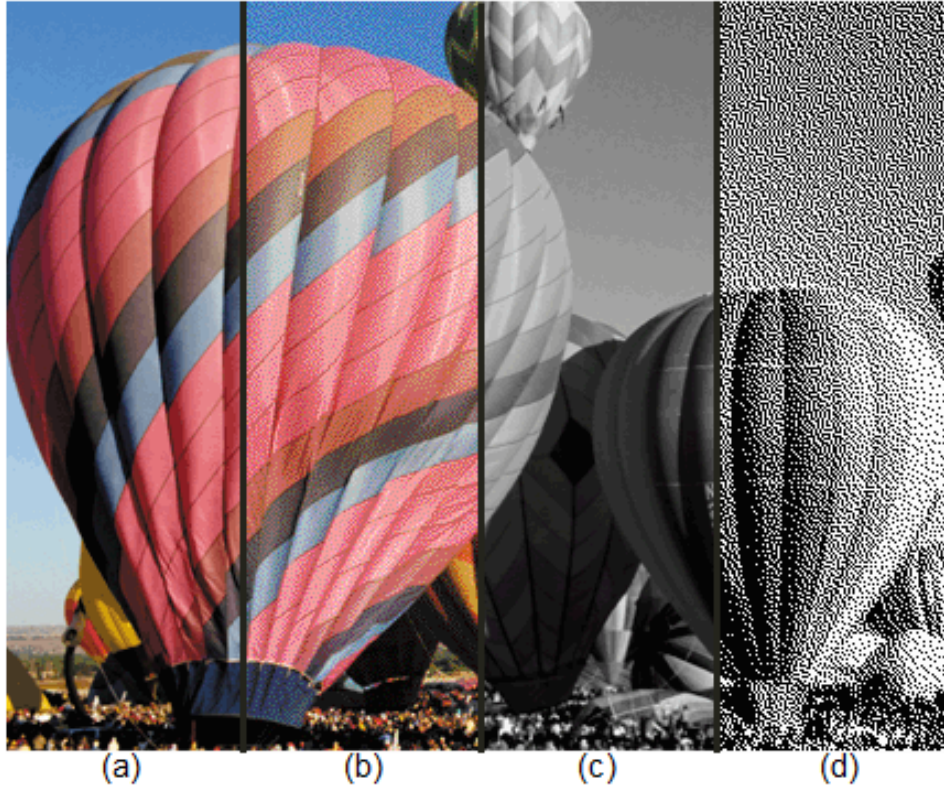


Şekil 2.2 Piksellerin Matris Gösterimi

Sayısal bir görüntüde her bir pikselde gösterilebilecek renk tonu sayısı o pikseli oluşturan bit sayısı ile doğru orantılıdır. Bit sayısının artması daha fazla ton ve daha fazla detay anlamına gelir. Çizelge 2.1, farklı bit sayılarında oluşturulabilecek ton sayılarını ve bunlarının ikili olarak gösterimini, Şekil 2.3 ise bit sayısının detaya olan etkisini göstermektedir. Şekil 2.3 (a), renkli görüntünün 24-bit (16 milyon renk) ile gösterimi, Şekil 2.3 (b) renkli görüntünün 8-bit (256 renk) ile gösterimi, Şekil 2.3 (c) gri seviye görüntünün 8-bit (256) ton ile gösterimi ve Şekil 2.3 (d) gri seviye görüntünün 1-bit (siyah/beyaz) gösterimidir.

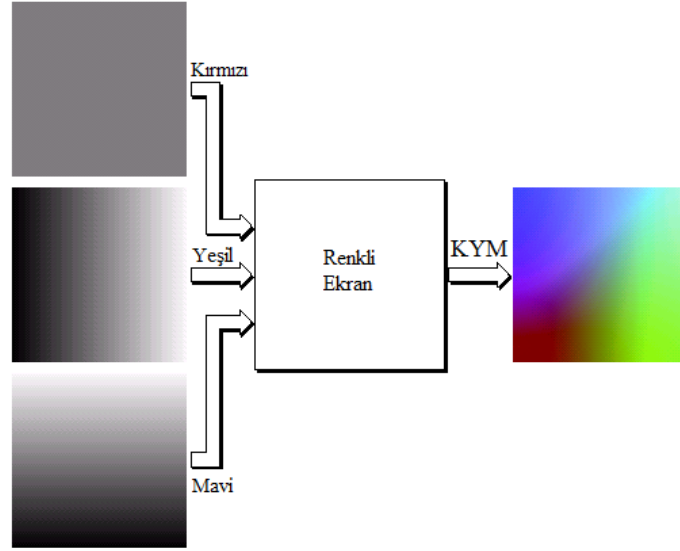
Çizelge 2.1 Bit Sayısı, Ton Sayısı İlişkisi

Bit Sayısı	Ton Sayısı	Örnekler
1	2	0, 1
2	4	00, 01, 10, 11
4	16	0000, 0101, 1111
8	256	00110011, 01010101
16	65,536	1010101010101010
24	16,777,216	111100001111000011110000



Şekil 2.3 Bit Sayısı, Detay İlişkisi. (a) 24-bit renkli, (b) 8-bit renkli, (c) 8-bit gri seviye, (d) 1-bit siyah/beyaz [27]

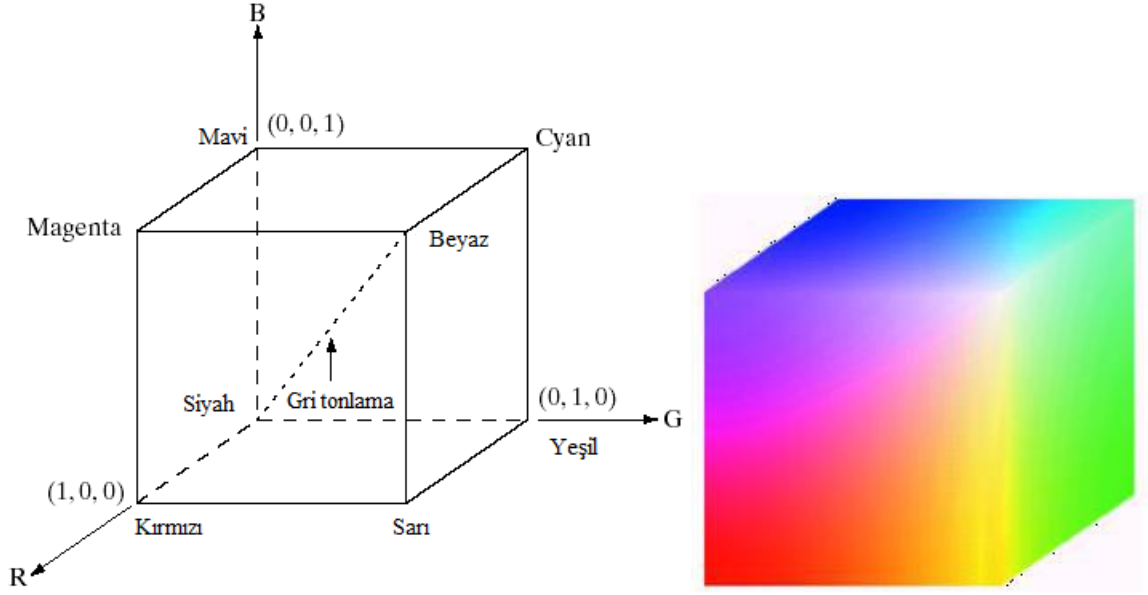
Renkli görüntüler elektromanyetik spektrumda, 0,4-0,5 μm dalga boylu mavi rengin, 0,5-0,6 μm dalga boylu yeşil rengin ve 0,6-0,7 μm dalga boylu kırmızı rengin, kırmızı-yeşil-mavi (RGB) kodlanmasıyla oluşturulur. Renkli resmin piksel alanının içinde kalan, kırmızı, yeşil ve mavi temel renk tonları ayrı renk kanallarına bölünür ve her renk için ayrı birer gri seviye hesaplanır. Yani RGB ile kodlanmış ve sıkıştırılmamış renkli bir resim aynı boydaki üç gri resim kadar yer kaplar (Şekil 2.4) [1].



Şekil 2.4 Renkli Görüntünün Oluşumu ([26]'dan Türkçeleştirilerek)

2.2. RGB Renk Uzayı

RGB renk uzayında, her renk birincil spektral bileşenleri olan kırmızı, yeşil ve mavi renklerin birleşimi ile gösterilir. RGB renk uzayı, kartezyen koordinat sistemi tabanlı bir renk uzayıdır. İlgili renk altuzay küpü Şekil 2.5' de gösterilmiştir. Burada RGB birincil değerleri küpün üç köşesinde; ikincil renkler cyan (camgöbeği), magenta (macenta) ve yellow (sarı) diğer köşelerde; siyah orijinde; ve beyaz orijinden en uzak köşededir. Bu renk uzayında, gri seviye (RGB değerleri için eşit olan) siyah ile beyaz arasındaki çizgi boyunca artar. Diğer renkler küpün üzerinde veya içerisinde olmak üzere, orijinden çizilen vektörler ile ifade edilir. Daha kolay anlaşılması için, Şekil 2.5' de gösterilen küp üzerindeki renklerin değerleri normalize edilmiş ve R, G ve B değerleri [0,1] aralığında kabul edilmiştir [26].



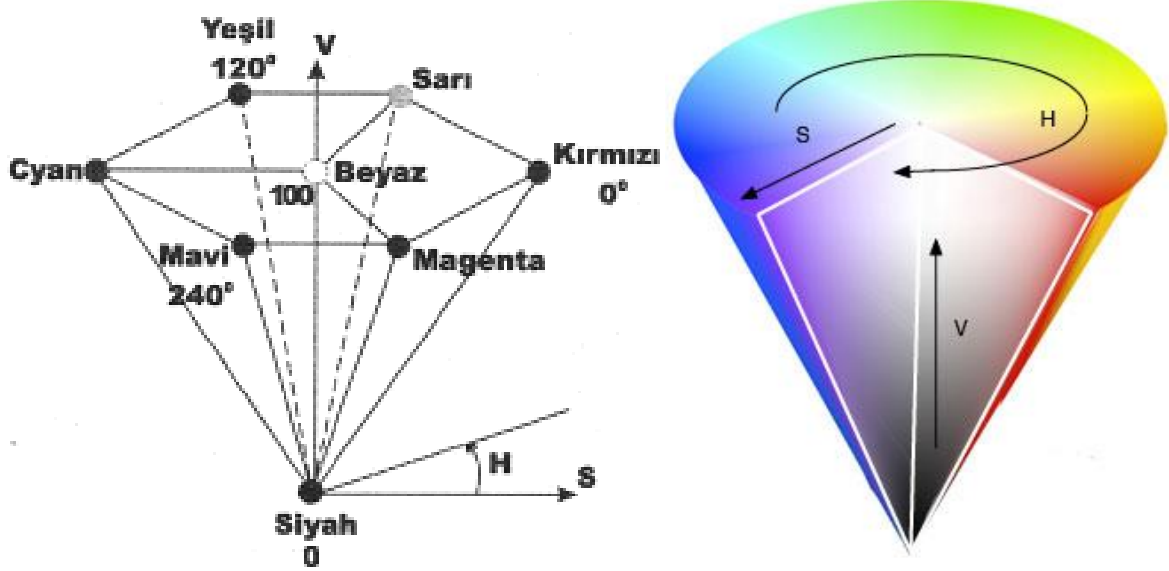
Şekil 2.5 RGB Renk Küpü ([26]'dan Türkçeleştirilerek)

RGB renk uzayında görüntüler, her biri ana renkler olmak üzere 3 görüntü bileşeninden oluşurlar. RGB renk uzayında, her bir pikseli temsil eden bitlerin sayısı piksel derinliği (pixel depth) olarak adlandırılır. Kırmızı, yeşil ve mavi görüntülerin her birinin 8-bit olduğu bir RGB görüntüyü ele aldığımızda, renkli görüntünün piksel derinliği üç görüntünün birleşimi olan 24-bit olur. Tam renkli (full-color) olarak adlandırılan 24-bitlik bir RGB renkli görüntüde, $(2^8)^3 = 16,777,216$ adet renk bulunur [26].

2.3. HSV Renk Uzayı

HSV renk uzayı özellikle görüntü işlemede kullanılan bir renk uzayıdır. HSV renk uzayında görüntü; renk (hue), doygunluk (saturation) ve parlaklık (value) bileşenlerinin birleşiminden oluşur [28]. Renk değeri, bir rengin diğer renklerden, rengin çeşidine ve tonuna göre ayrılmasını sağlayan niteliklidir. Rengün baskın dalga uzunluğunu belirleyen açısal bir değerdir. 0-360° arası değerler alır ve her değer farklı bir renge karşılık gelir. Doymunluk, bir rengin aynı değerdeki bir renkten ayırım derecesidir. Rengün canlılığını belirler. 0-100% arası değerler alır. Yüksek doymunluk canlı renklere neden olurken, düşük doymunluk rengin gri tonlarına yaklaşmasına neden olur. Doymunluğun 0 olması renk yok anlamına gelirken, 100 olması ise yoğun renk anlamına gelmektedir. Parlaklık değeri, rengin içindeki beyaz oranını belirler ve açık rengi koyu renkten ayırt etmekte kullanılır. 0-100% arası değerler alır. Parlaklığın 0 olması her zaman siyah renk anlamına gelir, 100 olması

ise doygunluk değerine bağlı olarak beyaz veya daha yoğun bir renk anlamına gelir. HSV renk uzayının geometrik şekli altıgen koni Şekil 2.6'da gösterilmiştir. RGB renk küpünün diyagonal çizgi boyunca iki boyutlu projeksiyonudur. Birincil renkler (kırmızı, yeşil, mavi) ve ikincil renkler (cyan, magenta, sarı) altıgenin köşelerinde yer alır [29].



Şekil 2.6 HSV Renk Konisi [29]

Şekil 2.6'da gösterilen altıgen koninin en üst kısmı $V = 100$ 'e veya maksimum yoğunluğa karşılık gelir. Koninin en altındaki nokta siyah rengi temsil eder ve bu noktada $V = 0$ 'dır. Kırmızı 0° olmak üzere tamamlayıcı renkler H ile ölçülen ve dikey eksene (V) göre alınan açıya göre 180° karşılıklıdır. S ile gösterilen, değeri dikey eksenden (V) koninin kenarlarına doğru 0 ile 100 arasında değişen değere rengin parlaklık oranı denir. $S = 0$, $V = 100$ noktası beyazdır. $S = 0$ için V'nin ara değerleri gri tonlarıdır. $S = 0$ ise H değeri önemsizdir. $V = 100$ ve $S = 100$ durumu, rengi H tarafından belirlenen saf pigment olarak adlandırılır. V değeri sabitken beyaz eklemek S değerini azaltmaya, S değeri sabitken siyah eklemek V değerini azaltmaya karşılık gelir. Tonlar ise hem S hem de V'nin azaltılmasıyla elde edilir [30].

2.4. RGB ve HSV Renk Uzaylarının Avantaj ve Dezavantajları

RGB renk uzayı, kamera ve ekranların çalışma mantığı ile aynıdır. Bu sebeple, bu cihazlar ile etkileşime girilirken kullanılması kolaylık sağlar. HSV renk uzayı, insan

görüşüne yakındır. Konik şekilde olması, renkler arasında daha rahat geçiş yapılmasını sağlar.

RGB renk uzayında her kanal renk ve aydınlanma bilgisi taşır. Bu sebeple aydınlanmada olan değişimler RGB değerlerinin tümünü ve buna bağlı olarak rengi etkiler. HSV kanalında ise, H ve S değerleri yalnızca renk bilgisini, V değeri ise yalnızca aydınlanma bilgisini taşır. Bu sebeple, HSV renk uzayı kullanılan görüntü işleme uygulamalarında, yalnızca H ve S kanalları dikkate alınarak aydınlanma değişimlerine karşı daha dayanıklı uygulamalar geliştirilebilir. Bu özellik HSV renk uzayının, bilgisayarla görme ve bölütleme işlemleri için gerekli görüntü analizinde tercih edilmesini sağlar [31].

2.5. RGB – HSV Renk Uzayı Dönüşümleri

2.5.1. RGB'den HSV'ye dönüşüm algoritması

RGB renk uzayından HSV renk uzayına geçişte kullanılan algoritma aşağıdaki adımlardan oluşmaktadır [32]. Algoritmada kullanılan R, G ve B değerleri [0, 1] aralığındadır. Değerlerin [0, 1] aralığına indirgenmesi her birinin 255 ile bölünmesi ile gerçekleştirilir.

1. $V = \max(R, G, B);$
2. $X = \min(R, G, B);$
3. $S = \frac{V-X}{V}$
4. $r = \frac{V-R}{V-X}, g = \frac{V-G}{V-X}, b = \frac{V-B}{V-X}$
5. *Eğer $R = V$ ise $H = (Eğer $G = X$ ise $5 + b$ değilse $1 - g)$;
eğer $G = V$ ise $H = (Eğer $B = X$ ise $1 + r$ değilse $3 - b)$;
diğer durumlarda $H = (Eğer $R = X$ ise $3 + g$ değilse $5 - r)$;$$$*
6. $H = \frac{H}{6}$

Bu algoritma matematiksel olarak ifade edilecek olursa [33];

$$H = \begin{cases} 60 \left(\frac{G - B}{\delta} \right), & \text{Eğer } MAX = R \text{ ise} \\ 60 \left(\frac{B - R}{\delta} + 2 \right), & \text{Eğer } MAX = G \text{ ise} \\ 60 \left(\frac{R - G}{\delta} + 4 \right), & \text{Eğer } MAX = B \text{ ise} \\ \text{Tanımsız}, & \text{Eğer } MAX = 0 \text{ ise} \end{cases} \quad (2.1)$$

$$S = \begin{cases} \frac{\delta}{MAX}, & \text{Eğer } MAX \neq 0 \text{ ise} \\ 0, & \text{Eğer } MAX = 0 \text{ ise} \end{cases}$$

$$V = MAX$$

burada $\delta = (MAX - MIN)$, $MAX = \max(R, G, B)$ ve $MIN = \min(R, G, B)$ 'dir. H değerinin $[0, 360]$ aralığında olması için,

$$H = H + 360, \text{ Eğer } H < 0 \text{ ise}$$

işlemi yapılmalıdır.

2.5.2. HSV'den RGB'ye dönüşüm algoritması

HSV renk uzayından RGB renk uzayına geçişte kullanılan algoritma aşağıdaki adımlardan oluşmaktadır [32]. Algoritmada kullanılan H, S, V değerleri ve işlem sonrasında elde edilen R, G, B değerleri $[0, 1]$ aralığındadır.

1. $H = 6 * H$;
2. $I = \text{floor}(H)$; $F = H - I$;
3. $M = V * (1 - S)$; $N = V * (1 - S * F)$; $K = V * (1 - S * (1 - F))$;
4. *Switch(I)*
 - case 0: $(R, G, B) = (V, K, M)$;
 - case 1: $(R, G, B) = (N, V, M)$;
 - case 2: $(R, G, B) = (M, V, K)$;
 - case 3: $(R, G, B) = (M, N, V)$;
 - case 4: $(R, G, B) = (K, M, V)$;
 - case 5: $(R, G, B) = (V, M, N)$;

burada floor(H) işlemi, H ondalık sayısının tabana yuvarlanarak tam sayı haline getirilmesidir. floor işleminin sonucu H değerinden küçük veya eşit bir tamsayı olur. “switch case” deęiminde l'nın deęerine göre caseler (durumlar) ierisinden yalnızca biri işlenir.

Bu algoritma matematiksel olarak ifade edilecek olursa;

$$(R, G, B) = \begin{cases} (V, K, M), & \text{Eğer } I = 0 \text{ ise} \\ (N, V, M), & \text{Eğer } I = 1 \text{ ise} \\ (M, V, K), & \text{Eğer } I = 2 \text{ ise} \\ (M, N, V), & \text{Eğer } I = 3 \text{ ise} \\ (K, M, V), & \text{Eğer } I = 4 \text{ ise} \\ (V, M, N), & \text{Eğer } I = 5 \text{ ise} \end{cases} \quad (2.2)$$

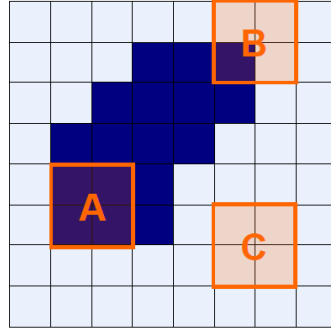
burada $H = 6 * H$, $I = (H) \bmod 1$, $M = V * (1 - S)$, $N = V * (1 - S(H - I))$, ve $K = V * (1 - S(1 - (H - I)))$ ' dir.

2.6. Morfolojik İşlemler

Morfoloji, biyolojide canlıların şekil ve yapıları ile ilgilenen bilime verilen isimdir. Matematiksel morfoloji, temel küme işlemlerine dayanan, görüntüdeki sınırlar (borders) ve iskelet (skeleton) gibi yapıların tanımlanması ve çıkartılması, gürültü giderimi, bölütleme gibi uygulamalar için gerekli bir araçtır. Görüntü işlemede genellikle, süzgeçleme, inceltme (thinning), budama (prunning) gibi uygulamalar için ön veya son işlem olarak kullanılır. Morfolojik işlemler, gri tonlu veya ikili (binary) görüntüler üzerinde uygulanabilirler.

Morfolojik operatörlerin iki girişı vardır; bunlardan birincisi üzerinde morfolojik işlem yapılacak olan görüntü, ikincisi ise morfolojik işlemin şeklini belirleyen yapı elemanıdır (structure element). Morfolojik işlemler, yapı elemanının, üzerinde işlem yapılacak olan görüntü üzerinde kaydırılması sonucu ortaya çıkan miss (kaçırmak), hit (isabet etmek) ve fit (denk gelmek) durumlarına göre gerçekleşir. Miss durumu; yapı elemanı üzerinde 1 olan bitlerden hiçbirinin işlem yapılan görüntüde 1 olan bitlerle akışmamasıdır. Hit durumu; yapı elemanı üzerinde 1 olan bitlerden en az bir tanesinin işlem yapılan görüntüde 1 olan bitlerle akışmasıdır. Fit durumu; yapı

elemanı üzerinde 1 olan bitlerden tümünün işlem yapılan görüntüde 1 olan bitlerle çakışmasıdır. Şekil 2.7, A: fit, B: hit, C: miss durumlarını göstermektedir.



Şekil 2.7 Miss, Hit ve Fit (C: Miss, B: Hit, A: Fit) [26]

Matematiksel morfolojinin temel operasyonları; aşındırma (erosion) ve yayma (dilation) olarak ikiye ayrılır. Bu temel operasyonlar kullanılarak, açma (opening), kapama (closing), sınır çıkarma (boundary extraction), bölge doldurma (region filling), birleşik elemanların çıkarımı (extraction of connected components), inceltme (thinning), kalınlaştırma (thickening) ve iskelet çıkarma (skeletonisation) işlemleri yapılabilir.

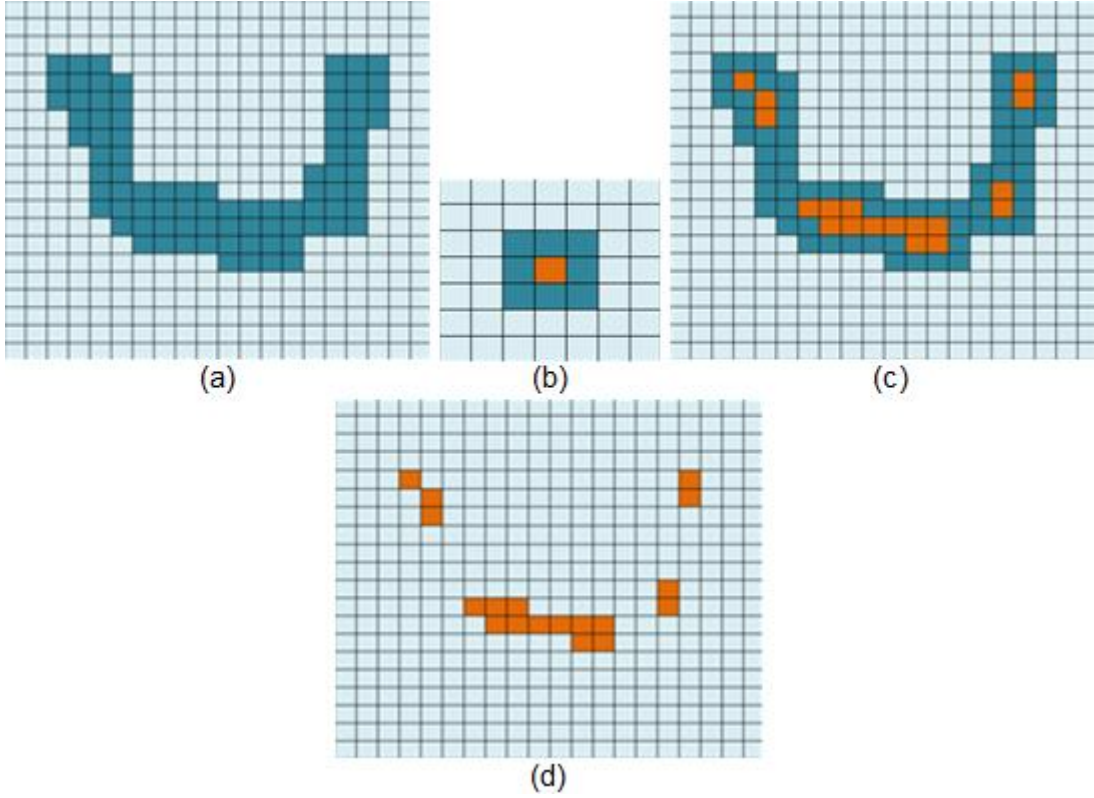
2.6.1. Aşındırma (Erosion)

Matematiksel morfolojinin temel operasyonlarından biridir. Ele alınan sınır bölgelerinin aşındırılmasında kullanılır. Aşındırılacak görüntü ve aşındırma işleminin şeklini belirleyen yapı elemanı olmak üzere iki girişi vardır. f görüntüsünün yapı elemanı s ile olan aşındırması $f \ominus s$ ile gösterilir. s yapısal elemanın orijini (x,y) olacak şekilde yerleştirildiğinde yeni piksel değeri;

$$g(x,y) = \begin{cases} 1, & \text{eğer } s, f \text{ görüntüsüne denk gelirse} \\ 0, & \text{diğer durumlarda} \end{cases}$$

formülüne göre belirlenir.

Şekil 2.8'de aşındırmaya örnek verilmiştir.



Şekil 2.8 Aşındırma örneği. Orijinal görüntü (a), yapı elemanı(b), aşındırılacak noktalar (c), işlenmiş görüntü (d) [34]

Aşındırma işlemi ile; yapışmış olan nesnel ayrılabilir (Şekil 2.9 (a)) ve çıkıntılar yok edilebilir (Şekil 2.9 (b)). Aşındırma işlemi nesnelere küçülmesine sebep olur.



Şekil 2.9 Yapışmış nesnelere ayrılması(a), çıkıntıların yok edilmesi(b) [26]

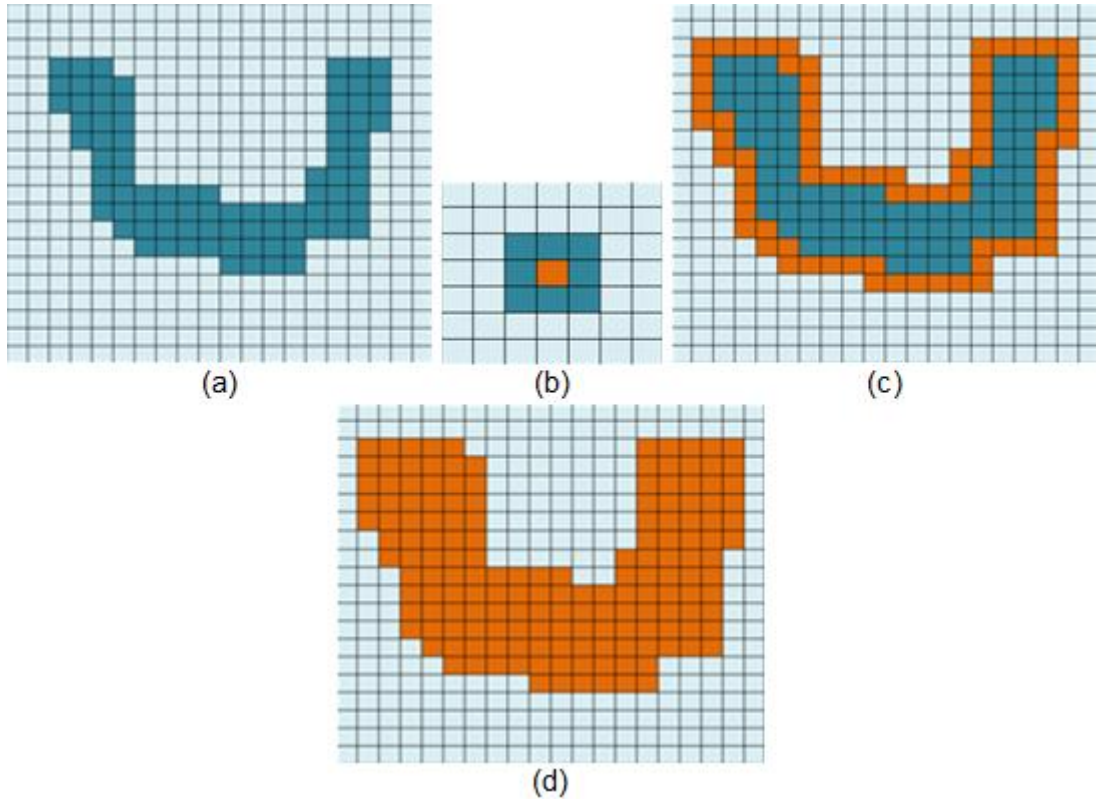
2.6.2. Yayma (Dilation)

Diğer bir temel morfolojik işlemdir. Ele alınan bölgenin sınırlarının genişletilmesinde kullanılır. Yayılacak görüntü ve yayma işleminin şeklini belirleyen yapı elemanı olmak üzere iki girişi vardır. f görüntüsünün yapı elemanı s ile olan yayması $f \oplus s$ ile gösterilir. s yapısal elemanın orijini (x,y) olacak şekilde yerleştirildiğinde yeni piksel değeri;

$$g(x,y) = \begin{cases} 1, & \text{eğer } s, f \text{ görüntüsüne isabet ederse} \\ 0, & \text{diğer durumlarda} \end{cases}$$

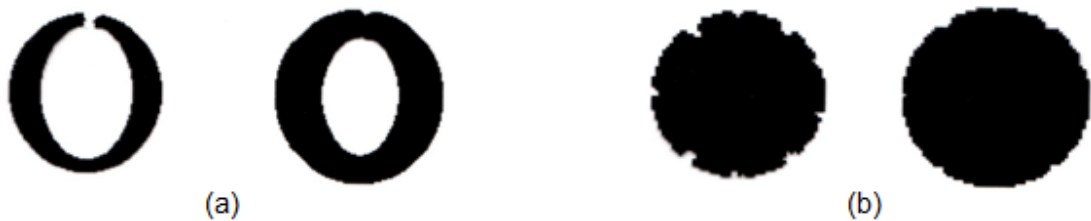
formülüne göre belirlenir.

Şekil 2.10'da yaymaya örnek verilmiştir.



Şekil 2.10 Yayma örneği. Orijinal görüntü (a), yapı elemanı (b), genişletilecek noktalar (c), işlenmiş görüntü (d) [34]

Yayma işlemi ile; kopukluklar onarılabılır (Şekil 2.11 (a)) ve girintiler yok edilebilir (Şekil 2.11 (b)). Yayma işlemi nesnelerin büyümesine sebep olur.

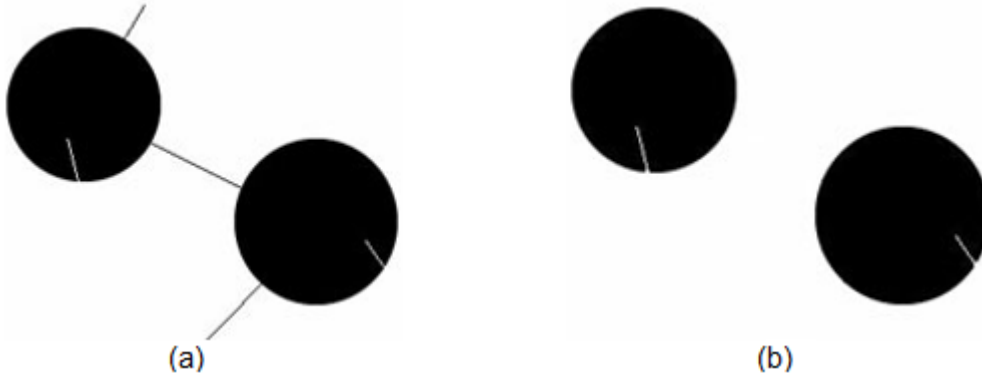


Şekil 2.11 Kopuklukların onarılması(a), girintilerin yok edilmesi(b) [26]

2.6.3. Açma (Opening)

f görüntüsünün yapı elemanı s ile olan açma işlemi $f \circ s$ ile gösterilir. Açma işlemi, aşındırmayı takip eden bir yayma işlemidir (Şekil 2.12).

$$f \circ s = (f \ominus s) \oplus s$$



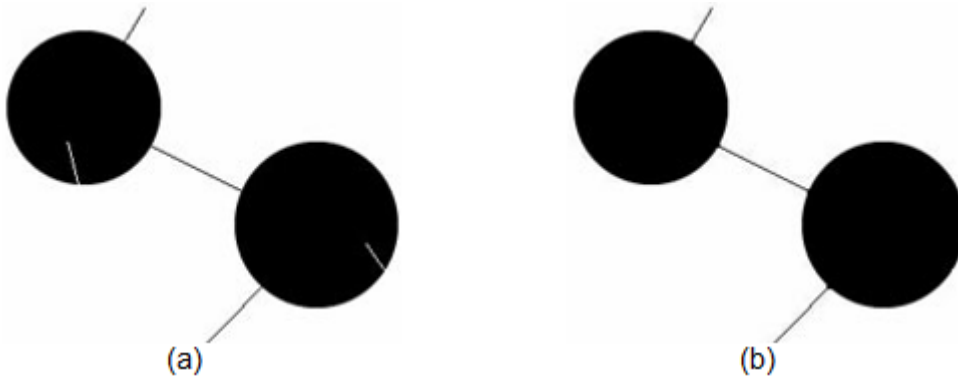
Şekil 2.12 Açma işlemi. Orijinal şekil (a), açma sonrası (b) [35]

Açma işlemi, aşındırma işlemi ile benzer özellikler gösterir. Cisimlerin kenarlarında bulunan parlak piksellerin (gürültü) kaldırılmasında kullanılır. Aşındırma işlemine göre görüntüye daha az zarar verir.

2.6.4. Kapama (Closing)

f görüntüsünün yapı elemanı s ile olan kapama işlemi $f \cdot s$ ile gösterilir. Kapama işlemi, yaymayı takip eden bir aşındırma işlemidir (Şekil 2.13).

$$f \cdot s = (f \oplus s) \ominus s$$



Şekil 2.13 Kapama işlemi. Orijinal şekil (a), kapama sonrası (b) [35]

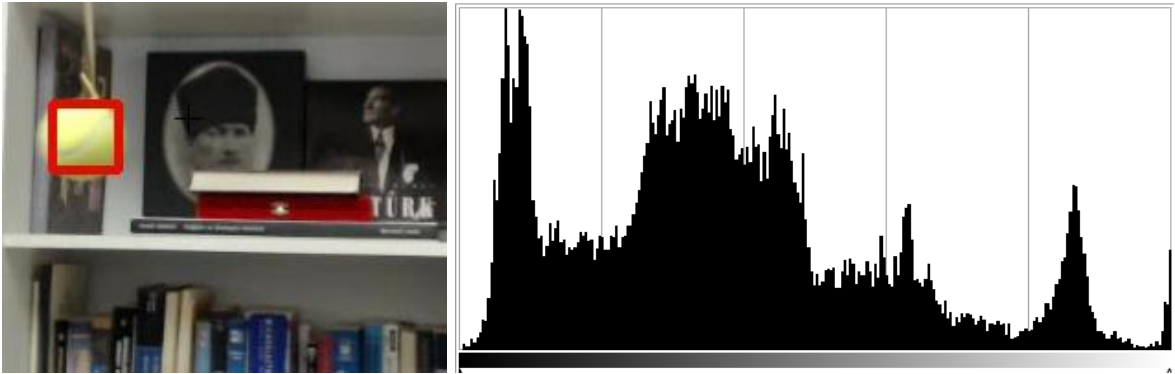
Kapama işlemi, yayma işlemi ile benzer özellikler gösterir. Cisimlerin kenar bölgelerini büyütürken arka planda kalan siyah pikselleri ortadan kaldırır ve boşlukları doldurur. Yayma işlemine göre görüntüye daha az zarar verir.

2.7. Görüntü Histogramı

Görüntü histogramı, sayısal bir görüntü içerisinde her renk değerinden kaç adet olduğunu gösteren grafikdir. Histogram grafiği; yatay ekseninde (x eksen) ışık değeri (intensity) ve dikey ekseninde (y eksen) piksel sayısı olacak şekilde hazırlanan bir grafikdir (Şekil 2.14). Bu grafiğe bakılarak resmin parlaklık durumu ya da tonları hakkında bilgi sahibi olunabilir. Histogram özellikle bölütlemeye (segmentation) olmak üzere, görüntü işleme uygulamalarında yaygın olarak kullanılır. Histogram matematiksel olarak Eşitlik (2.3)'deki gibi gösterilir;

$$h(r_k) = n_k \quad (2.3)$$

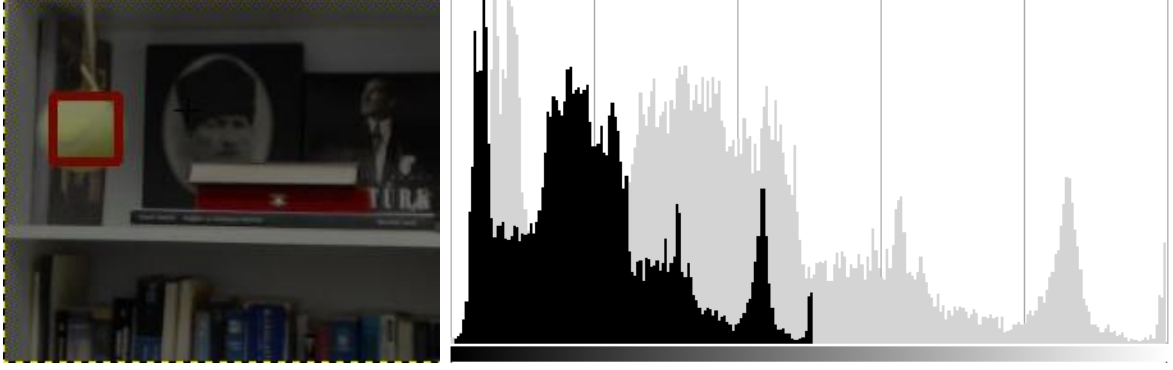
bu gösterimde r_k : k'nıncı parlaklık değerini, n_k : k'nıncı parlaklık değerinin görüntüdeki sayısını ifade eder.



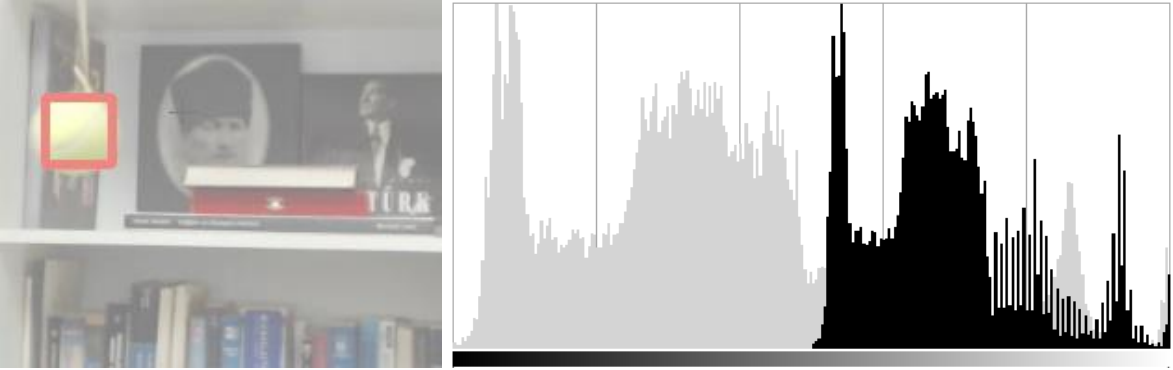
Şekil 2.14 Görüntü Histogramı

Bazı görüntülerin elde edilmesi sırasında, aydınlatmanın yetersiz kalması sebebi ile renk değerlerinin belli bir yerde kümelendiği renk dağılımı bozuklukları oluşabilir. Histogram grafiğinin küçük değerlerde toplandığı görüntüler koyu renkli (Şekil 2.15), histogram grafiğinin yüksek değerlerde toplandığı görüntüler parlak renkli (Şekil 2.16) olurlar. Histogram grafiğinin ortada toplandığı (düşük kontrastlı) görüntülerde, görüntünün tümünün renkleri birbirine yakın olur ve görüntüdeki cisimlerin birbirlerinden ayırt edilmesi zorlaşır (Şekil 2.17). Bu bozuklukların giderilmesi için

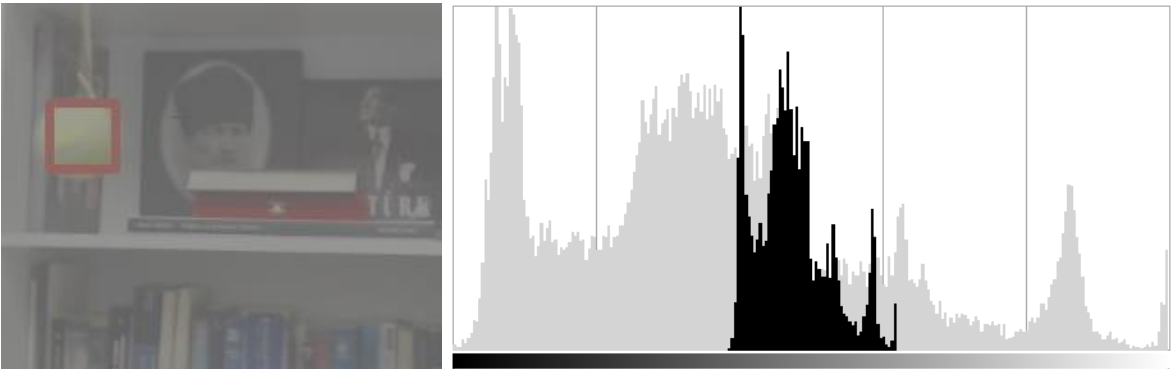
histogram eşitleme (histogram equalization) uygulanır. Histogram eşitleme işleminde, resmin kümülatif gri seviye dağılım skalası üzerinde normal dağılım uygulanır. Bu yeniden dağılım, gri seviye dağılımına dengeleme sağlar. Şekil 2.18'de Şekil 2.15'de görülen koyu renkli görüntüye histogram eşitleme uygulandığında elde edilen görüntü görülmektedir.



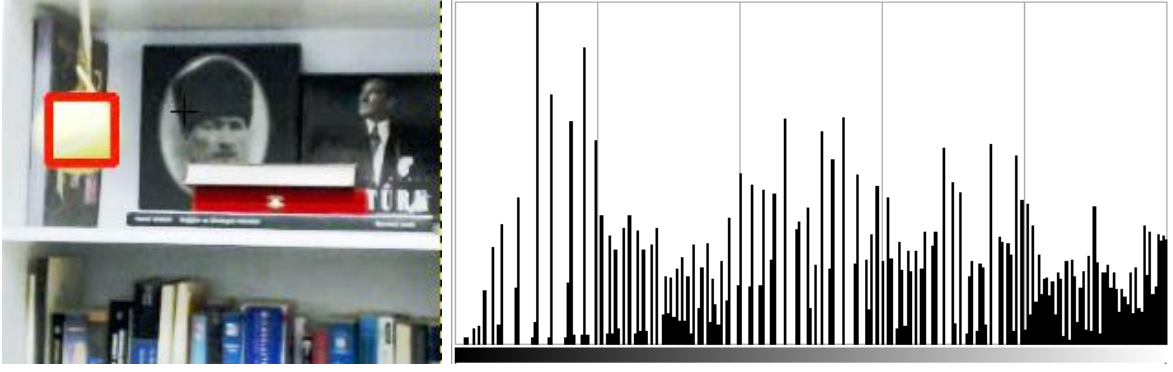
Şekil 2.15 Düşük değerlerde toplanan histogram grafiği



Şekil 2.16 Yüksek değerlerde toplanan histogram grafiği



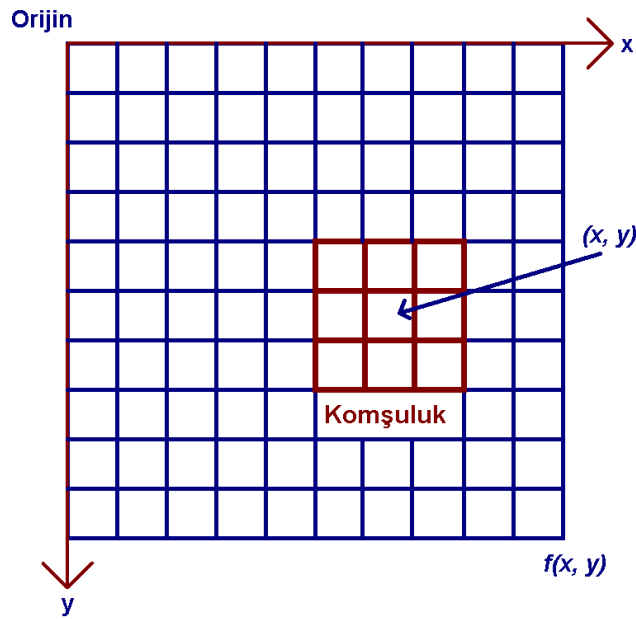
Şekil 2.17 Orta değerlerde toplanan histogram grafiği



Şekil 2.18 Histogram eşitleme uygulanmış bir görüntü (Şekil 2.15'ye uygulanmıştır)

2.8. Uzamsal Süzme

Uzamsal bir süzgeç, genellikle küçük bir dikdörtgenden oluşan bir komşuluktan ve komşuluk içerisinde kalan görüntü piksellerine uygulanmak üzere, önceden tanımlanan bir işlemden meydana gelmektedir. Süzme işlemi, koordinatları komşuluk merkezinin koordinatlarına eşit olan ve değeri süzme işleminin sonucu olan yeni bir piksel yaratır. Süzölmüş bir görüntünün oluşabilmesi için, süzgecin merkezinin girdi görüntüsü üzerindeki tüm pikselleri gezmiş olması gerekir. Görüntü pikselleri üzerinde yapılan işlem doğrusal ise, süzgeç *doğrusal uzamsal süzgeç* olarak adlandırılır. Aksi haller doğrusal olmayan süzmedir. Şekil 2.19'da uzamsal süzmeyi yapan 3 x 3 büyüklüğünde bir komşuluk görülmektedir.



Şekil 2.19 Uzamsal süzme ve komşuluk (Neighbourhood)

Şekil 2.20'de 3 x 3 komşuluk ile uzamsal süzme teknikleri gösterilmiştir. Burada kırmızı renkli kare ve aynı renkli değerler maske ve katsayılarını göstermektedir. Mavi renkli değerler ise görüntünün maske altında kalan piksellerini göstermektedir. Görüntü üzerindeki herhangi bir (x, y) noktasında, süzme işleminin sonucu olan $g(x, y)$, süzgeç katsayıları ve süzgecin içerisinde kalan görüntü piksellerinin çarpımlarının toplamı şeklinde hesaplanır:

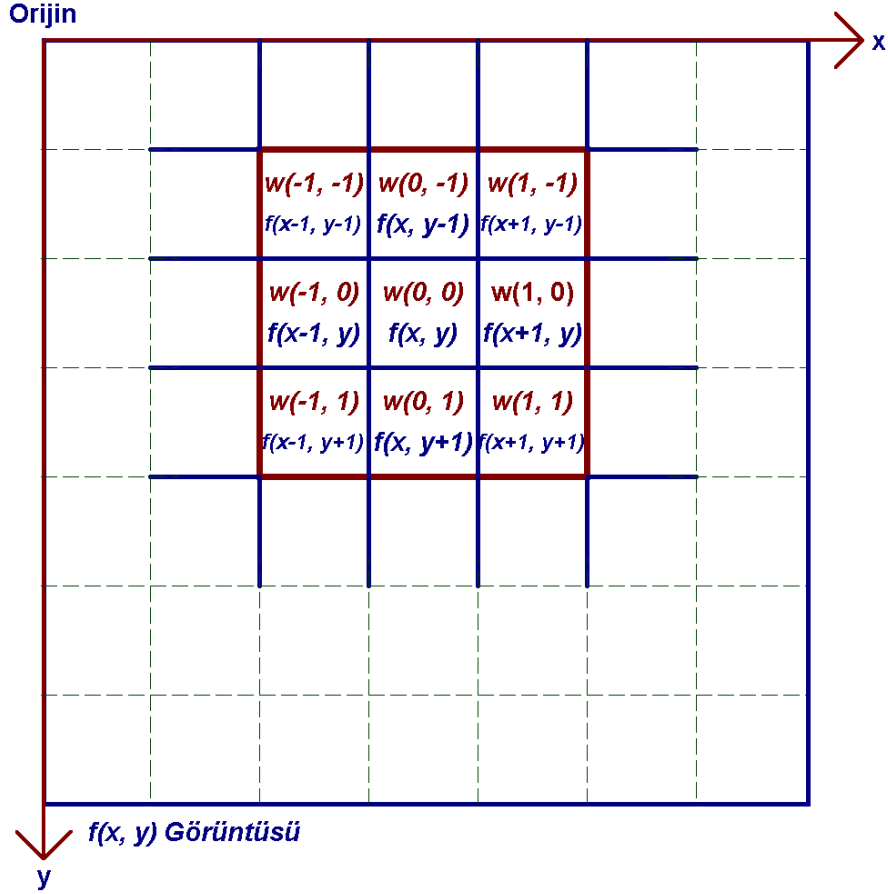
$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots \quad (2.4)$$

$$+ w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

Görüldüğü gibi, süzgecin merkezindeki katsayı $w(0, 0)$, (x, y) noktasındaki piksel ile hizalanmıştır. $m \times n$ büyüklüğündeki bir maske için, m ve n değerlerinin tek ve pozitif tamsayı değerleri olması beklenir. Bu durumdaki en küçük süzgeç büyüklüğü 3 x 3'dür. $M \times N$ büyüklüğündeki bir görüntünün, $m \times n$ büyüklüğünde bir süzgeç ile doğrusal uzamsal süzmesi Eşitlik (2.5)'de gösterilmiştir.

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t) \quad (2.5)$$

burada x ve y değerleri, w süzgecinin her pikselinin, f görüntüsünün her pikselini gezeceği şekilde değiştirilir.



Şekil 2.20 3 x 3 komşuluk ile uzamsal süzme teknikleri

2.9. Görüntü Bulanıklaştırma / Yumuşatma (Blurring / Smoothing)

Yumuşatma süzgeçleri, görüntüyü bulanıklaştırma ve gürültüyü azaltma amacıyla kullanılırlar. Görüntü bulanıklaştırma, nesne çıkarımı öncesinde küçük detayların ortadan kaldırılması ve çizgiler üzerindeki küçük boşlukların doldurulması gibi ön işlemlerde kullanılır. Gürültü azaltma işlemi, doğrusal ve doğrusal olmayan süzme ile yapılan bulanıklaştırma ile yapılabilir.

2.9.1. Doğrusal süzgeç ile yumuşatma

Doğrusal süzgeç ile yapılan yumuşatma işleminin çıktısı, süzgeç maskesi içerisinde kalan komşuluk piksellerinin ortalamasına eşittir. Bu süzgeçler ortalama alıcı süzgeçler (averaging filters) ve alçak geçiren süzgeçler (lowpass filters) olarak da bilinirler.

Yumuşatma süzgeçleri, bir görüntüdeki her bir pikselin değerinin süzgeç maskesi içerisinde kalan komşuluğun yoğunluk değerleri ortalamasına eşitlenmesi fikrine

dayanır. Bu şekilde, görüntü içerisindeki keskin yoğunluk geçişleri ortalama değere çekilerek yumuşatılır. Görüntü içerisinde rastgele dağılmış olan gürültü, genellikle keskin yoğunluk geçişlerine sebep olacağından, yumuşatmanın en etkili uygulama alanı gürültü azaltmaktır.

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Şekil 2.21 3 x 3 yumuşatma (ortalama alan) süzgeç maskesi

Şekil 2.21’de 3 x 3 büyüklüğünde ortalama (yumuşatma) süzgeci görülmektedir. Bu süzgecin kullanımı, maske altında kalan piksellerin ortalamasını verir. Bu işlemin matematiksel gösterimi Eşitlik (2.6)’da verilmiştir.

$$R = \frac{1}{9} \sum_{i=1}^9 z_i \quad (2.6)$$

$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

Şekil 2.22 3 x 3 ağırlıklı (weighted) ortalama alan süzgeç maskesi

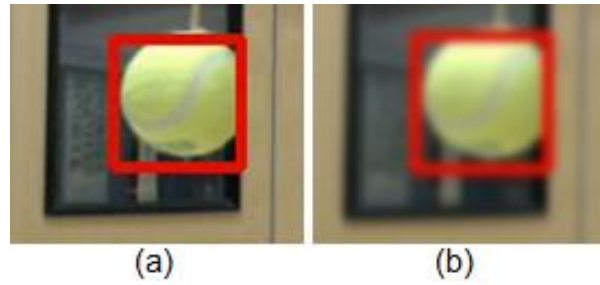
Şekil 2.22’de 3 x 3 büyüklüğünde ağırlıklı ortalama süzgeci görülmektedir. Bu süzmede, pikseller farklı katsayılar ile çarpılır. Buradaki amaç bazı piksellerin diğerlerinden daha fazla ön plana çıkarılmasıdır. Şekil 2.22’de maskenin ortasındaki piksel diğerlerinden daha büyük bir katsayı ile çarpılmış ve ortalamanın hesaplanmasında o piksele daha fazla önem verilmiştir. Diğer pikseller merkeze

olan uzaklıkları ile ters orantılı olarak ağırlıklandırılırlar. Bu şekilde yapılan süzmenin amacı, yumuşatma işlemi sırasında oluşan bulanıklaşma miktarını azaltmaktır.

$M \times N$ büyüklüğündeki bir $f(x, y)$ görüntüsünün, $m \times n$ (m ve n tek sayı) büyüklüğündeki $w(s, t)$ ağırlıklı ortalama süzgeci ile süzülmesi Eşitlik (2.7)'de verilmiştir.

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)} \quad (2.7)$$

burada, $a = (m - 1) / 2$, $b = (n - 1) / 2$ değerlerini almaktadır. Eşitlik (2.7), $x = 0, 1, 2, \dots, M - 1$ ve $y = 0, 1, 2, \dots, N - 1$ için uygulandığında süzölmüş görüntü elde edilmektedir.



Şekil 2.23 Görüntü yumuşatma. Orijinal görüntü (a), yumuşatılmış görüntü (b)

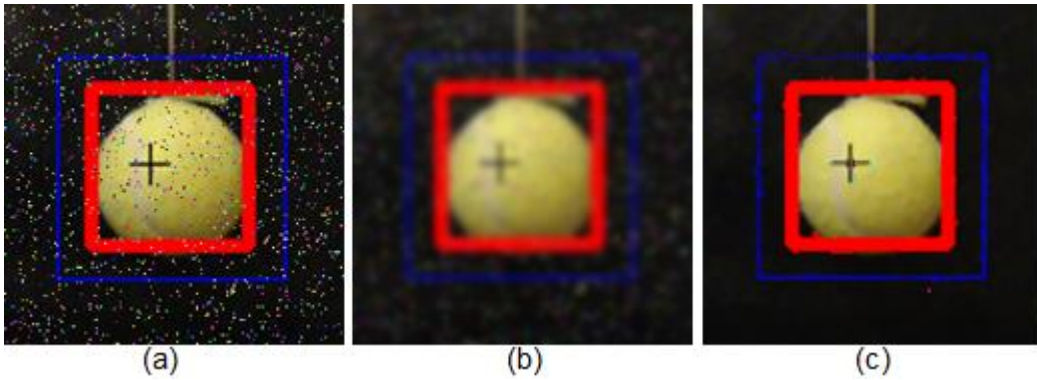
Bir görüntüye (Şekil 2.23 (a)) ağırlıklı ortalama alan süzgeç ile yumuşatma uygulanması sonucu elde edilen yeni görüntü Şekil 2.23 (b)'de verilmiştir. Şekil 2.23'de görüldüğü gibi, yumuşatma işlemi ile ince detaylar ortadan kaldırılmış, ilgi odağı olan nesne ön plana çıkarılmıştır.

Uzamsal ortalama almanın bir diğer önemli kullanım alanı, görüntüyü bulanıklaştırarak küçük boyutlu objelerin arka plan ile kaynaşmalarını sağlamak ve büyük boyutlu nesnelere kolayca belirlenir hale getirmektir. Bu işlem uygulanırken, arka plana karışacak olan nesnelerin boyutu, görüntüye uygulanacak maskenin boyutu ile orantılıdır.

2.9.2. Sıra istatistiksel (Order statistic – Doğrusal olmayan) süzgeçler

Sıra istatistiksel süzgeçler, süzgeç maskesi içerisinde kalan piksellerin sıralanmasına ve merkezde kalan pikselin değerinin sıralı pikseller içerisinde belirlenen sıraya denk gelen piksel ile değiştirilmesi mantığına dayanan süzgeçlerdir. Bu süzgeçlerin en yaygın kullanılanı median (orta değer) süzgeçlerdir. Bu süzgeçte değişim, süzgeç maskesi içerisinde kalan pikseller yoğunluklarına göre sıralandıklarında ortada kalan değer ile yapılır. Median süzgeçlerin yaygın olarak kullanılmalarının sebebi, rastgele gürültünün bazı özel durumlarında, gürültüyü üstün bir başarı ile gidermeleri ve bu işlemi yaparken görüntüyü doğrusal süzgeçlere göre çok daha az bulanıklaştırmalarıdır. Median süzgeçler, görüntü üzerinde siyah ve beyaz noktalar olarak etki bırakan, salt-and-pepper (tuz ve biber) gürültüsünün giderilmesinde ekstra bir etkiye sahiptirler.

Şekil 2.24 (a)'de tuz ve biber gürültüsüne maruz kalmış bir izleme ekranı görüntüsü görülmektedir. Şekil 2.24 (b)'de bu görüntüye ortalama alma süzgeci (averaging filter) uygulanması sonucu elde edilen görüntü görülmektedir. Şekil 2.24 (c)'de ise orijinal görüntüye orta değer süzgeci (median filter) uygulanması sonucu elde edilen görüntü görülmektedir. Şekil 2.24 genel olarak ele alındığında, tuz ve biber gürültüsü için orta değer süzgecinin çok daha başarılı olduğu görülmektedir.



Şekil 2.24 Gürültülü orijinal görüntü (a), 3 x 3 ortalama alma süzgeci uygulanmış görüntü (b), 3 x 3 orta değer süzgeci uygulanmış görüntü (c)

2.10. Pikseller Arası Komşuluk (Pixel Neighborhood)

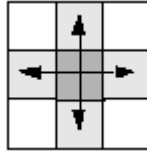
Komşuluk, bir görüntü içerisinde hangi piksellerin diğer pikseller ile bağlantılı olduğunu tanımlar. İkili bir görüntüdeki birbirlerine bağlı piksellerin oluşturduğu gruba, nesne (object) veya bağlantılı bileşen (connected component) denir.

Hangi piksellerin bağlantılı bir bileşen oluşturacağı, piksel komşuluğunun nasıl tanımlandığına bağlıdır. Piksel komşuluğu iki boyutlu ve üç boyutlu olmak üzere, iki boyutlu için 4 ve 8 komşusu; üç boyutlu için, 6, 18 ve 26 komşusu olarak tanımlanabilir.

2.10.1. İki boyutlu komşuluk

2.10.1.1. 4-komşusu

4-komşusu, piksellerin kenarlarının birbirlerine değdiği durumda oluşan komşuluktur. Bunun anlamı, iki pikselin aynı nesnenin parçası olabilmeleri için birbirleriyle yatay veya dikey yönde bağlı olmaları gerektiğidir (Şekil 2.25) [36].

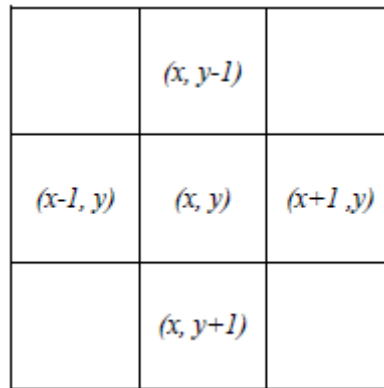


Şekil 2.25 4-komşuluğun bağlantı yönleri

(x, y) koordinatlarındaki bir p pikselinin, 4-komşularının koordinatları;

$(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$ 'dir.

Bu piksel kümesi $N_4(p)$ ile gösterilir.

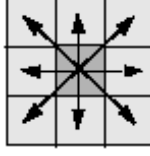


Şekil 2.26 p pikselinin 4-komşuluğu $N_4(p)$

2.10.1.2. 8-komşusu

8-komşusu, piksellerin kenarlarının veya köşelerinin birbirlerine değdiği durumda oluşan komşuluktur. Bunun anlamı, iki pikselin aynı nesnenin parçası olabilmeleri

için birbirleri ile yatay, dikey veya diyagonal yönde bağlı olmaları gerektiğidir (Şekil 2.27) [36].



Şekil 2.27 8-komşuluğun bağlantı yönleri

(x, y) koordinatlarındaki bir p pikselinin, 8-komşularının koordinatları;

$(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$, $(x + 1, y + 1)$, $(x + 1, y - 1)$, $(x - 1, y + 1)$, $(x - 1, y - 1)$ 'dir.

Bu piksel kümesi $N_8(p)$ ile gösterilir.

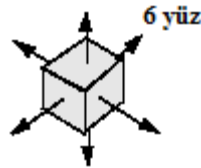
$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$
$(x-1, y)$	(x, y)	$(x+1, y)$
$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$

Şekil 2.28 p pikselinin 8-komşuluğu $N_8(p)$

2.10.2. Üç boyutlu komşuluk

2.10.2.1. 6-komşusu

6-komşusu, piksellerin yüzlerinin (faces) birbirlerine değdiği durumda oluşan komşuluktur (Şekil 2.29) [36].



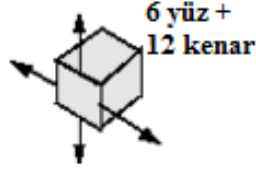
Şekil 2.29 6-komşuluğun bağlantı yönleri

(x, y, z) koordinatlarındaki bir p pikselinin, 6-komşularının koordinatları;

$(x \pm 1, y, z), (x, y \pm 1, z), (x, y, z \pm 1)$ 'dir.

2.10.2.2. 18-komşusu

18-komşusu, piksellerin yüzlerinin veya kenarlarının birbirlerine değdiği durumda oluşan komşuluktur (Şekil 2.30) [36].



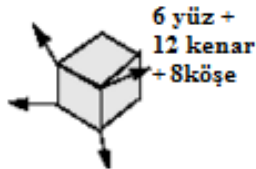
Şekil 2.30 18-komşuluğun bağlantı yönleri

(x, y, z) koordinatlarındaki bir p pikselinin, 18-komşularının koordinatları 6-komşulara ilave gelen;

$(x \pm 1, y \pm 1, z), (x \pm 1, y \pm 1, z \mp 1), (x \pm 1, y, z \pm 1), (x \pm 1, y, z \mp 1), (x, y \pm 1, z \pm 1), (x, y \pm 1, z \mp 1)$ 'dir.

2.10.2.3. 26-komşusu

26-komşusu, piksellerin yüzlerinin, kenarlarının veya köşelerinin birbirlerine değdiği durumda oluşan komşuluktur (Şekil 2.31) [36].



Şekil 2.31 26-komşuluğun bağlantı yönleri

(x, y, z) koordinatlarındaki bir p pikselinin, 26-komşularının koordinatları 18-komşulara ilave gelen;

$(x \pm 1, y \pm 1, z \pm 1), (x \pm 1, y \pm 1, z \mp 1), (x \pm 1, y \mp 1, z \pm 1), (x \pm 1, y \mp 1, z \mp 1), (x \mp 1, y \pm 1, z \pm 1), (x \mp 1, y \pm 1, z \mp 1), (x \mp 1, y \mp 1, z \pm 1), (x \mp 1, y \mp 1, z \mp 1)$ 'dir.

2.11. Görüntü Bölütleme (Segmentation)

Bir görüntüyü, görüntüyü oluşturan parçalarına veya nesnelere ayırma işlemine görüntü bölütleme denir [26]. Başka bir deyişle görüntü bölütleme, homojen bir küme

içerisinde aynı karakteristiğe sahip alanların çıkartılması, arka plandan ayrılması, belirgin bir hale getirilmesi işlemdir [37]. Görüntü işleme uygulamalarının büyük kısmında ön işlemlerde görüntünün içerdiği detaylarla ilgilenilmez. Önemli olan görüntü üzerindeki ilgili nesnelerin alanlarıdır. Bu alanlar görüntünün geri kalanından başarılı bir şekilde ayrıldıktan sonra, üst seviye işlemler ile ilgili alanın detaylarına inilir. Bu sebeple, görüntü işleme uygulamalarında kullanılacak olan nesne tanıma gibi üst seviye işlemlerde ilk ve en zor adım görüntü bölütlemesidir. Nesne tanıma işleminde daha sonradan yapılacak olan, öznitelik çıkarımı, örüntü tanıma gibi yüksek seviyeli uygulamaların başarısı nesne bölütleme işleminin başarısına bağlıdır. Başarılı bir bölütlemeden, görüntüdeki nesnelerin arka plandan ayrıştırılması, aynı doku ve renk özelliklerine sahip alanların oluşturulması beklenir [38].

Literatürde birçok görüntü bölütleme tekniği vardır ancak bu tekniklerin her biri farklı koşullarda farklı sonuçlar verebilmektedir. Bir diğer deyişle, görüntü bölütleme işleminde kullanılacak olan belirli bir teknik bulunmamaktadır. Her tekniğin artıları ve eksileri vardır. Bu teknikler içerisinden doğru olanı, gürültüye karşı duyarlılık, aydınlatma koşulları, büyüklük, arka plan ile cismi ayrıştırma zorluğu gibi değerlere göre seçilmelidir.

Görüntü bölütleme algoritmaları, eşikleme yöntemi ile bölütleme, kenar tabanlı bölütleme, bölgesel tabanlı bölütleme, karma bölütleme ve harekete dayalı bölütleme olmak üzere genellikle 5 temel yaklaşım üzerine kurulmuşlardır.

2.11.1. Görüntü bölütlemenin temelleri

Bir görüntünün kapladığı tüm uzamsal bölgeye R diyelim. Görüntü bölütleme işlemini R bölgesini R_1, R_2, \dots, R_n şeklinde alt bölgelere aşağıdaki özelliklere uyacak şekilde bölme işlemi olarak tanımlayabiliriz [26].

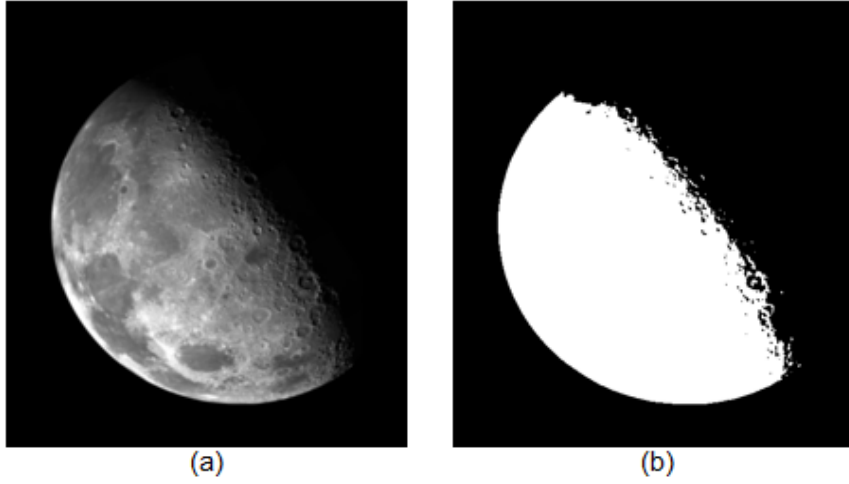
- a) $\bigcup_{i=1}^n R_i = R$.
- b) R_i bağlı komşulukları olan bir bölgedir, $i = 1, 2, \dots, n$.
- c) Tüm i 'ler ve j 'ler için $i \neq j$ olmak üzere $R_i \cap R_j = \emptyset$.
- d) $Q(R_i) = DOĞRU, i = 1, 2, \dots, n$.
- e) Komşu bölgeler R_i ve R_j için $Q(R_i \cup R_j) = YANLIŞ$.

(a) özelliđi, bölütleme işleminin tamamlanmış olması için her pikselin bir bölgeye ait olması gerektiđini gösterir. (b) özelliđi, bölütlenen bölgedeki piksellerin, kendi bölgesindeki piksellerle 4-lü veya 8-li komşulukla bağlantılı olduđunu gösterir. (c) özelliđi, bölgelerin birbirlerinden ayrık olması gerektiđini yani bir bölgeye ait bir pikselin başka bir bölgeye daha dahil olamayacağını gösterir. (d) özelliđi, bir bölgeye dahil olan piksellerin sağlaması gereken özellikler ile ilgilidir. Örneđin, R_i bölgesindeki tüm piksellerin aynı gri seviye değere sahip olduđu durumda $Q(R_i) = DOĐRU$ 'dur. (e) özelliđi, iki farklı bölgedeki piksellerin bađlılık koşulunda belirlenen gri seviye değelerindeki benzerlik kriterlerinin birbirinden farklı olduđunu gösterir.

2.11.2. Görüntü eşikleme (Thresholding) ile bölütleme

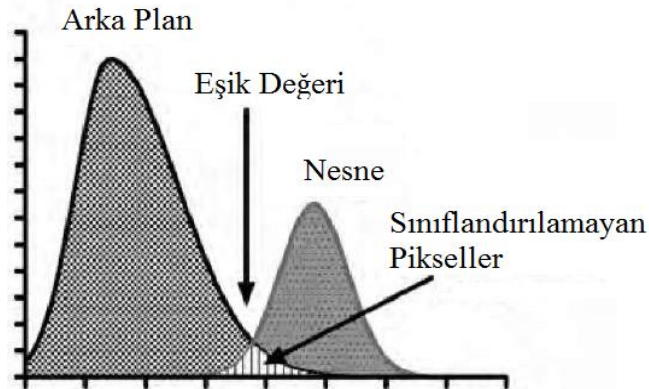
Gri seviye görüntüleri, ikili yani siyah-beyaz görüntüye çevirmek için kullanılan bir yöntemdir. Görüntü bölütleme işlemlerinin en popüler metodudur. Eşikleme yöntemi ile bölütlemeye nesneye ait pikseller kullanılarak bölütleme yapılır. Piksellerden, gri seviye değeri belirli bir değerin üzerinde olanlar bir grubu oluştururlar. Bu değere eşikleme değeri (T) adı verilir. Eşikleme değerinin üzerinde kalan gri seviye değeri ikili sistemde beyaza karşılık gelen 255 değerine çıkarılırken, eşikleme değerinin altında kalan gri seviye değeri ikili sistemde siyaha karşılık gelen 0 değerine düşürülür. Bu şekilde, gri seviye bir görüntü siyah-beyaza çevrilmiş olur ve yeni görüntüde beyaz renkte olan pikseller bölütlenmiş nesneye ait pikseller olur. Bu sebeple, eşik değerinin doğru seçilmesi görüntü eşikleme yöntemi için çok önemlidir. Görüntü eşikleme işleminin matematiksel gösterimi Eşitlik (2.8)'de yapılmıştır. Görüntü eşikleme örneđi Şekil 2.32'de verilmiştir.

$$s = \begin{cases} 255, & r > T \\ 0, & r \leq T \end{cases} \quad (2.8)$$



Şekil 2.32 Görüntü eşikleme örneği. Eşikleme öncesi görüntü (a), eşikleme sonrası elde edilen siyah beyaz görüntü (b)

Görüntü eşiklemenin başarısı üzerindeki en büyük etken, bölütleme yapılacak olan görüntünün histogramındaki tepeliklerin genişlik ve derinlikleridir (Şekil 2.33). Tepeliklerin bu iki özelliğini etkileyen anahtar faktörler: (1) tepe noktaları arasındaki mesafe (tepe noktaları arasındaki mesafenin artması bölgelerin daha başarılı bir şekilde ayrılmasını sağlar). (2) görüntüdeki gürültü oranı (gürültü arttıkça tepeliklerin genişliği artar ve birbirlerine karışmaya başlarlar, bu da bölgelerin başarılı bir şekilde ayrılmasını engeller). (3) nesnelere ve arka planın görece büyüklükleri (arka plan üzerinde çok küçük kalan nesnelere gürültü gibi görünür ve arka plandan ayrılması zorlaşır). (4) aydınlatmanın tek biçimliliği (görüntü üzerinde farklı bölgelerde farklı aydınlatmaların olması ayrımı zorlaştırır). (5) görüntünün yansıma özelliklerinin tek biçimliliği (görüntü üzerindeki farklı bölgelerde farklı yansıma özellikleri görünmesi ayrımı zorlaştırır), şeklinde sıralanabilir.



Şekil 2.33 Görüntü eşikleme uygulanan histogram grafiği. Arka plan, nesne ve sınıflandırılmayan pikseller

2.11.3. Kenar tabanlı bölütleme

Kenar tabanlı bölütleme, nesnenin kenarlarının kullanılarak arka plandan ayrılmasıyla yapılan bölütlemedir. Nesnenin sınırları, farklı nitelikteki bölgeler arasındaki geçişlerden, örneğin renk ve doku, ya da görüntüdeki süreksizliklerden kaynaklanır. Bu durum nesne sınırlarının gri seviyede devamsızlık göstermesine sebep olur. Kenar tabanlı bölütlemede bu devamsızlıkları kullanarak bölütleme yapılır. Bir görüntüye, bayırgı (gradient) işleçleri olan Canny, Prewitt, Roberts, Sobel gibi süzgeçler uygulandığında yüksek değer veren pikseller nesne kenarlarına aday olur. Aday pikseller sonraki adımlarda birleştirilerek nesne kenarlarını temsil eden kapalı eğriler oluşturulur.

Yoğunluk seviyesindeki lokal ve ani değişimlerin oluşturduğu devamsızlıklar baz alınarak yapılan bu bölütlemede görüntünün üç özelliği üzerinde durulur. Bunlar; tekil noktalar, çizgiler ve kenarlardır. Kenar bulucular, kenar piksellerini bulmak üzere tasarlanmış lokal görüntü işleme metotlarıdır. Herhangi bir tarafındaki arka plan yoğunluğu, kendi piksellerinin yoğunluğundan çok büyük veya çok küçük olan bir çizgi, kenar parçası olarak yorumlanabilir.

Bir görüntü üzerindeki, ani ve lokal yoğunluk değişimleri türev kullanılarak saptanabilir. Bu amaçla kullanıldığında, birinci ve ikinci derece türevler en iyi sonucu verirler.

Sayısal bir fonksiyonun türevi, farklılıklar şeklinde tanımlanır. Bu farklılıkların tahmin edilmesi için farklı yöntemler vardır. Bu tahminlerde; birinci derece türevin (1) sabit yoğunluktaki alanlarda 0 olması; (2) yoğunluk değişiminin başladığı bir noktada 0 olmaması; (3) yoğunluk değişimi boyunca 0 olmaması zorunludur. Benzer bir şekilde ikinci derece türevin (1) sabit yoğunluktaki alanlarda 0 olması; (2) yoğunluk değişiminin başladığı ve bittiği noktalarda 0 olmaması; (3) yoğunluk değişimi boyunca 0 olması zorunludur.

Türevlerin zorunlulukları incelendiğinde özetle şu sonuçlar ortaya çıkarılabilir: (1) birinci derece türevler genellikle görüntülerdeki kalın kenarları ortaya çıkarır. (2) ikinci derece türevlerin hassas detaylara olan tepkisi daha büyüktür, ince çizgiler, izole noktalar ve gürültü gibi etkenlere daha fazla tepki verirler. (3) ikinci derece türevler rampa ve basamak geçişlerinde çift kenar tepkisi oluştururlar. (4) ikinci

türevin işareti bir kenara olan geçişin aydınlıktan karanlığa veya karanlıktan aydınlığa doğru olduğunun belirlenmesinde kullanılabilir.

Kenar tabanlı bölütlemeye, kenarların belirlenmesi için üç temel adım vardır [26]:

1. Gürültüyü azaltmak için görüntü yumuşatma.
2. Kenar noktalarının belirlenmesi. Kenar noktaları olabilecek bütün aday noktaların görüntü içerisinde ayrılmasını sağlayan lokal bir işlemdir.
3. Kenar yerini tayin etme. Bu adımın amacı, kenar noktası adayları arasından gerçek bir kenarı oluşturan doğru noktaların ayrılmasıdır.

Birinci ve ikinci türevleri bir görüntünün her bir pikseli için hesaplamada kullanılan yöntem uzamsal süzmedir. Uzamsal süzme işlemi Bölüm 2.8'de bahsedildiği gibi, 3 x 3 maske ile Eşitlik (2.4) kullanılarak yapılır.

Görüntü üzerindeki izole noktaların belirlenmesine en iyi yöntem olan ikinci türevi kullanan Laplace operatörü (Laplacian) maskesi Şekil 2.34'de gösterilmiştir.

0	1	0
1	-4	1
0	1	0

Şekil 2.34 Laplace operatörü maskesi

Görüntü üzerindeki çizgilerin belirlenmesinde en iyi yöntem olan ikinci türevi kullanan Laplace operatörü maskesi, yatay (horizontal), +45°, dikey (vertical) ve -45° eğimli çizgilere en yüksek tepkiyi verecek şekilleri ile Şekil 2.35'de gösterilmiştir.

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

Şekil 2.35 Çizgi belirleme maskeleri, yatay, +45°, dikey ve -45°

Görüntü üzerindeki köşegen kenarların belirlenmesinde kullanılabilecek bayırgı işleçlerinin ilk örneklerinden biri olan “Roberts cross-gradient” işleçleri [39] Şekil 2.36’da gösterilmiştir.

-1	0	0	-1
0	1	1	0

Roberts

Şekil 2.36 Roberts bayırgı işleci maskeleri

Kenar yönü belirlemede, Roberts işleçlerinden daha başarılı olan 3 x 3 büyüklüğündeki Prewitt işleçleri [40] ve Prewitt işleçlerine göre sağladıkları yumuşatma özelliği sayesinde, gürültüden daha az etkilenen Sobel işleçleri [41] Şekil 2.37’de gösterilmiştir.

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

Prewitt

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

Sobel

Şekil 2.37 Prewitt ve Sobel işleçleri

2.11.4. Bölgesel tabanlı bölütleme

Bölgesel tabanlı bölütleme, nesneye ait piksellerin uzamsal yakınlıkları ve gri seviyede önceden belirlenmiş benzer özelliklerine göre yapılan bölütlemedir. Bu işlem sırasında gruplandırılan birbirleriyle bağlantılı pikseller bir bölge oluşturur ve oluşan bu bölgeler görüntüde bölütlenmiş nesnelere temsil eder. Bölgesel tabanlı bölütlemede asıl amaç görüntüyü maksimum homojenliğe sahip alanlarına ayırmaktır.

Bölgesel tabanlı bölütlemede, pikselleri veya alt bölgeleri önceden belirlenen büyüme kriterine göre gruplandırarak daha büyük bölgeler oluşturmak amaçlanır. Bu bölütleme türünde, bölütlenecek olan nesnelere ve ilgisiz nesnelere üzerinden tohum (seed) adı verilen piksel veya piksel grupları seçilir. Tohumların belirlenmesinden sonra, bu noktalardan başlanarak, tohum noktalarına komşu olan ve tohum noktalarına belirlenen kriterleri sağlayacak kadar benzerlik gösteren piksellerin tohum gruplarına eklenmesi ile bölgeler büyütülür. Büyüme kriteri, piksellerin veya alt bölgelerin, yoğunluk değerine, doku özelliklerine veya renk özelliklerine göre yapılabilir. Bu işlemin algoritmik ifadesi aşağıdaki gibidir;

1. Bölütlenmesi istenilen bölgeler içerisinde ilk tohum değerleri seçilir,

2. Tohumların komşuluk pikselleri için, tohum ve piksel arasındaki benzerliğe bakılır,
3. Benzerlik seviyesi benzerlik kriterinin üstünde olan pikseller, tohuma eklenerek küçük bölge oluşturulur,
4. Yeni oluşan bölgedeki tüm piksellerin komşuluk pikselleri için, tohum ve piksel arasındaki benzerliğe bakılır,
5. Benzerlik seviyesi, benzerlik kriterinin üstünde olan pikseller küçük bölgeye dahil edilerek daha büyük bölgeler oluşturulur,
6. Bu işlem tüm pikseller bir bölgeye ait olana dek devam eder.

Bir $g(x, y)$ pikselinin, A_i tohumunun bölgesine ait olup olmadığı Eşitlik(2.9)'da gösterilen şekilde belirlenir.

$$Q = \begin{cases} DOĞRU, & \text{eğer } |g(x, y) - A_i(x, y)| \geq T \text{ ise} \\ YANLIŞ, & \text{diğer durumlar} \end{cases} \quad (2.9)$$



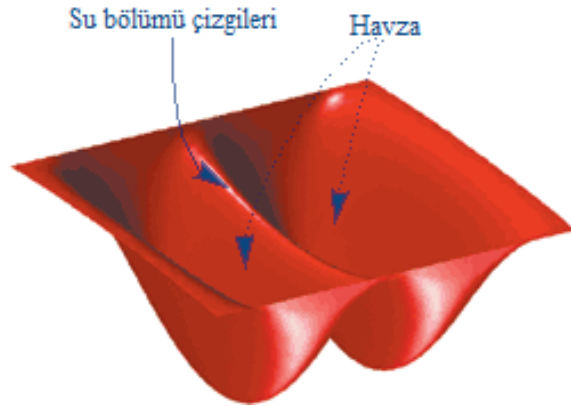
Şekil 2.38 Bölgesel bölütleme, tohumların seçilmesi (a), küçük grupların oluşturulması (b), küçük grupların büyütülerek büyük bölgeleri oluşturması (c)

Bölgesel tabanlı bölütleme işleminde, tohumların seçimi algoritma performansına direkt olarak etkilidir. Yarı otomatik olan sistemlerde, tohum seçimi kullanıcı tarafından yapılır ve sistem bu tohumlara göre bölütleme işlemini gerçekleştirir. Tam otomatik sistemlerde ise, tohum seçimi başka bir görüntü işleme tekniğinin çıktısından faydalanarak sistem tarafından yapılır. Tohum seçiminde olduğu gibi, benzerlik kriterinin seçimi de algoritma performansına direkt olarak etki eder. Benzerlik kriteri değerinin belirlenmesinde de, tohum seçimindeki gibi kullanıcı

girdisine dayanan yarı otomatik ve görüntü işleme tekniklerinin çıktısından faydalanılan tam otomatik seçim yöntemleri mevcuttur.

2.11.5. Karma bölütleme

Karma bölütleme, nesneye ait piksellerin ve nesne sınırlarının kullanılmasıyla yapılan bölütlemedir. Morfolojik su bölümü çizgisi (Morphological Watershed) yöntemi en temel karma bölütleme yöntemidir. Morfolojik su bölümü çizgisi yöntemi görüntünün topolojisini kullanır. Bu yöntemde, görüntü coğrafi bir şekil olarak kabul edilir ve görüntü piksellerinin değerleri yükseklik olarak kabul edilir. Değeri düşük olan pikseller çukurları, değeri yüksek olan pikseller ise yükselteleri gösterir. Topoloji ters çevrilip su doldurulmaya başlanır. Bu durumda su, en dip noktadan başlayarak farklı havzalara dolacaktır. İki farklı havzadan gelen suyun birbirine karışmaya başladığı noktalarda barajlar inşa edilerek bölütlerin birbirinden ayrılması sağlanır (Şekil 2.39). Bu barajlar su bölümü çizgileri (watershed) olarak adlandırılır. Su bölümü çizgileri ile her bir minimum nokta, bu minimum ile ilişkili havzayı sınırlandıracak şekilde çevrilmiş olur. Çevrilen bu bölümler görüntüde farklı bölgeleri temsil eder [42] [43].



Şekil 2.39 Su bölümü çizgisi yöntemi ([44]'den Türkçeleştirilerek)

2.11.6. Harekete dayalı bölütleme

Harekete dayalı bölütleme tekniklerinin çoğu birbirini izleyen sıralı görüntülerdeki farkı bulmaya dayanır. Bu işlem, hareketli nesnenin bulunduğu ortamdan t_i ve t_j zamanlarında alınan $f(x, y, t_i)$ ve $f(x, y, t_j)$ görüntülerinin piksel piksel kıyaslanmasıyla yapılır [26]. Aynı ortamın farklı zamanlarda alınan görüntülerini kıyaslamamanın en temel yolu, iki görüntünün birbirlerinden çıkarılmasıyla bulunan fark

görüntüsünün elde edilmesidir. Bu işlemde, hareket eden nesnenin olmayıp yalnızca sabit elemanların yer aldığı görüntü, referans görüntüsü olarak kabul edilir. Hareketli nesnenin olduğu takip eden görüntüler, referans görüntüsünden çıkarılarak sabit elemanların yok edildiği, yalnızca hareketli nesnelerin görüldüğü fark görüntüsü elde edilir.

t_i ve t_j gibi iki farklı zamanda alınmış görüntülerden elde edilen fark görüntüsünün matematiksel ifadesi Eşitlik(2.10)'da verilmiştir.

$$d_{ij}(x, y) = \begin{cases} 255, & \text{eğer } |f(x, y, t_i) - f(x, y, t_j)| > T \text{ ise} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (2.10)$$

burada T eşik değerini ifade eder ve fark görüntüsü üzerinde hangi noktaların hareketli bir cisim gösterdiğinin kararı bu değere göre verilir. Düşük T değerleri sistemin ufak farkları dahi hareketli cisim olarak görmesine sebep olurken, yüksek T değerleri, hareketli cisim tespiti için çok büyük değişimler olma gerekliliğini doğurur.

Harekete dayalı bölütleme uygulamalarında, yalnızca hareketsiz elemanların bulunduğu referans görüntüsünün elde edilmesinin mümkün olmadığı durumlar olabilir. Bu gibi durumlarda, hareketli nesnenin bulunduğu iki görüntü kullanılarak, hareketli nesnenin hareketinden oluşan bir fark görüntüsü elde edilir. Bu tipteki fark görüntüsünün matematiksel ifadesi Eşitlik(2.11)'de verilmiştir.

$$g_{ij}(x, y) = f(x, y, t_i) - f(x, y, t_j) \quad (2.11)$$

Bu yöntemde, piksellerin birbirlerinden çıkarılması sebebiyle, elde edilen $g_{ij}(x, y)$ görüntüsünün piksellerinin değer aralığı $[-255, 255]$ aralığındadır. Bu sebeple, fark görüntüsü üzerinde normalizasyon işlemi uygulanarak değerlerin $[0, 255]$ aralığına çekilmesi gerekir.

3. NESNE İZLEME YÖNTEMLERİ

Nesne izleme, hareketli bir nesnenin izlediği yolun görüntü düzlemi üzerinde tahmin edilmesi problemidir. Nesne izleme işleminin gerçekleştirilmesinde ilk adım, nesnenin uygun bir gösteriminin tanımlanmasıdır. İkinci adım, izleyiciye girdi olarak uygulanacak olan nesne özelliklerinin belirlenmesidir. Nesne izleme algoritmalarının çoğunluğunda, izlenecek olan nesnenin ilk veya her çerçevede belirlenmesi gerekir. Farklı izleme algoritmalarının başarımları, izledikleri nesnelerin görünüşlerine, şekillerine, sayılarına, ortam aydınlatmasına ve nesne ile kameranın hareketlerine bağlıdır.

3.1. Nesne Gösterimi

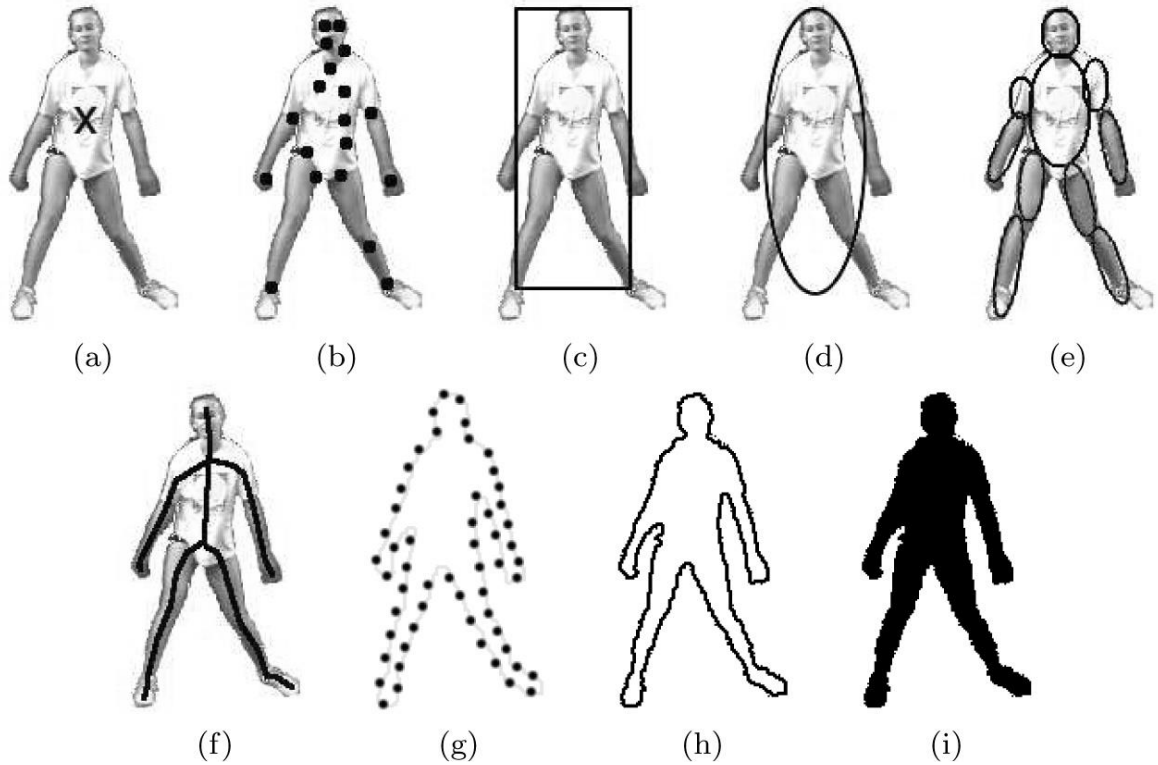
Bir izleme senaryosunda nesne tanımı, daha ileri analizler için ilgi duyulan herhangi bir cisim olarak yapılabilir. Bu nesnelere şekilleri veya görünüşleri ile sembolize edilerek gösterilirler. Nesnelerin şekillerine ve görünüşlerine göre yapılabilecek gösterimler şu şekildedir [45];

- *Noktalar ile gösterim.* Nesne, nesnenin kütle merkezinde bulunan bir nokta ile (Şekil 3.1 (a)) veya noktalar kümesi ile (Şekil 3.1 (b)) ifade edilir. Genellikle, görüntü içerisinde küçük bölgeleri işgal eden nesnelerin izlenmesinde kullanılır.
- *Temel geometrik şekil ile gösterim.* Nesne şekli dikdörtgen, elips ve benzeri şekillerle ifade edilir (Şekil 3.1 (c, d)). Bu şekillerle gösterilen nesnelerin hareketleri genellikle ötelemeli, afin veya izdüşümsel olmaktadır. Temel geometrik şekil ile gösterim genellikle katı, esnek olmayan nesnelere için uygun olmakla birlikte, esnek nesnelere izlemek için de kullanılır.

Temel geometrik şekil ile gösterim, tez çalışması kapsamında geliştirilen sistemde kullanılan nesne gösterim yöntemidir.

- *Nesne silüeti ve çevre çizgisi ile gösterim.* Çevre çizgisi bir nesnenin sınırlarını tanımlar (Şekil 3.1 (g, h)). Çevre çizgisinin içinde kalan alan nesnenin silüeti olarak adlandırılır (Şekil 3.1 (i)). Çevre çizgisi ve silüet gösterimleri, karmaşık esnek cisimlerin izlenmesine uygundur.

- *Eklemlı şekil modelleri ile gösterim.* Eklemlı nesnelere, birbirlerine birleşme yerlerinden bağlanan gövde parçalarından oluşurlar. Buna örnek olarak insan vücudu; gövde, bacaklar, kollar ve kafanın eklemler ile bağlanması ile oluşur. Bu gösterimde parçalar arasındaki ilişki kinematik hareket modelleri ile çıkarılabilir. Eklemlı şekil gösterimi, nesneyi oluşturan parçaların silindirik veya elips gibi geometrik şekiller ile modellenmesi ile yapılır (Şekil 3.1 (e)).
- *İskelet modeli ile gösterim.* Nesnenin iskeleti, nesnenin silüetine orta eksen dönüşümü uygulanarak çıkarılabilir. İskelet modeli genellikle nesne tanımada kullanılır. İskelet gösterimi, eklemlı ve esnek olmayan nesnelere için kullanılabilir (Şekil 3.1 (f)).



Şekil 3.1 Nesne gösterimleri. Kütle merkezi (a), çoklu noktalar (b), dikdörtgen şekil (c), eliptik şekil (d), parça temelli çoklu şekil (e), nesne iskeleti (f), çevre çizgisi kontrol noktaları (g), tüm çevre çizgisi (h), nesne silüeti (i)

3.2. İzleme İçin Özellik Seçimi

Nesne izlemede doğru özellik seçimi kritik bir rol oynar. Görsel bir özellikte en çok aranan özellik tekilliktir. Tekil olan bir özellik seçimi, nesnelere öznelik uzayı içerisinde kolayca fark edilebilmelerini sağlar. Nesne izleme algoritmalarının

çoğunluğu, görsel özelliklerin birleşimlerini kullanırlar. En yaygın görsel özellikler şu şekildedir [45];

- *Renk*. Bir nesnenin rengini en çok etkileyen fiziksel faktörler, ışık kaynağının tayfsal güç dağılımı ve nesne yüzeyinin yansıtma özellikleridir. Görüntü işlemede renkleri temsil etme amacıyla, RGB, CIE L*u*v [46], CIE L*a*b [46] ve HSV gibi farklı renk uzayları kullanılabilir. Farklı durumlara özel avantaj ve dezavantajları olan bu renk uzaylarının hangisinin nesne takibinde kullanılmasının daha verimli olduğuna dair kesin bir kanı bulunmamaktadır.

HSV renk uzayı kullanılarak elde edilen renk görsel özelliği, tez çalışması kapsamında geliştirilen sistemde kullanılan izleme özelliğidir.

- *Kenarlar*. Nesne sınırları, görüntü yoğunluğu üzerinde genellikle büyük değişimlere sebep olurlar. Kenarların önemli bir özelliği, renk özellikleri ile kıyaslandıklarında, aydınlatma değişimlerine karşı daha az duyarlı olmalarıdır. Nesnelerin sınırlarını izleyen algoritmalar genellikle kenar özelliklerini kullanırlar.
- *Optik akış*. Optik akış bir bölge içerisindeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Ardışık çerçevelerdeki eş piksellerin parlaklıklarının sabit kalacağı varsayımı yapılarak, parlaklık kısıtı üzerinden hesaplanır. Optik akış genellikle hareket tabanlı bölütleme ve izleme uygulamalarında kullanılan bir özelliktir. Horn ve Schunk [1981], Lucas ve Kanade [1981], Black ve Anandan [1996], ve Szelinski ve Coughlan [1997] popüler optik akış hesaplama yöntemleridir.
- *Doku*. Doku, bir yüzeyin pürüzsüzlük ve düzenlilik gibi özelliklerinin miktarını belirleyen yoğunluk değişimi ölçümüdür. Renk ile kıyaslandığında, doku tanımlayıcılarının oluşturulması için gereken bir ön işlem adımı vardır. Kenar özelliklerinde olduğu gibi, doku özellikleri de renk ile kıyaslandıklarında aydınlatma değişimlerine karşı daha az duyarlıdır.

3.3. Nesne Belirleme

Bütün izleme algoritmaları, her çerçeve için veya nesnenin video görüntüsüne girdiği ilk anda çalışacak nesne belirleme yöntemlerine ihtiyaç duyar. Nesne belirleme için kullanılan yaygın yöntemlerden biri tek bir çerçevedeki bilgileri kullanma yöntemidir.

Bununla birlikte, bazı nesne belirleme yöntemlerinde, birbirini takip eden çerçevelerden alınan zamana bağlı bilgi kullanılarak yanlış belirlenmelerin sayısı azaltılmaya çalışılır. Bu zamana bağlı bilgi genellikle, çerçevelerin farklarından elde edilen, birbirlerini takip eden çerçeveler üzerinde yer değiştiren bölgeleri ön plana çıkaran bilgilerdir. Bir görüntü içerisindeki nesnenin bölgesi belirlendikten sonra, bu nesneyi takip eden çerçevelerde kendisi ile eşleştirme işlemi izleyicinin görevidir.

En yaygın olarak kullanılan nesne belirleme yöntemleri Çizelge 3.1’de kategoriler ve uygulayıcı yöntemleri ile birlikte verilmiştir.

Çizelge 3.1 Nesne belirleme kategorileri

Kategoriler	Temsili Yöntemler
Nokta Belirleyiciler (Point Detectors)	Moravec Belirleyicisi (Moravec’s detector) [Moravec 1979], Harris Belirleyicisi [Harris ve Stephens 1988], Ölçek bağımsız özellik dönüştürücü (Scale invariant feature transform) [Lowe 2004], Afin bağımsız nokta belirleyici [Mikolajczyk ve Schmid 2002].
Bölütme (Segmentation)	Ortalama Kayma (Mean shift) [Comaniciu ve Meer 1999], Çizge kesik (Graph-cut) [Shi ve Malik 200], Aktif çevre çizgileri (Active contours) [Caselles vd. 1995].
Arka plan modelleme (Background modelling)	Gaussianlar karışımı (Mixture of Gaussians) [Stauffer ve Grimson 2000], Öz arka plan (Eigenbackground) [Oliver vd. 2000], Wall flower [Toyama vd. 1999], Dinamik doku arka planı (Dynamic texture background) [Monnet vd. 2003].
Güdümlü sınıflandırıcılar (Supervised classifiers)	Destek vektör makinesi (Support vector machine) [Papageorgiou vd. 1998], Sinir ağları (Neural networks) [Rowley vd. 1998], Uyarlanabilir arttırma (Adaptive boosting) [Viola vd. 2003].

3.3.1. Nokta belirleyiciler (Point detectors)

Nokta belirleyiciler, kendi çevrelerinde anlamlı bir dokuya sahip olan ilgi noktalarının (interest points), görüntüler içerisinde bulunması için kullanılırlar. İlgi noktaları, devinim, çift kameralı görüntü alımı (stereo) ve izleme problemlerinde uzun süredir kullanılan terimlerdir. İlgi noktalarında aranan özellik, kamera bakış açısından ve ortam aydınlatmasındaki değişimlerden bağımsız olmalarıdır. Literatürde sıkça kullanılan ilgi noktası belirleyiciler; Moravec ilgi işleci (Moravec's interest operatör) [47], Harris ilgi noktası belirleyicisi (Harris interest point detector) [48], KLT belirleyici [49] ve SIFT belirleyicidir [50].

3.3.2. Arka plan çıkarımı

Arka plan çıkarımı ile nesne belirleme, sahnenin arka plan modeli denilen bir gösteriminin çıkarılması ve bu gösterimin kullanılarak takip eden her çerçeve için sapmaların bulunması ile yapılır. Arka plan modeli üzerindeki anlamlı değişimler, görüntünün o bölgesinde hareketli bir nesnenin olduğu anlamına gelir. Bu bölgedeki pikseller daha ileri işlemlerde kullanılmak üzere işaretlenir. İşaretlenen pikseller üzerinde genellikle, bağlı elemanların çıkarılması algoritması kullanılarak nesneye karşılık gelen bağlı bölgeler elde edilir. Bu işlemin tümüne arka plan çıkarımı denir. Literatürde sıkça kullanılan arka plan çıkarımı yöntemleri; Gaussianlar karışımı (Mixture of Gaussians) [51], öz arka plan (Eigenbackground) [52] ve dinamik doku arka planıdır (Dynamic texture background) [53].

Arka plan çıkarımının en önemli kısıtlaması sabit kameralara olan gereksinimidir. Kameranın hareketi genellikle arka plan modelinde bozulmaya sebep olur. Bu sebeple sabit kameralı uygulamalarda kullanılması daha sağlıklıdır.

3.3.3. Bölütleme

Görüntü bölütleme algoritmalarının amacı görüntüyü algısal olarak benzer bölgelere ayırmaktır. Her bölütleme algoritmasının çözmeye çalıştığı iki problem; iyi bir bölütleme için gereken kriter ve etkili bir bölütleme tekniğidir. Nesne izleme ile ilgili olarak kullanılacak bölütleme teknikleri şu şekildedir;

- *Ortalama değer kaymalı kümeleme (Mean-Shift clustering)*. Comaniciu ve Meer [54] tarafından geliştirilen bu yöntemde, uzamsal konum ve renk uzayı

bilgilerine göre kümeleme yapılır. Algoritma çalışmaya her biri bir kümeyi temsil eden, çok sayıda küme merkezinin görüntü üzerinde rastgele konumlandırılması ile başlar. Daha sonra her küme, merkezi o kümenin merkezinde olduğu kabul edilen, bir elipsoidal içinde kalan değerlerin ortalaması olan noktaya taşınır. Bu işlem küme merkezleri hareket etmemeye başlayana kadar tekrar eder. Ortalama değer kayması tekrarları sırasında, bazı kümeler birleşerek yeni bir küme oluşturabilirler.

Ortalama değer kaymalı kümeleme, tez çalışması kapsamında geliştirilen sistemde kullanılan nesne belirleme yöntemidir.

- *Çizge kesik kullanarak görüntü bölütleme (Graph-Cut)*. Bir görüntü, “büyük bir resim” olarak düşünülmekte ve boğumları (V) pikseller olan, bağları (E) ise piksellerin parlaklık, renk veya doku benzerliğine bağlı olarak oluşturulan bir ağırlıklandırılmış çizge (G) [55] üzerine oturtulur [56]. Çizge üzerinden bölütleme, kesik değerlerinin çeşitli özel matrislerin özdeğer vektörlerinin ayrışımı ile eniyilenmesi ve bu işlemin döngüsel bir şekilde çizgenin parçalanması için kullanılması sonucunda sağlanır [56].
- *Aktif çevre çizgileri (Active contours)*. Bölütlemenin, nesnenin sınırları üzerinde kapalı bir çevre çizgisi evrilmesiyle sağlandığı yöntemdir. Çevre çizgileri nesne bölgesini sıkıca çevreler. Çevre çizgisinin evrilmesi, çizginin bölgeye olan uyumluluğunu gösteren enerji fonksiyonu (Eşitlik (3.1)) kullanılarak yapılır [57].

$$E(\Gamma) = \int_0^1 E_{int}(v) + E_{im}(v) + E_{ext}(v) ds \quad (3.1) \quad [57]$$

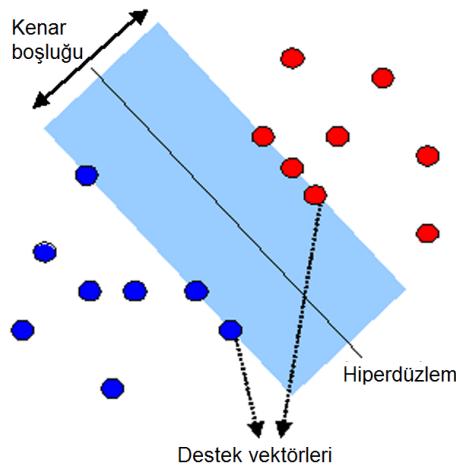
burada Γ ark boyunu, E_{int} bükülme sonucu oluşan oluşun enerjisini, E_{im} görünüm temelli enerjiyi, E_{ext} dış kısıtlamaların enerjisini ifade etmektedir.

3.3.4. Öğretici ile öğrenme

Nesne belirleme işlemi, nesnenin farklı açılardan elde edilen görüntülerinden oluşan örnek setinin otomatik olarak öğrenilmesi ile yapılabilir. Sistem öğrenme sırasında girilen girdileri, istenilen çıkışlar ile eşleştiren fonksiyonları üretir. Öğretici ile

öğrenme yöntemlerinde, her bir nesne sınıfı için büyük miktarda örneğe ihtiyaç duyulur. Nesne izleme ile ilgili olarak kullanılacak öğretici ile öğrenme teknikleri şu şekildedir;

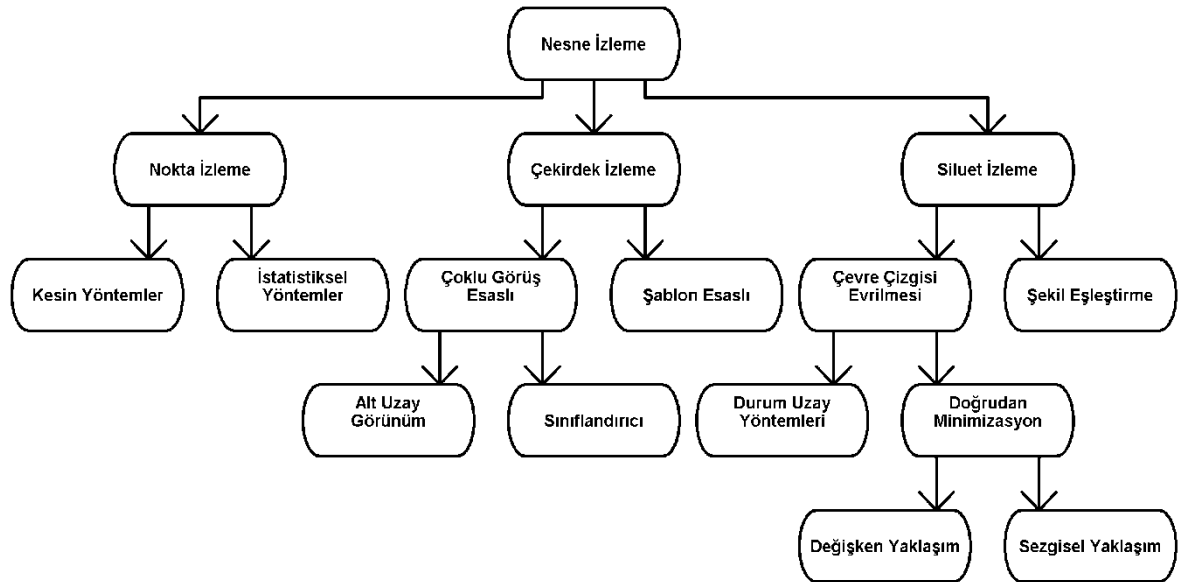
- *Uyarlanabilir arttırma (Adaptive boosting)*. Arttırma, her biri kısmen doğru olan birden çok temel kümelendiricinin tekrarlı olarak birleştirilmesi ile kesin doğru olan kümelendiricinin bulunmasıdır. Bu algoritmanın ilk adımı, ağırlıkların ilk değerlerinin örnek setleri içinde rastgele olarak dağıtılmasıdır. Bir sonraki adımda, arttırma sistemi kümelendirilememiş verilerin ağırlıkları ile orantılı olan en düşük hatayı veren temel bir kümelendiriciyi seçer. Seçilen kümeleyici tarafından kümelendirilemeyen verilerin ağırlıkları arttırılır. Algoritma bu şekilde kümelendirilemeyen veriler üzerinde bir sonraki tekrarda daha iyi performans gösterecek başka kümelendiricilere yönelir.
- *Destek vektör makinesi (Support vector machine) [58]*. Destek vektör makineleri, bir sınıfı diğerinden ayıran maksimum kenar boşluğu olan hiperdüzlemi kullanarak bir veri setini iki kümeye ayırır [59] (Şekil 3.2). Hiperdüzlemin kenar boşluğu, hiperdüzlem ve ona en yakın olan veri noktalarının arasındaki uzaklıktır. Kenar boşluğu çizgileri üzerinde kalan veri noktaları destek vektörü olarak adlandırılır. Nesne belirlemede kullanıldığında, destek vektörü olan veriler nesne sınıfını, kenar boşluğu dışında kalan veriler nesne olmayan sınıfı temsil ederler.



Şekil 3.2 Destek vektör makinesi

3.4. Nesne İzleme

Nesne izleyicinin amacı, hareketli bir nesnenin konumunu video akışının her bir çerçevesinde bularak, nesnenin izlediği yolu çıkarmaktır. Nesnenin belirlenmesi ve video akışındaki çerçeveler arasında doğruluğunun sağlanması işlemi ayrı ayrı veya birlikte yapılabilir. Ayrı ayrı olan durumda, nesne olması muhtemel bölgeler nesne belirleme algoritması ile elde edilir ve izleyici bu nesnelere çerçeveden çerçeveye doğrular. Birleşik olan durumda ise, nesne bölgesi ve doğruluğu önceki çerçevelerden alınan nesne konumu ve bölge bilgileri kullanılarak tekrarlı olarak güncellenir. Her iki yöntemde de nesnelere, Bölüm 3.1’de bahsi geçen şekil ve görünüm modelleri ile temsil edilirler.



Şekil 3.3 İzleme yöntemlerinin sınıflandırılması ([45]’den Türkçeleştirilerek)

Çizelge 3.2 İzleme kategorileri [45]

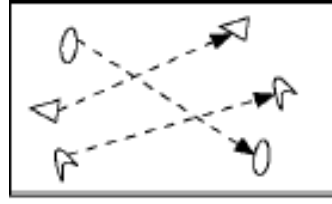
Kategoriler	Temsili Yöntemler
Nokta İzleme (Point Tracking)	
<ul style="list-style-type: none">Deterministik (Rassal olmayan) yöntemler (Deterministic methods)	MGE izleyici [Salari ve Sethi 1990], GOA izleyici [Veenman vd. 2001].
<ul style="list-style-type: none">İstatistiksel yöntemler (Statistical methods)	Kalman süzgeci [Broida ve Chellappa 1986], JPDAF [Bar – Shalom ve Foreman 1988],

	PMHT [Streit ve Lugnbuhl 1994].
Çekirdek İzleme (Kernel Tracking)	
<ul style="list-style-type: none"> Şablon ve yoğunluk esaslı görünüm modelleri (Template and density based appearance models) 	Ortalama değer kayması (Mean-shift) [Comaniciu et. Al. 2003], KLT [Shi ve Tomasi 1994], Tabakalama (Layering) [Tao et. al. 2002].
<ul style="list-style-type: none"> Çoklu görüş görünüm modelleri (Multi-view appearance models) 	Öz izleme (Eigentracking) [Black ve Jepson 1998], SVM izleyici [Avidan 2001]
Siluet İzleme (Silhouette Tracking)	
<ul style="list-style-type: none"> Çevre çizgisi evrilmesi (Contour evolution) 	Durum uzay modelleri (State space models) [Isard ve Blake 1998], Değişken yöntemler (Variational methods) [Bertalmio et. al. 2000], Sezgisel yöntemler (Heuristic methods) [Ronfard 1994].
<ul style="list-style-type: none"> Şekil eşleştirme (Matching shapes) 	Hausdorff [Huttenlocher et. al. 1993], Hough dönüşümü [Sato ve Aggarwal 2004], Histogram [Kang et. al. 2004].

Şekil 3.3'de izleme yöntemlerinin sınıflandırılması gösterilmiştir. Çizelge 3.2'de her kategori en çok kullanılan uygulama yöntemleri ile birlikte tablanmıştır. Şekil 3.3 ve Çizelge 3.2'de görüldüğü gibi nesne izleme yöntemlerinin ana kategorileri; nokta, çekirdek ve siluet izlemedir.

3.4.1. Nokta izleme

Arka arkaya gelen çerçevelerde belirlenen nesnelere noktalar ile ifade edilirler. Bu noktaların bir araya getirilmesi, nesnenin pozisyon ve hareket bilgisini de içeren önceki durumuna bağlı olarak yapılır. Nokta izleme yaklaşımı, nesnenin her çerçevede belirlenmesini sağlayan harici bir yöntem ihtiyacı duyar.



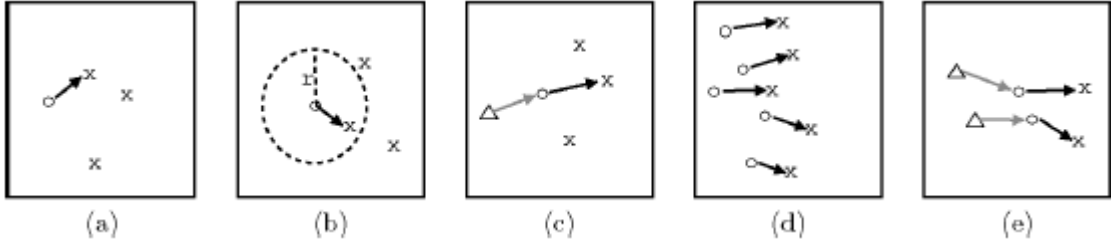
Şekil 3.4 Çok noktalı izleme

Nokta izleme yöntemlerini iki ana kategoriye ayırmak mümkündür. Bu kategoriler; nitelikli hareket sezgileyciler kullanan *deterministik (rassal olmayan) yöntemler* ve nesne ölçümleri ile belirsizlikleri değerlendiren *istatistiksel yöntemlerdir*.

3.4.1.1. Deterministik yöntemler

Deterministik yöntemler, $t - 1$ çerçevesindeki bir noktayı, t çerçevesinde karşılık gelen bir nokta ile ilişkilendirmek için hareket kısıtlamalarını kullanırlar. Noktalar arasındaki ilişkiyi kurmak amacıyla en yaygın olarak kullanılan kısıtlamalar şu şekildedir [45];

- *Yakınlık (Proximity)*. Bir çerçevedeki nesnenin yerinin takip eden çerçevede çok fazla değişmeyeceğini kabul eder (Şekil 3.5 (a)).
- *Maksimum hız (Maximum velocity)*. Nesnenin hareketi için maksimum bir hız tanımlar ve o hızla bağlantılı olan çember dışında olabilecek uyuşmaları göz ardı eder (Şekil 3.5 (b)).
- *Küçük hız değişimi (Small velocity change)*. Nesnenin hızında ve yönünde sert değişimler olmayacağını kabul eder (Şekil 3.5 (c)).
- *Alışılmış hareket (Common motion)*. Küçük bir komşuluk içerisindeki nesnelerin hızlarının birbirlerine yakın olduğunu kabul eder (Şekil 3.5 (d)).
- *Katılık (Rigidity)*. Nesnelerin katı olduğunu ve nesne üzerindeki iki noktanın yerlerinin değişmeyeceğini kabul eder (Şekil 3.5 (e)).
- *Yakınsal tekbiçimlilik (Proximal uniformity)*. Yakınlık ve küçük hız değişimi sabitlerinin birleşimidir.



Şekil 3.5 Hareket sabitleri. Yakınlık (a), maksimum hız (b), küçük hız değişimi (c), alışılmış hareket (d), katılık (e)

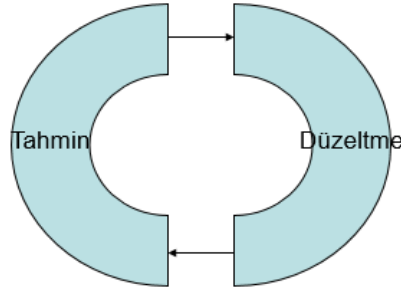
3.4.1.2. İstatistiksel yöntemler

Nesne izleme yapılacak görüntülerdeki sürekli gürültüler, nesnelere ani ve rastgele hareketleri nesne izlemenin problemleri arasında yer alır. İstatistiksel yöntemler bu problemleri, nesne durumunu tahmin etme aşamasında, olası tüm ölçümleri ve model belirsizliklerini göz önüne alarak çözer. İstatistiksel yöntemler, durum uzaylarını kullanarak nesnenin pozisyonu, hızı ve ivmesini modeller.

İstatistiksel yöntemleri kullanarak izleme yapan pek çok izleyici vardır. Bu yöntemlerden biri olan Kalman süzgeci [25], farklı algoritmalar ile birlikte de kullanılır. Kalman süzgeçleri, anlık olarak elde edilen gürültülü gözlem bilgisi ile önceki durumlara dayanarak yapılan tahminleri birleştirir. Diğer bir deyişle, kalman süzgeçleri gözlem yapar ve yaptığı gözlemlerden elde ettiği bilgiyi kullanarak, anlık durum belirsizliklerini ve gürültüyü giderir.

Tez çalışması kapsamında geliştirilen sistemde kalman filtresi iki amaç için kullanılmıştır. Bunlardan birincisi, nesne izleme sırasında oluşan gürültüyü gidererek nesnenin hareketinin yumuşatılması, ikincisi ise nesnenin önünün kapanması durumunda oluşacak belirsizliklerin giderilmesidir.

Kalman süzgeci, dinamik bir sistemde modelin önceki bilgileri ile şu anki giriş ve çıkış bilgilerini kullanarak durumlarını tahmin edebilen bir süzgeçtir. Kalman süzgeci, nesnelere sert ve ani hareketlerini süzerek yumuşatma ve sistemin ölçülemeyen durumlarını tahmin etme amacı ile kullanılır. Bu bağlamda iki adımdan oluşurlar. Bunlardan ilki tahmin / zaman güncelleme, diğeri ölçüm güncelleme ya da düzeltmedir. Tahmin yapılacak veriler bu iki yapı arasında sürekli bir döngü içerisinde çalışır ve tahminin tekrarlı yapı içerisinde gerçek değere yakınsaması sağlanır (Şekil 3.6).



Şekil 3.6 Kalman süzgeci çalışma şeması

Tahmin adımı, durum modelini kullanarak, değişkenlerin yeni durumlarını tahmin eder. Eşitlik (3.2)'de önceki durumun ve hata kovaryansının tahmin edilmesi işlemleri matematiksel olarak ifade edilmiştir:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (3.2)$$

$$P_k^- = AP_{k-1}A^T + Q$$

Düzeltilme adımı, şu anki gözlemler z^t 'yi kullanarak nesnenin durumunu günceller. Eşitlik (3.3)'de kalman kazancının hesaplanması, Eşitlik (3.4)'de ise ölçüme göre gerçek tahminin belirlenmesi ve hata kovaryansının güncellenmesi işlemleri matematiksel olarak ifade edilmiştir:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.3)$$

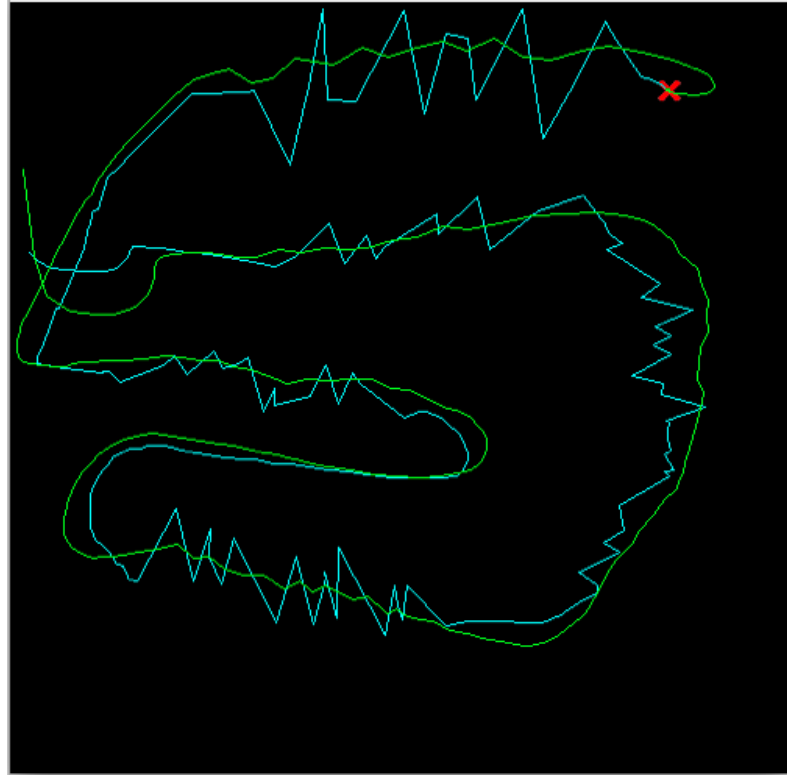
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.4)$$

$$P_k = (I - K_k H)P_k^-$$

Bu eşitliklerde, k anları, \hat{x}_k^- k anındaki durum tahminini, \hat{x}_{k-1} $k - 1$ anındaki durum tahminini, u_k sisteme dışarıdan uygulanacak kontrol sinyalini, A , B ve H sistemin transfer fonksiyonundan elde edilen durum matrislerini, P_k^- k anındaki hata kovaryansını, P_{k-1} $k - 1$ anındaki hata kovaryansını, K_k kalman kazancını, z_k k anındaki ölçüm değerini göstermektedir. Eşitlikler arasındaki R ve Q parametreleri, gürültü kovaryans matrisleridir. R ölçüm gürültüsünü temsil etmektedir ve küçük olması ölçümlerin doğruluğunu arttırmaktadır. Q ise işlem / süreç gürültüsüdür.

Kalman süzgeci uygulanan sistemde model tahmini, gözlem ile karşılaştırılır. Elde edilen fark, kalman kazancı (kalman gain) olarak bilinen bir çarpan ile ölçeklendirilir. Daha sonra sıradaki tahminleri iyileştirmek için modele bir girdi olarak geri besleme uygulanır. Böylece yöntem, gerçek bilinmeyen değerlere, tek bir ölçüme veya sadece model tahminlerine dayanarak elde edilebilecek tahminlerden daha yakın tahminler üretmeye yakınsar [60].

Şekil 3.7’de iki boyutlu bir uzayda hareket eden cismin kalman süzgeci ile takibi görülmektedir. Cismin izlediği yol açık mavi ile gösterilmiştir. Kalman fitresinin izleme sonucu ise yeşil renk ile gösterilmektedir. Görüldüğü gibi, kalman süzgeci cismin ani hareketlerini yumuşatmış ve sistemi daha kararlı bir hale getirmiştir.



Şekil 3.7 Kalman süzgeci örneği

3.4.2. Çekirdek izleme

Çekirdek, nesnenin şekil ve görünümünü ifade eder. Örneğin; histogram ile ilişkili dikdörtgen bir şablon veya eliptik bir şekil çekirdek olabilir. Nesnelere, arka arkaya gelen çerçevelerde çekirdeğin hareketinin hesaplanması ile izlenir. Bu hareket genellikle öteleme, dönme ve afin gibi parametrelili dönüşümlerdir.

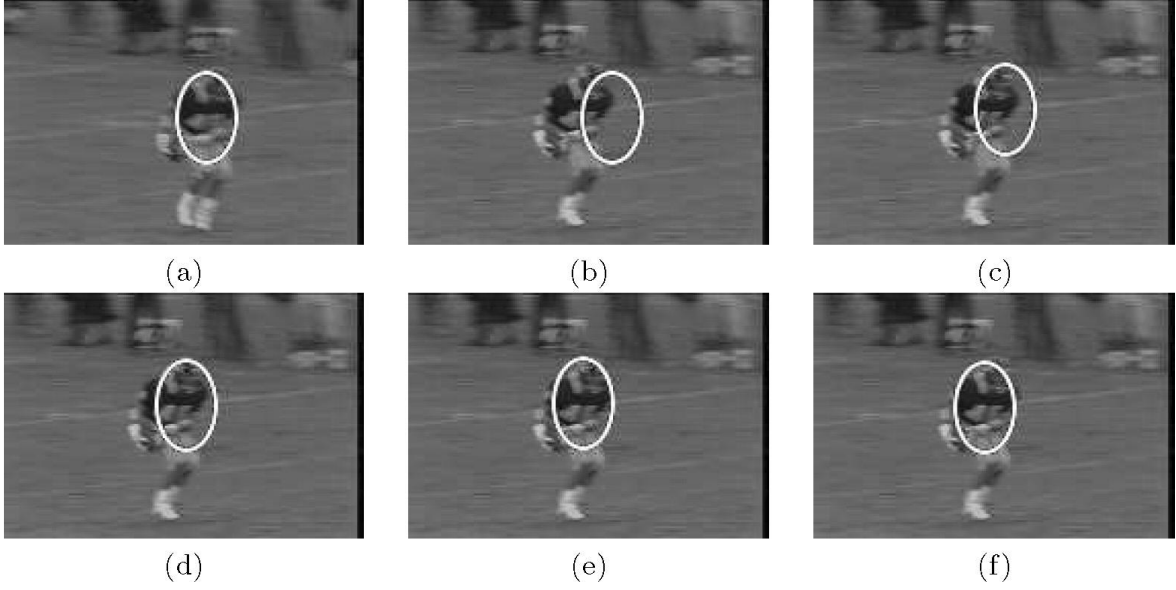
Çekirdek izleme algoritmaları, kullandıkları gösterim modelleri, izledikleri nesne sayısı ve nesne hareketini tahmin etme yöntemleri ile diğer izleyicilerden ayrılırlar. Çekirdek izleme yöntemlerini iki ana kategoriye ayırmak mümkündür. Bu kategoriler; ismini kullandığı görüntüleme modelinden alan *şablon ve yoğunluk esaslı görünüm modeli* ve *çoklu görüş görünüm modelleridir* [45].

3.4.2.1. Şablon ve yoğunluk esaslı görünüm modeli

Şablon ve yoğunluk temelli gösterim yöntemi, basit ve hesaplanması kolay olduğundan özellikle gerçek zamanlı izleme gereksinimi olan sistemlerde kullanılır. Şablon kullanılarak yapılan gösterimde genellikle görüntünün yoğunluk ve renk özelliklerinden yararlanır. Şablon kullanımının yanı sıra, dikdörtgen veya elipsoid şeklindeki bölgeler içerisinde kalan piksellerin renk histogramları da izleme için kullanılabilir [61].

Renk histogramları kullanan ve hızlı olması nedeniyle izleme algoritmalarının çoğunluğunda tercih edilen, yoğunluk esaslı görünüm modeli uygulamalarından biri ortalama değer kayması izleyicisidir (Mean-Shift) [62]. Ortalama değer kayması izleyicisi, Q nesnesinin histogramının bir benzerini, nesnenin bulunduğu varsayılan bölge P içerisinde tekrarlı olarak karşılaştırma yaparak arar. Her tekrarda, ortalama değer vektörü histogram benzerliği artacak şekilde hesaplanır. Bu işlem yakınsama gerçekleşene kadar devam eder. Yakınsamanın gerçekleşmesi genellikle 5 – 6 tekrar gerektirir. Benzerliğine bakılacak olan histogram, nesne üzerinden alınacak bir çekirdek bölgeden elde edilir [45]. Şekil 3.8’de ortalama değer kayması adımları gösterilmiştir. $t - 1$ anında alınmış bir görüntüde izlenecek olan nesne belirlenmiş ve nesnenin histogram özellikleri çıkarılmıştır (Şekil 3.8 (a)). t anında nesne hareket etmiş ve izleme algoritmasının başlangıç değeri $t - 1$ anındaki pozisyonda kalmıştır (Şekil 3.8 (b)). t anında elde edilen görüntü üzerinde ortalama değer kayması tekrarları yapılmış ve nesne bulunarak t çerçevesinde izleme tamamlanmıştır (Şekil 3.8 (c, d, e,f)).

Şablon ve yoğunluk esaslı görünüm modeli Mean-Shift izleyicisinin, geliştirilmiş versiyonu olan ve Bölüm 5.2.3’de bahsedilen CAMSHIFT izleyicisi, tez çalışması kapsamında geliştirilen sistemde kullanılan nesne izleme yöntemidir.



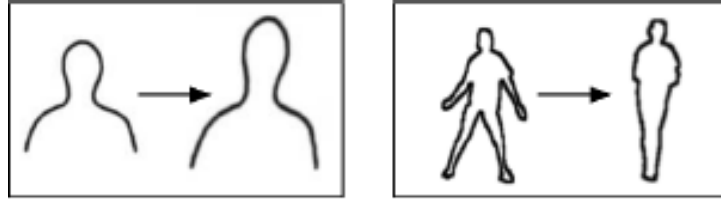
Şekil 3.8 Ortalama değer kayması tekrarları. $t-1$ anında hesaplanmış nesne pozisyonu (a), bir önceki nesne pozisyonu başlangıç değeri alınan t anındaki çerçeve (b), ortalama değer kayması tekrarları ile pozisyon güncelleme (c, d, e), t anındaki son pozisyon (f)

3.4.2.2. Çoklu görüş görünüm modelleri

Önceki izleme yöntemlerinde kullanılan histogramlar, şablonlar ve benzeri gösterim modelleri çevrimiçi olarak üretilmekteydi [45]. Başka bir deyişle, modellerin temsil ettiği nesne özellikleri en son gözlemler kullanılarak elde edilmekteydi. Nesnelere farklı açılardan bakıldıklarında farklı şekillerde görünebilirler. İzleme esnasında, izlenen nesnenin hareketi sebebiyle görünümünde olabilecek dramatik değişimler, görünüm modelinin o nesne için çalışmayacak hale gelmesine ve buna bağlı olarak izlemenin kaybedilmesine sebep olabilir. Bu problemin ortadan kaldırılması için, izlenecek olan nesnenin farklı görünümleri çevrimdışı olarak öğrenilebilir ve izlemede kullanılabilir [63].

3.4.3. Siluet izleme

İzleme, nesne bölgesinin her çerçevede tahmin edilmesi ile yapılır. Siluet izleme yöntemleri, nesne bölgesinin içerisinde kalan şifrelenmiş bilgiyi kullanırlar. Bu bilgi, görünüm yoğunluğu ve genellikle ayırıt haritası şeklinde olan şekil modelleri formunda olabilir.



Şekil 3.9 Çevre çizgisi değişimi örnekleri

İzlenmesi istenilen nesnelerin, temel geometrik şekiller ile tanımlanamayan karmaşık şekilleri olabilir. Siluet temelli yöntemler, bu tarzdaki nesnelerin kesin tanımını yapmak için kullanılır. Siluet izleme yöntemlerini iki ana kategoriye ayırmak mümkündür. Bu kategoriler; nesne silüetini şu anki çerçevede arayan *şekil eşleştirme* ve bir başlangıç çevre çizgisini şu anki çerçevede yeni konumuna eviren *çevre çizgisi izlemedir* [45].

3.4.3.1. Şekil eşleştirme

Şekil eşleştirme uygulaması bakımından çekirdek izleme yöntemi olan şablon temelli gösterim ile izleme yöntemine benzer. Nesnenin silüeti ve ilişkili modeli şu anki çerçevede aranır. Arama işlemi, bir önceki çerçevede elde edilen nesne silüetinin benzerinin aranması ile gerçekleşir. Nesne silüetine dayalı bir izleme yöntemi olduğundan katı olmayan nesnelerin bu yöntem ile izlenmesi mümkün değildir. Görüş açısı ve aydınlatma değişimlerinin etkilerinin en aza indirgenmesi amacıyla, siluet modeli her çerçevede bulunan nesne ile yenilenerek güncel tutulur [64].

3.4.3.2. Çevre çizgisi izleme

Çevre çizgisi (contour) izleme yöntemleri, şekil eşleştirme yöntemlerinin aksine, önceki çerçevedeki bir başlangıç çevre çizgisini şu anki çerçevedeki yeni pozisyonuna tekrarlı bir şekilde evirir. Çevre çizgisi evirme işlemi, nesnenin bir kısmının bir önceki çerçevedeki nesne bölgesi ile örtüşmesini gerektirir. Çevre çizgisi evirme işlemi, durum uzay modellerinin kullanılması veya enerji fonksiyonunun kullanımı ile gerçekleştirilir [65].

4. KULLANILAN YAZILIM VE DONANIMLAR

4.1. C++

C++, Bell Laboratuvarlarında 1979 yılında geliştirilmeye başlanmıştır. C'yi kapsayan, genel amaçlı bir programlama dilidir. Emirli (Imperative), nesne yönelimli (object-oriented) ve genel programlama özelliklerine sahiptir. Sınıflar sayesinde yeni veri türleri yaratılabilir veya var olan türlerden yenileri türetilebilir. Düşük seviyeli bellek kullanımı için kolaylıklar sağlar. Sistem programlamada (gömülü sistemler veya işletim sistemi çekirdeği (kernel)) kullanılmak üzere, performans, verimlilik ve esnek kullanıma uygun olarak tasarlanmıştır.

Orta seviye bir programlama dili olan C++, yukarıda sayılan özellikleri ve düşük seviye (Assembler) dillere olan yakınlığı sonucu sağladığı yüksek performans ile gerçek zamanlı ve yoğun işlem yükü gerektiren uygulamalarda öne çıkar. Bu sebeple, yüksek seviye dillere göre (Örn. Matlab) daha fazla tercih edilir.

4.2. OpenCV

OpenCV (Open Source Computer Vision – Açık Kaynaklı Bilgisayarla Görü) açık kaynak kodlu bir bilgisayarla görü ve makine öğrenme yazılımı kütüphanesidir. BSD (Berkeley Software Distribution) lisansı ile lisanslanmıştır, ticari kullanım dahil herkese açık ve ücretsizdir. C ve C++ dilleri kullanılarak yazılan kütüphane, Linux, Windows ve Mac OS X gibi masaüstü işletim sistemleri ile Android, iOS ve BlackBerry 10 gibi mobil işletim sistemlerinde kullanılabilir [66].

OpenCV işlem verimliliği sağlamak amacıyla, özellikle gerçek zamanlı uygulamalar hedef alınarak geliştirilmiştir. Hız açısından optimize edilmiş olan OpenCV kütüphanesi çok çekirdekli işlemci avantajını kullanabilecek şekilde tasarlanmıştır [67].

Kütüphane içerisinde 2500'den fazla optimize edilmiş algoritma vardır. Buna, klasik ve modern tekniklerden oluşan, bilgisayarla görü ve makine öğrenme algoritmaları da dahildir. Bu algoritmalar, insan yüzlerinin belirlenmesi ve tanınmasında, nesnelerin tanımlanmasında, video görüntülerindeki insan hareketlerinin sınıflandırılmasında, kamera hareketlerinin izlenmesinde, hareketli nesnelerin izlenmesinde, nesnelerin 3 boyutlu modellerinin çıkarılmasında, yüksek

çözünürlüklü görüntülerin elde edilmesi amacıyla görüntü birleştirmede, görüntü veri tabanında benzer görüntülerin bulunmasında, fotoğraflarda oluşan kırmızı göz etkisinin kaldırılmasında, göz hareketlerinin izlenmesinde ve benzeri alanlarda kullanılabilirler. 47 bin üzerinde kullanıcısı olduğu ve 7 milyondan fazla yüklendiği düşünülen OpenCV özellikle şirketler, araştırma grupları ve hükümet organları tarafından kullanılmaktadır [67].

Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota gibi tanınan firmalar tarafından kullanıldığı bilinen OpenCV, İsrail tarafından sınır ihlali gözetiminde, Çin tarafından maden ocaklarının gözetiminde, Avrupa'da boğulma vakalarının tespitinde, Japonya'da insan yüzü tespitinde ve Dünya üzerinde fabrika ürünlerinin denetlenmesinde kullanılmaktadır [67].

OpenCV kütüphanesi modüler bir yapıya sahiptir. Her bir modül içerisinde, modüle özgü veya paylaşılan kütüphaneler mevcuttur. Bu modüller [68];

- Core: Temel veri yapılarını (IpImage, cvPoint, cvSize, cvMat, cvHistogram... gibi) ve diğer modüllerin kullandığı temel fonksiyonları bünyesinde barındırır.
- Imgproc: Doğrusal ve doğrusal olmayan görüntü süzme, geometrik dönüşümler, renk uzayı dönüşümleri, histogramlar ve benzeri özellikleri barındıran görüntü işleme modülüdür.
- Video: Devinim kestirimi, arka plan çıkarımı ve nesne izleme algoritmalarını barındıran video analiz modülüdür.
- Calib3d: Temel çoklu görü geometri algoritmaları, tek ve stereo kamera kalibrasyonları, nesne poz kestirimi, 3 boyutlu yeniden yapılandırma elemanlarını bünyesinde barındırır.
- Features2d: Özellik belirleyicileri, tanımlayıcıları ve karşılaştırmaları bünyesinde barındırır.
- Objdetect: Nesnelerin belirleyicileri ve yüz, göz, insan, araba gibi önceden tanımlı sınıfları bünyesinde barındırır.
- Highgui: Slider, form gibi OpenCV kütüphanesi içerisinde tanımlanmış pek çok nesneyi yaratabilmemizi sağlayan bir grafik arabirimi olmakla beraber, resim ve videoları kaydetmek, yüklemek ve hafızadan silmek için gerekli giriş / çıkış (I / O) fonksiyonları da içerir.

- Gpu: Farklı OpenCV modüllerindeki algoritmaların GPU ile hızlandırılmış hallerini bünyesinde barındırır.

Bu çalışmada; core, imgproc, video, features2d ve highgui modüllerinden yararlanılmıştır.

4.3. Arduino IDE

Arduino, kolay kullanımlı donanım ve yazılımı amaçlayan açık kaynaklı bir platformdur. Ivrea Interaction Design Institute tarafından, elektronik ve programlama deneyimi olmayan öğrenciler hedef alınarak, kolay kullanılan bir geliştirme ortamı olarak ortaya çıkmıştır. Zaman içerisinde daha büyük kitlelere ulaşarak, Internet of Things (IoT), giyilebilir cihazlar, 3B yazıcılar ve gömülü ortamlar gibi gelişmiş ve yaygın olan alanlarda kullanılmaya başlanmıştır [69].

Açık kaynaklı Arduino yazılımı, kod yazımını ve yazılan kodların Arduino kartlara yüklenmesini kolaylaştırır. GPL (General Public License) ile lisanslanmıştır, herkese açık ve ücretsizdir. Arduino IDE (Integrated Development Environment – Bütünleşmiş Geliştirme Ortamı), kodların yazılması ve geliştirilmesi için bir kod editörüne ve hazırlanan kodların derlenerek karta yüklenmesini sağlayan derleyiciye sahiptir. Java programlama dili ile geliştirilen Arduino IDE, Windows, Mac OS X ve Linux ortamlarında çalışabilir. Arduino IDE, C ve C++ dillerini destekler ve bu diller ile yapılan çalışmaları derleyerek karta yükleyebilir [70].

4.4. Logitech® HD Pro Webcam C920

Tez çalışmasında geliştirilen sistem, özellikle görüntü işleme uygulamalarında yaygın olarak kullanılan, Logitech® HD Pro C920 web kamera ile donatılmıştır. Bu kamera, yüksek seviye bir tüketici ürünü olup optik keskinlik bakımından alanındaki en ileri seviye cihazlardandır. Windows üzerinde çalışabilen bir sürücüye sahiptir ve kendi yazılımı aracılığıyla objektif ayarları yapılabilir. Bu sayede, ışığın farklı durumlarına adapte olabilir ve lazer ışınının düştüğü noktanın bulunmasında kullanılan, Bölüm 5.2.1'de anlatılan algoritmanın kullanılmasına olanak sağlar.



Şekil 4.1 Logitech® HD Pro C920 web kamera

Logitech® HD Pro C920 web kameranın teknik özellikleri şu şekilde sıralanabilir [71];

- Full HD 1080p (1920 x 1080 piksel) görüntü alabilme imkanı,
- H.264 görüntü sıkıştırma ile yüksek çözünürlükteki görüntüyü USB 2.0 üzerinden 60fps olarak aktarabilme,
- Otomatik zayıf ışık düzeltme özelliği ile loş ışıkta bile net görüntüler yakalayabilme,
- Full HD, beş bileşenli Carl Zeiss® cam objektif çok keskin ve net görüntüler yakalarken, birinci sınıf 20 adımlı otomatik netleme özelliği, tutarlı yüksek tanımlılık sağlamak için sorunsuz ve hassas ayar yapar.

4.5. PhantomX Robot Turret

Tez çalışmasında geliştirilen sistemin hareketli hale getirilmesi için kullanılan PhantomX AX-18A Robot Turret, amatörler, robot bilimciler ve araştırmacılar için fiziksel programlamaya başlamayı kolaylaştırmak amacıyla, Trossen Robotics tarafından geliştirilmiş, yüksek performanslı bir Pan / Tilt platformudur. Kurulumu oldukça basit olan bu kit, gücünü ArbotiX - M Robocontroller kontrolcüden ve AX-12A / AX-18A Dynamixel aktüatörlerden alır. Sistem, yüksek performanslı, tamamıyla programlanabilen ve Arduino uyumlu bir Pan / Tilt platformu oluşturacak tüm donanıma sahiptir. Sisteme eklenebilecek olan harici aygıtlar ile farklı uygulamalar yapılabilir. Bu tez çalışmasında, sisteme web cam bağlanmış ve sistem Arduino IDE programı ile geliştirilen bir yazılım ile kontrol edilen iki adet Dynamixel AX-18A aktüatör ile hareketli hale getirilmiştir. Şekil 4.2'de PhantomX Robot Turret

görüntülenmiştir. Turret üzerindeki, ArbotiX - M Robocontroller kontrolcü ve Dynamixel AX-18A aktüatörler ile ilgili bilgi alt başlıklar altında verilmiştir.



Şekil 4.2 PhantomX Robot Turret

4.5.1. Arbotix – M robot kontrolcüsü

ArbotiX robot kontrolcüsü, küçük ve orta büyüklükteki robotlar için geliştirilmiş ileri düzeyde bir kontrol gereçidir. Vanadium Labs tarafından geliştirilmiştir. Güçlü bir AVR mikrodenetleyici, XBEE kablosuz radyo, çift motor sürücü ve 3-pinli servo başlıkları olan IO girişleri vardır. Mini anlamına gelen ArbotiX – M, orijinal ArbotiX'den %25 oranında daha küçük olacak şekilde tasarlanmıştır ve bazı ek özellikler ile güncellenmiştir.



Şekil 4.3 Arbotix-M Robot Kontrolcüsü

Arbotix – M robot kontrolcüsünün teknik özellikleri şu şekilde sıralanabilir [72] [73];

- 16MHz AVR mikrodenetleyici (ATMEGA644p),
- Biri servo kontrolüne ayrılmış iki adet seri port,
- 8'i analog giriş olarak çalışabilen 42 adet IO portu,
- 8 analog girişte ve 20 dijital IO üzerinde 3-pinli (toprak, besleme, sinyal) servo başlıkları,
- Bir çift 1A motor sürücü,
- 6 adet PWM portu,
- Arduino IDE ile programlanabilme ve Dynamixel AX-12A / AX-18A servoları sürmek üzere açık kaynak kütüphaneler.

4.5.2. Dynamixel AX – 18A servo motor

Robotis tarafından geliştirilen Dynamixel akıllı aktüatör serisi, çok eklemlili robotların yapımında, eğitimsel ve araştırma amaçlı kendin yap (Do It Yourself – DIY) projelerinde kullanılabilen hepsi bir arada (all in one) aktüatörlerdir. Hepsi bir arada yapı içerisinde, küçültme dişlisi (reduction gear), kontrolcü (controller), sürücü (driver) ve iletişim ağı (network) yapıları mevcuttur [74].



Şekil 4.4 Dynamixel aktüatör hepsi bir arada yapısı

Dynamixel AX – 18A aktüatörler, kendi hızını, sıcaklığını, pozisyonunu, voltajını ve yükünü kararlı çalışabilmek için geri besleme ile takip edebilir. Bununla birlikte, Dynamixel AX – 18A servo aktüatörlerde şaft pozisyonu korumak için kullanılan kontrol algoritması her bir servo motor için tek tek ayarlanabilir. Bu da kullanıcıya, motor hız ve güç yanıtını kontrol etme imkanı sunar. Tüm sensor kontrolleri ve pozisyon kontrolü, servo içerisinde yerleşik olarak bulunan bir mikrodenetleyici tarafından yapılır. Bu yapı, ana denetleyicinin diğer işlevleri yerine getirmesi için serbest kalmasını sağlar.

Dynamixel AX – 18A, TTL bağlantı sistemine sahiptir ve 3 pinli bir girişi vardır. Bu girişlerden biri data, biri toprak ve biri de besleme girişleridir. Dynamixel AX – 18A direk olarak bilgisayar üzerinden arayüzler yardımıyla kontrol edilebileceği gibi, Arduino benzeri kartlar ile de kontrol edilebilir.

Dynamixel AX – 18A, potansiyometre ile pozisyon ölçümü yapar ve 300° pozisyon kontrolü yapabilir. Tekerlek moduna getirilerek, tam tur hareket ettirilebilir ve hız kontrolü yapılabilir.



Şekil 4.5 Dynamixel AX - 18A aktüatör

Dynamixel AX – 18A aktüatörün teknik özellikleri şu şekilde sıralanabilir [75];

- Çalışma voltajı : 12V
- Saplanma torku (Stall torque) : 18,3 kg / cm, 1,8 Newton metre
- Yüksüz hız : 97 rpm
- Çözünürlük : 0,29°
- Çalışma Açısı : 300° veya sürekli tam tur
- Maksimum ve bekleme akımı : 2200mA ve 50mA
- Haberleşme hızı : 7343 bps ~ 1Mbps
- Pozisyon, sıcaklık, yük voltajı, giriş voltajı, PID geri beslemeleri.

5. YAPILAN ÇALIŞMALAR

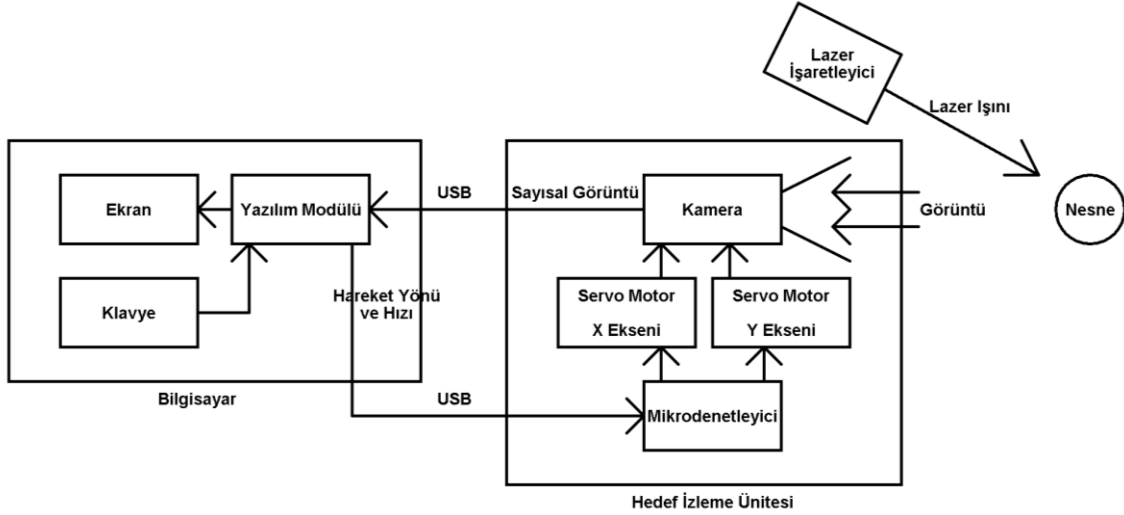
5.1. Sistemin Genel İncelemesi

Bu tez çalışmasında, birden fazla hareketli cismin serbestçe hareket ettiği üç boyutlu bir ortamda, bu cisimlerden lazer ışını ile anlık olarak işaretlenmiş olanına kilitlenen bir hedef izleme sistemi geliştirilmiştir. Nesne izleme, x ve y düzlemlerindeki bakış açısı iki servo motor ile ayarlanabilen taşıyıcı sisteme yerleştirilmiş bir video kameranın, hedef olarak seçilen cismin görüntü alanının merkezinde kalacak şekilde hareket ettirilmesiyle sağlanmıştır.

Yapılan çalışmada, izlenecek nesnenin seçilmesi ve nesnenin arka plandan ayrılması işlemlerinin yapılabilmesi için nesnenin ilk olarak lazer işaretleyici tarafından işaretlenerek sisteme gösterilmesi gerekir. İşaretleme anlık olarak yapılır, lazer ışını yok olduğunda sistem ışının çarptığı son nesneyi hedef olarak seçerek, hedef ile ilgili gerekli bilgileri çıkarıp arka plandan ayırma ve izleme işlemini yapmaya başlar. Sistemin çalışması için gerekli olan görüntü işleme teknikleri bir bilgisayar üzerinde uygulanmıştır. Bilgisayar üzerinde koşturulan algoritma, algoritmanın işleyiş yapısına uyması ve anlaşılmasının kolay olması amacıyla kendi içerisinde, lazer izleme, nesne belirleme ve nesne izleme olmak üzere üç bölümde incelenebilir. Kamerayı nesne izleme algoritmasından gelen pozisyon bilgisine göre, x ve y düzlemlerinde döndürerek nesnenin görüntü alanının merkezinde kalmasını sağlayacak mekanik düzeneği ise hedef izleme ünitesi adı altında ayrı bir bölümde incelemek mümkündür.

Şekil 5.1’de sistemin blok şeması görülmektedir. Blok şema üzerinde sistemin iki bileşeni olan bilgisayar ve hedef izleme ünitesi gösterilmiştir. Harici bir kaynak olan lazer işaretleyici ile izlenecek olan cisim işaretlenir. Hedef izleme ünitesi üzerinde yer alan kamera, görüntüleri alarak USB veri yolu üzerinden bilgisayarda koşturulmakta olan algoritmaya girdi olarak verir. Kullanıcı bilgisayar üzerindeki klavye yardımı ile bazı ayarları yapabilir. Kamera görüntüsü bilgisayar ekranında görüntülenmektedir. Nesne, izleme sırasında monitör üzerinde işaretlenmiş bir şekilde görülmektedir. Nesne izleme algoritmasının çıkışı olan hareket yönü ve hız bilgileri, yine USB veri yolu üzerinden hedef izleme ünitesine gönderilir. Hareket yönü ve hız bilgilerini alan mikrodenetleyici, x ve y eksenlerinde hareketi sağlayacak

olan servo motorları döndürerek kameraya hareket kazandırır. Bu sayede izlenen cisim her zaman kamera görüntüsünün merkezinde tutulur ve izleme işlemi fiziksel olarak da yapılmış olur.



Şekil 5.1 Sistemin blok şema gösterimi

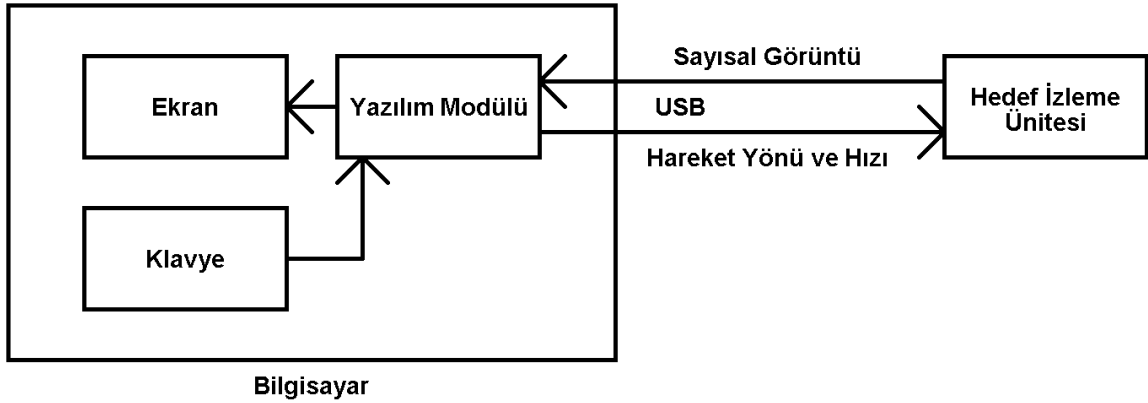
Şekil 5.2'de sistemin nesne izleme sırasındaki görüntüsü görülmektedir. Bu görüntüde, bilgisayar ve hedef izleme ünitesi ayrı ayrı görülmektedir. Bilgisayar ekranındaki görüntü üzerinde, izlenmekte olan nesne kırmızı kare içerisinde alınarak kullanıcıya gösterilmektedir. Ekran üzerinde, izlenen nesnenin histogramı, histogramın oluşturulmasında kullanılan, nesne içerisinde kalan en büyük kare alan ve kamera ayarları da görülmektedir.



Şekil 5.2 Sistemin nesne izleme sırasında görünümü

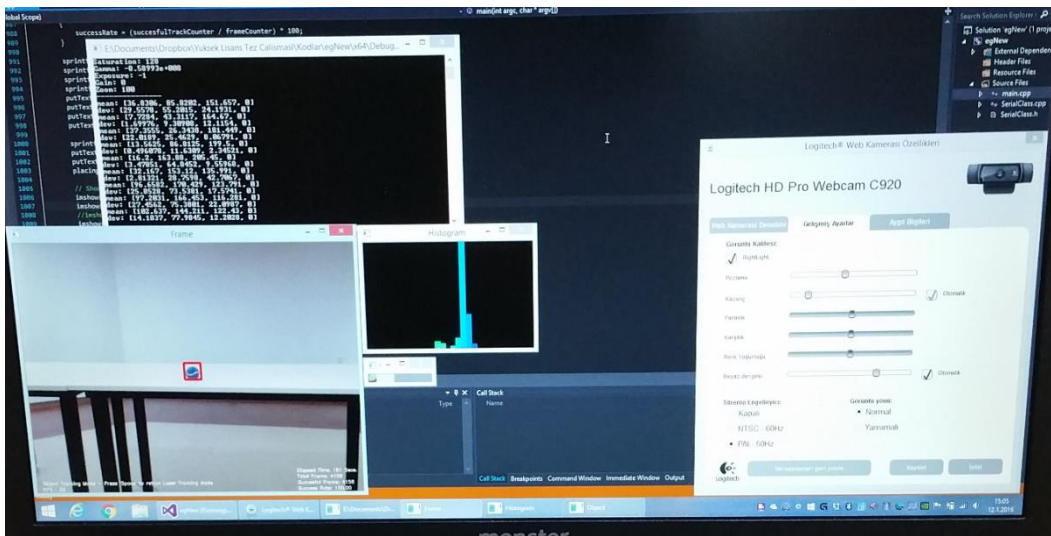
5.2. Bilgisayar

Bilgisayar, sistemdeki bütün hesaplamaların yapıldığı birimdir. Kullanıcı ile etkileşim bu birim üzerinde gerçekleşir. Bilgisayarın blok şema gösterimi Şekil 5.3'de ekran görüntüsü ise Şekil 5.4'de verilmiştir. Bilgisayar, hedef izleme ünitesinden gelen görüntüleri USB veri yolunu kullanarak alır. C++ dili ve OpenCV kütüphanesi kullanılarak yazılmış olan algoritmayı kullanarak görüntüyü işler. İşlenmiş olan görüntü, bilgisayar üzerindeki ekran aracılığı ile kullanıcıya gösterilir. Kullanıcı değiştirmek istediği bazı parametreleri bilgisayar üzerindeki klavye yardımı ile değiştirebilir. Bilgisayar, işlediği görüntülerde izlediği nesnenin konum bilgisini hesaplar, nesneyi kamera görüş alanının merkezinde tutmak için gerekli olan hareketin yön ve hızını USB veri yolunu kullanarak hedef izleme ünitesine gönderir.



Bilgisayar

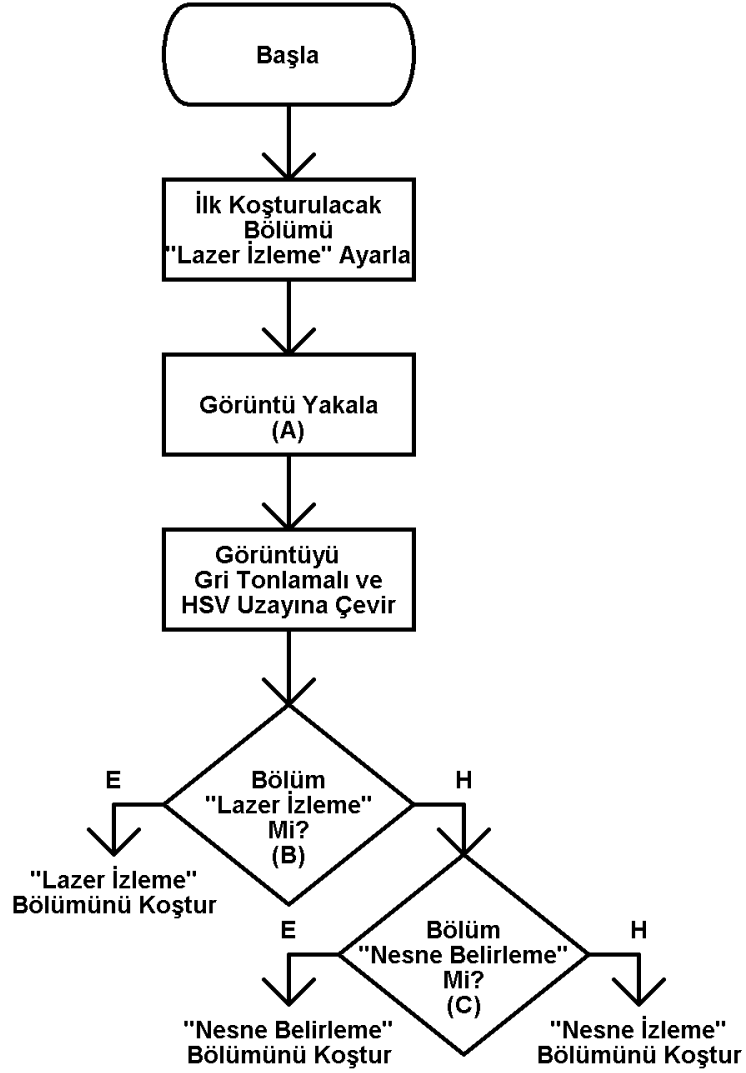
Şekil 5.3 Bilgisayar blok şema gösterimi



Şekil 5.4 Bilgisayar ekran görüntüsü

Geliştirilen sistemin gerçek zamanlı izleme yapabilmesi, nesne izleme algoritmasının hareket halindeki nesnelerin hareketlerini kaçırmayacak kadar hızlı çalışmasına bağlıdır. Bilgisayar üzerinde koşturulan algoritma performans artışı sağlaması amacıyla, alınan her bir çerçevede yalnız birinin çalışacağı bölümlere ayrılmıştır. Bu şekilde, sistemin her bir çerçeve için algoritmanın o an gereksiz olan kısımlarını koşturması önlenmiştir. Bu bölümler, kullanım sırasına göre lazer izleme, nesne belirleme ve nesne izleme bölümleridir. Sistem çalışmaya başladığında, lazer ile işaretleme işlemi yapılanaya kadar algoritmanın yalnızca lazer izleyici kısmını çalıştırır. Lazer ile işaretleme yapıldıktan sonra, takip eden çerçevede yalnızca nesne belirleme bölümü çalıştırılır ve artık gerek duyulmayan lazer izleme işleminden tasarruf edilmiş olur. Nesne belirleme işleminden sonra çalıştırılacak olan ve kullanıcı tarafından herhangi bir değişiklik istenmediği sürece, çalıştırılmasına devam edilecek olan bölüm nesne izleme bölümüdür. Sistem nesne izleme yaparken, algoritma içerisinde lazer izleme ve nesne belirleme ile ilgili kısımları atlayarak işlem gücünden tasarruf eder.

Bilgisayar üzerinde koşturulan algoritma çalışmaya başladığında aktif olan algoritma bölümü lazer izlemedir. Kamera, sistemin başlatılması ile birlikte görüntü yakalamaya başlar ve yakaladığı görüntünün gri tonlama ve HSV renk uzayı dönüşümlerini yaparak algoritmanın ileriki bölümlerinde kullanılacak veriyi hazırlar. Algoritmanın her bölümünde ortak olan görüntü yakalama ve renk dönüşümü işlemlerini içeren görüntü yakalama kısmının akış diyagramı Şekil 5.5'de verilmiştir.



Şekil 5.5 Algoritma başlangıcı. Görüntü yakalama kısmı

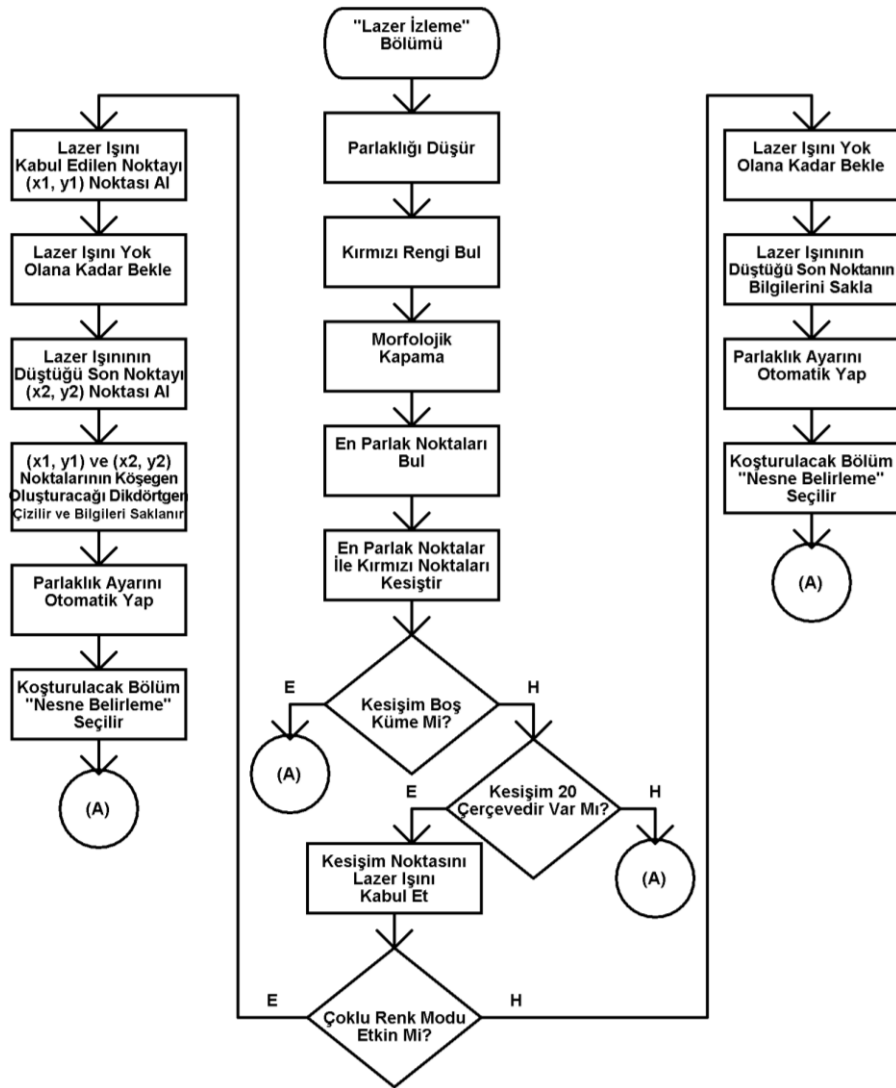
Şekil 5.5’de verilen görüntü yakalama kısmının akış diyagramı algoritmik olarak adım adım incelenirse;

1. Sistem çalışmaya, seri haberleşme için gerekli olan portların açılması, kameranın aktif hale getirilmesi gibi ilk ayarlamaları yaparak başlar.
2. Algoritmada ilk çalışacak bölüm olarak, lazer işaretlemenin yapılmasını sağlayan “Lazer İzleme” bölümü seçilir.
3. Kameradan görüntü yakalanır. Sistem, çalışması süresince her tekrarda bu adıma geri dönerek yeni görüntü yakalar. (Kameradan görüntü yakalama adımı, tezin ilerleyen bölümlerinde (A) adımı olarak adlandırılacaktır).
4. Yakalanan görüntünün gri tonlama ve HSV renk uzayı dönüşümleri gerçekleştirilir.

5. Algoritmanın bundan sonraki adımda hangi bölümü koşturacağı kontrol edilir. Bu kontrolün sonucuna göre, "Lazer İzleme", "Nesne Belirleme" veya "Nesne İzleme" bölümlerinden biri koşturulur.

5.2.1. Lazer izleme

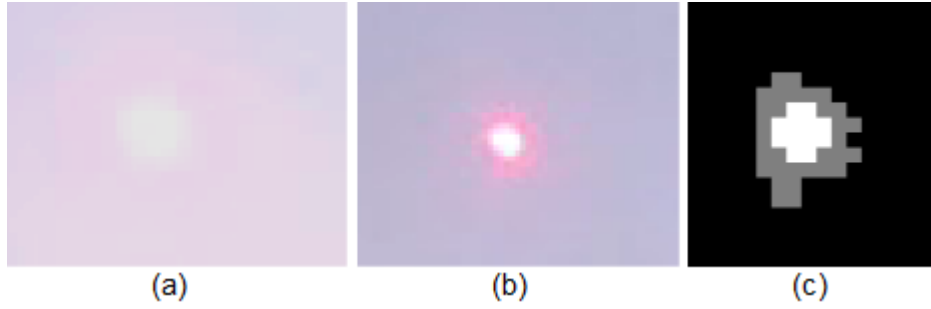
Lazer izleme, sistemin izleyeceği nesnenin seçimi için gerekli olan adımdır. Bu adımın amacı, lazer işaretleyici tarafından gönderilen lazer ışınının ortamın aydınlatmasından bağımsız olarak tespit edilmesidir. Sistemin algılaması gereken lazer ışını kırmızı renklidir. Sistemin, ışını en zor durumlarda algılayabilecek kadar hassas, farklı renk ve parlamalara yönelmeyecek kadar dayanıklı olması gerekir. Algoritmanın lazer izleme için kullanılan kısmının akış diyagramı Şekil 5.6'da verilmiştir.



Şekil 5.6 Lazer izleme bölümü akış diyagramı

Şekil 5.6'da verilen lazer izleme bölümü akış diyagramı algoritmik olarak adım adım incelenirse;

1. Lazer izleme bölümündeki amaç, lazer ışınının belirlenmesidir. Lazer ışınları kameralara beyaz bir nokta gibi görünür. Bunun sebebi, kameranın lazer ışınının yoğunluğu karşısında doyuma ulaşması ve rengi gösterebileceği en büyük değer olan beyaz ile göstermesidir. Bu problem kameranın pozlandırma süresinin değiştirilmesiyle çözülebilir [12].



Şekil 5.7 Kırmızı lazer ışınının görünümü. Yanlış pozlandırma süresi ile görünüm (a), doğru pozlandırma süresi ile görünüm (b), (b)'nin gri seviye gösterimi (c)

Şekil 5.7 (a)'da pozlandırma ayarı yapılmadığı durumda lazer ışınının kamera görüntüsü görülmektedir. Şekil 5.7 (b)'de ise pozlandırma ayarı sonrasındaki kamera görüntüsü görülmektedir. Lazer izleme bölümünün ilk adımında, lazer ışınının saptanabilmesi için kameranın pozlandırma süresi ve kazanç değerleri ile oynanarak görüntünün ortalama parlaklık seviyesi 15'in altına indirilmiştir.



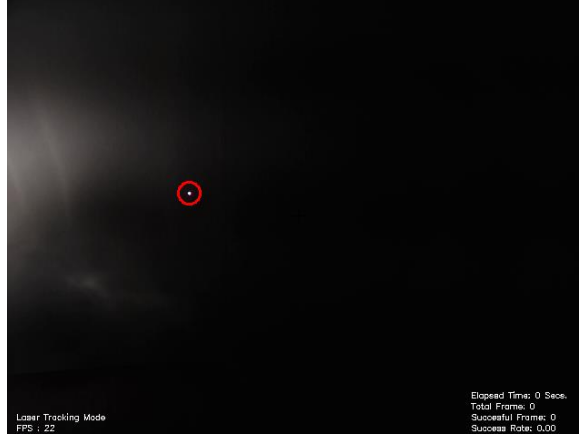
Şekil 5.8 Lazer izleme bölümünde, kameranın pozlandırma süresi ve kazanç değerlerinin değiştirilmesi ile görüntünün parlaklık seviyesinin düşürülmesi

2. Bir sonraki adım, sistemde işaretleme amacıyla kullanılacak olan kırmızı lazerin rengini bulmayı amaçlayan kırmızı rengi bulmadır. Bir cismin, kırmızı olarak görünebilmesi için R, G ve B değerleri içerisinde kırmızı değer olan R'nin diğer değerlerden büyük olması beklenir. Rengin kırmızı olarak görülebilmesi için, R değerinin belirli bir seviyenin üzerinde olması ve rengin beyaz olmaması için G ve B değerleri toplamının belirli bir seviyenin altında olması gerekir. Gereken koşulların matematiksel gösterimi Eşitlik (5.1)'de verilmiştir.

$$g(x, y) = \begin{cases} \text{Kırmızı}, & R > G \text{ ve } B, R > 180 \text{ ve } G + B < 280 \\ \text{Kırmızı değil}, & \text{diğer durumlar} \end{cases} \quad (5.1)$$

3. Eşitlik (5.1) kullanılarak bulunan noktalara morfolojik kapama uygulanarak, kırmızı bölge içerisinde kalan ancak eşitliğe uymadığı için boş bırakılan bölgeler doldurulur. Görüntü üzerindeki en kırmızı noktalar bu işlemlerin sonunda bulunmuş olur.
4. Kırmızı noktaların bulunmasından sonraki adım lazer ışınının düştüğü noktanın bulunmasıdır. Pozlandırma süresi doğru olarak ayarlanan bir kamerada, lazer ışınının düştüğü noktanın görüntü üzerindeki en parlak nokta olması beklenir. Bu sebeple sistem her çerçevede, en parlak nokta değerinin %88'i altında olan pikselleri 0'a eşitler. Bu işlem sonrasında görüntüden geriye yalnızca lazer ışını olabilecek alan ve çevresi kalır. Bu görüntü ile kırmızı noktaların bulunduğu görüntü kesiştirilir ve olası lazer ışını noktası bulunur. Kesişim kümesi boş küme olursa, sistem yeni bir görüntü yakalamak üzere görüntü yakala adımına geri döner.
5. Lazer ışını yapısı gereği fazla yayılmaz. Bu özellik kullanılarak kesişim görüntüsü üzerinde bir doğruluk testi yapılır. Bu testte, bir önceki adımda oluşturulan kesişim görüntüsündeki lazer ışını olması muhtemel piksellerin sayısının 120'yi geçmemesi beklenir. 120 piksel üzerinde nokta varsa, sistem bunun lazer ışınından ziyade, lazer özellikleri taşıyan başka bir cisim olduğunu var sayar.
6. Kesişim kümesinin 120 pikselden daha küçük olduğu durumun en az 20 çerçeve boyunca sürmesi beklenir. Sistem 20 çerçeve sonrasında noktanın kesin olarak

lazer ışını olduğuna karar verir. 20 çerçeve beklenmesinin sebebi, herhangi bir yansıma sebebiyle anlık oluşabilecek parlamaların lazer ışını olarak görülmemesini sağlamaktır.



Şekil 5.9 Parlaklığı düşürülmüş görüntü üzerinde lazer ışınının belirlenmesi

7. Lazer ışını belirlendikten sonra sistemin hangi modda çalıştığı kontrol edilir. Sistemin normal modda çalıştığı durumdaki algoritma adımları (a) ile belirtilmiştir. Karmaşık renkli nesnelerin izlenmesi amacıyla 'Çoklu Renk Modu' uygulanan algoritma adımları ise (b) ile belirtilmiştir. Çoklu renk modunda, sistem lazer ışınının düştüğü son noktadan ziyade, lazer ışını ile çizilen bir dörtgen içerisinde kalan alanı izlenecek nesneymiş gibi kabul eder. Şekil 5.10'da çoklu renk modunun çalışması gösterilmiştir. Şekil 5.10 (a)'da görülen farklı renkleri barındıran bir kitap, Şekil 5.10 (b)'de ki gibi lazer işaretleyici ile çizilen bir dörtgenin içinde kalan alanın renk bilgisi (Şekil 5.10 (c)) ile izlenir.
8. (a) Sistem ışının yok olmasını beklemeye başlar. Işın yok olduğu anda ışının düştüğü son noktanın pozisyon değerini, nesne belirleme bölümü için saklar.

(b) Sistem lazer ışını olarak belirlenen noktayı (x_1, y_1) noktası olarak belirler ve lazer ışınının yok olmasını beklemeye başlar. Işın yok olduğu anda ışının düştüğü son noktanın pozisyonunu (x_2, y_2) olarak belirler. Elde edilen (x_1, y_1) ve (x_2, y_2) noktaları köşegen olacak şekilde bir dikdörtgen oluşturulur (Şekil 5.10 (b)) ve dikdörtgenin pozisyon değeri, nesne belirleme bölümü için saklanır.
9. Parlaklık değeri düşürülen kamera ayarları otomatik moda getirilerek, nesne izleme süresince kameranın renk değişimlerine adapte olması sağlanır.

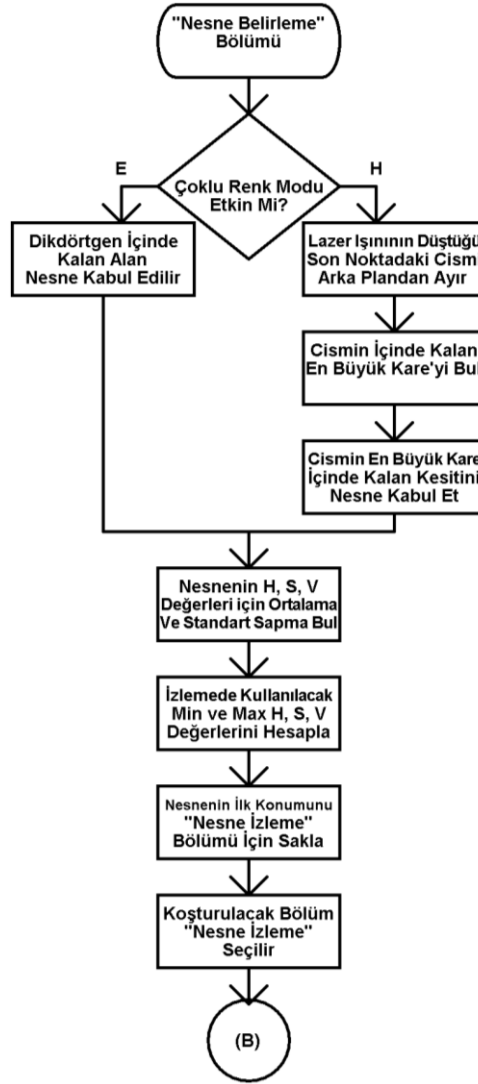
10. Bölüm “Nesne Belirleme” olarak ayarlanır ve görüntü yakalama adımına dönülerek, yeni yakalanan görüntü üzerinde nesne belirleme bölümü uygulanır.



Şekil 5.10 Çoklu renk modu gösterimi. Orijinal görüntü (a), çoklu renk modu ile takip edilmek üzere renk bilgisi çıkarılacak alan seçimi (b), seçilen alanın HSV gösterimi (c)

5.2.2. Nesne belirleme

Nesne belirleme, sistemin izleyeceği nesne ile ilgili özellikleri çıkarması için gerekli olan adımdır. Bu adımın amacı, lazer işaretleyici ile işaretlenen cisim arka plandan ayırarak izlenecek olan nesneyi belirlemek ve renk bilgilerini elde etmektir. Sistem yeni bir görüntü yakalayıp gri tonlama ve HSV renk uzayı dönüşümlerini yaptıktan sonra, çalışma bölümü olan “Nesne Belirleme Bölümü”nün gereklerini yapar. Algoritmanın nesne belirleme için kullanılan kısmının akış diyagramı Şekil 5.11’de verilmiştir.



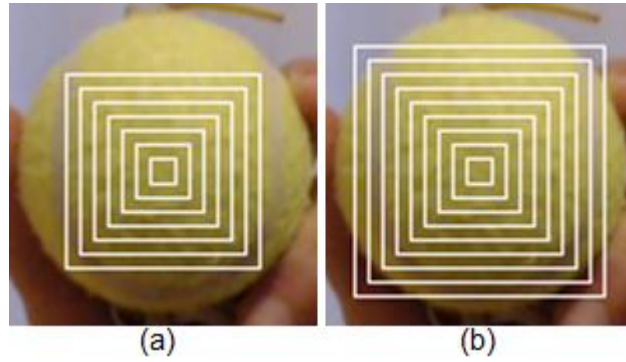
Şekil 5.11 Nesne belirleme bölümü akış diyagramı

Şekil 5.11’de verilen nesne belirleme bölümü akış diyagramı algoritmik olarak adım adım incelenirse;

1. Nesne belirleme bölümünün ilk adımında, sistemin hangi modda çalıştığı kontrol edilir. Sistemin normal modda çalıştığı durumdaki algoritma adımları (a) ile belirtilmiştir. ‘Çoklu Renk Modu’ uygulanan algoritma adımları ise (b) ile belirtilmiştir.
2. (a) Nesne belirleme bölümünün ikinci adımında, lazer izleme bölümünden gelen pozisyona Bölüm 2.11.4’de bahsedildiği gibi bölgesel tabanlı bölütleme yapılmak üzere tohum atılır. Bu tohumun 4-komşuluğuna bakılarak, hue (renk) değeri üzerinden bölgesel tabanlı bölütleme yapılır ve işaretlenen cisim arka plandan ayrılır.

(b) Lazer izleme bölümünden gelen dikdörtgen içerisinde kalan alan izlenecek nesne kabul edilir.

3. (a) Elde edilen cisim görüntüsüne morfolojik kapama uygulanarak, bölütlemenin tam yapılamadığı, cismin içinde boş kalan noktalar doldurulur.
4. (a) Elde edilen cismin merkezinde 10 x 10 piksel büyüklüğünde bir alanın hue (renk) ortalaması hesaplanır ve ortalamadaki sapma belirli bir değerin üstüne çıkana kadar kare büyütülerek işlem devam eder (Şekil 5.12 (a)). Kare cismin dışına taşıtığında ortalama büyük bir sapma meydana gelir (Şekil 5.12 (b)). Bu sapma meydana geldiğinde, bir önceki karenin boyutu esas alınarak cismin içerisinde kalan en büyük kare alan elde edilmiş olur. Bu alan, izlenecek olan nesneyi temsil eder.



Şekil 5.12 Cismin merkezine yerleştirilen 10 x 10 piksellik karenin büyütülerek renk ortalaması hesaplanması. Kare cismin içinde (a), kare cismin dışında (b)

5. İzlenecek nesnenin H (renk), S (doygunluk) ve V (parlaklık) değerlerinin ortalamaları ve standart sapmaları hesaplanır.
6. Minimum ve maksimum H, S, V değerleri için, Eşitlik (5.2) kullanılır.

$$(H, S, V)_{min} = (H, S, V)_{ort} - ((H, S, V)_{standart\ sapma} * 3) \quad (5.2)$$

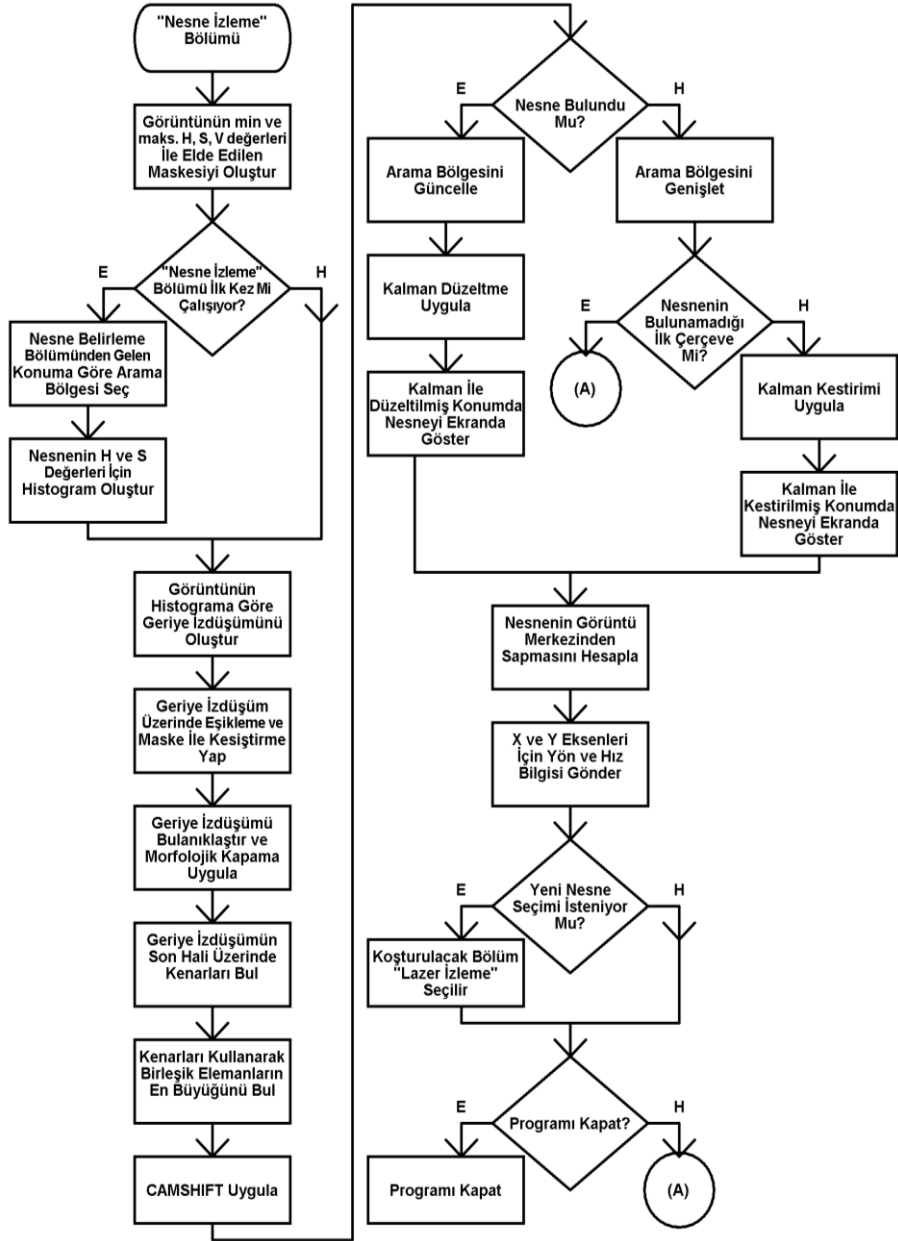
$$(H, S, V)_{maks} = (H, S, V)_{ort} + ((H, S, V)_{standart\ sapma} * 3)$$

7. Nesnenin pozisyon değeri ve nesne içerisinde kalan karenin pozisyonu nesne izleme bölümü için saklanır.

8. Bölüm “Nesne İzleme” olarak ayarlanır ve eldeki görüntüde izleme yapılmak üzere nesne izleme bölümüne geçilir.

5.2.3. Nesne izleme

Nesne izleme, sistemin nesneyi izlemesi ve nesnenin konumuna göre hareketin sağlanması için gerekli olan adımdır. Bu adımın amacı, nesne belirleme bölümünden gelen konum ve renk bilgilerini kullanarak nesneyi izlemek ve nesnenin kamera görüntü alanının merkezinde kalması için gerekli olan hareket yönü ve hızını hesaplayarak hedef izleme ünitesine göndermektir. Algoritmanın nesne izleme için kullanılan kısmının akış diyagramı Şekil 5.13’de verilmiştir.



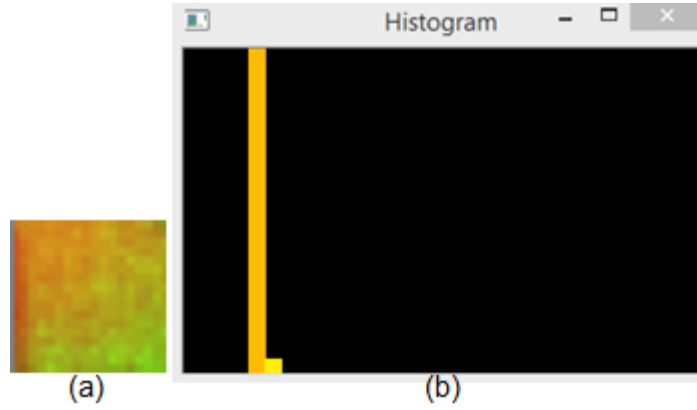
Şekil 5.13 Nesne izleme bölümüne akış diyagramı

Şekil 5.13’de verilen nesne izleme bölümü akış diyagramı algoritmik olarak adım adım incelenirse;

1. Nesne izleme bölümünün ilk adımında, nesne belirleme bölümünde elde edilen minimum ve maksimum H, S, V değerleri kullanılarak bir maske görüntüsü elde edilmiştir. Bu maskenin matematiksel ifadesi Eşitlik (5.3)’ de verilmiştir.

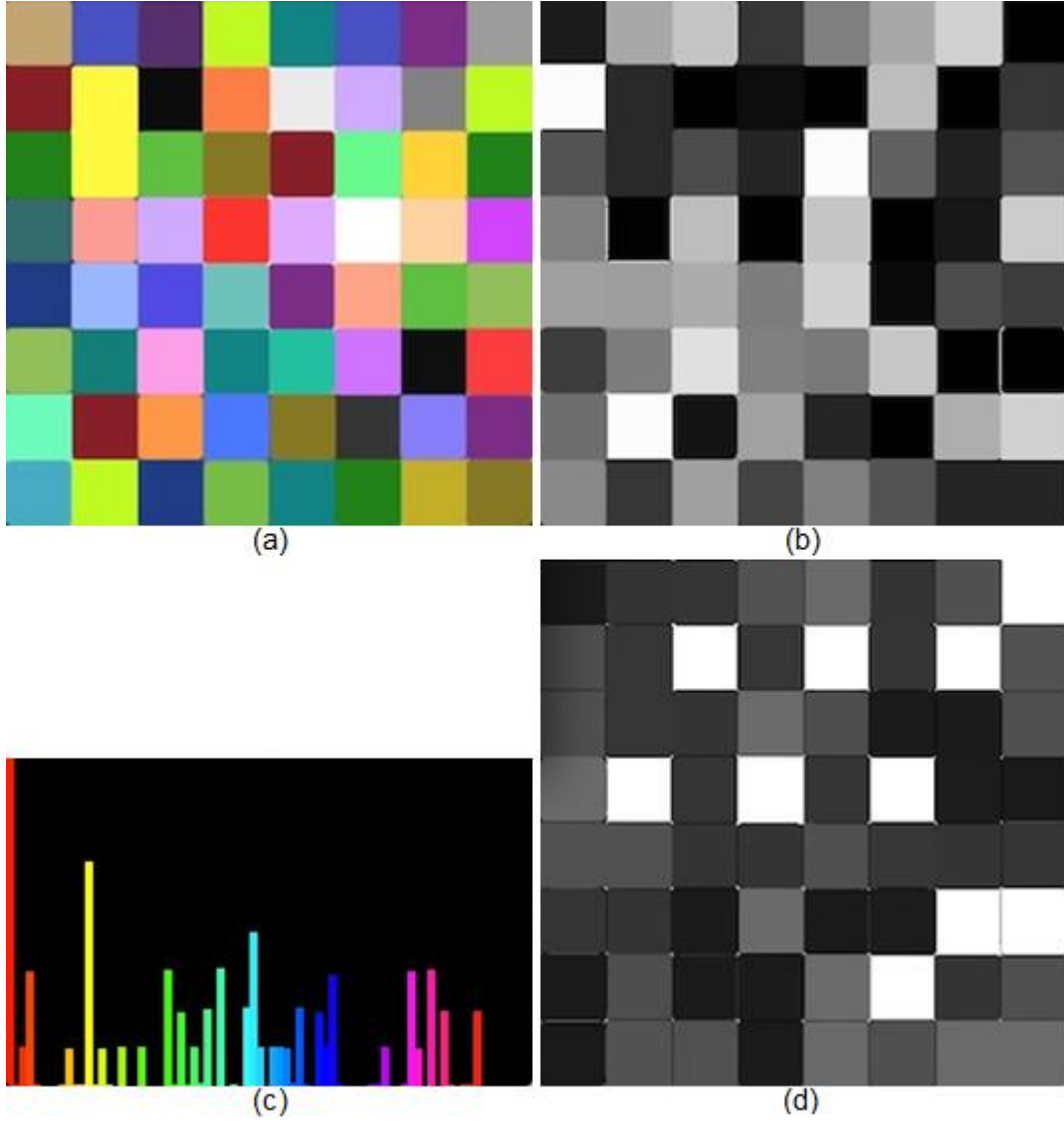
$$mask(x, y) = \begin{cases} 255, & (H, S, V)_{min} < g(x, y)_{H,S,V} < (H, S, V)_{max} \\ 0, & diğer durumlar \end{cases} \quad (5.3)$$

2. Bir sonraki adımda nesne izleme bölümünün ilk tekrarında olup olmadığı kontrol edilir. İlk tekrarda ise, nesne belirleme bölümünden gelen konum bilgisine göre arama bölgesi seçilir. Daha sonra nesnenin H ve S değerleri için histogram oluşturulur. Nesne izleme bölümü ilk tekrarında değilse, bu adımlar zaten yapılmış olacağından direkt olarak bir sonraki adıma geçilir. Şekil 5.14 (a)'da nesne belirleme bölümünden gelen, nesnenin içerisinde kalan en büyük kare alanın HSV gösterimi görülmektedir. Şekil 5.14 (b)'de ise Şekil 5.14 (a)'nın histogramı görülmektedir.



Şekil 5.14 Nesne olarak seçilen bölgenin HSV renk uzayında gösterimi (a), nesnenin histogram gösterimi (b)

3. Nesnenin histogramı elde edildikten sonra her çerçevede elde edilen görüntünün bu histograma göre geriye izdüşüm (back projection) görüntüsü oluşturulur. Geriye izdüşüm, nesne histogram dağılımının görüntüde en yüksek benzerlik sağladığı yerleri gösteren bir gösterimdir. Başka bir deyişle, geriye izdüşüm renk özelliklerinden yararlanılarak elde edilen bir özelliğin, Bölüm 3.4.2'de bahsedilen çekirdek izleme metodu kullanılarak başka bir görüntüde bulunmasıdır [76]. Geriye izdüşümün elde edilişi Şekil 5.15'de gösterilmiştir. Buna göre, geriye izdüşüm renge göre histogram çıkarıldıktan sonra, görüntü içerisindeki her pikselin renk değerinin, o rengin histogramdan elde edilen ağırlığı ile değiştirilmesi sonucunda elde edilir.



Şekil 5.15 Geriye izdüşüm. Renkli mozaik görüntü (a), (a) görüntüsünün hue kanalı (b), (b)'nin histogramı (c), histogramın görüntü içerisinde benzerlik sağladığı noktalar, geriye izdüşüm (d) [77]

4. Geriye izdüşüm görüntüsü üzerinde eşikleme yapılarak, belirli bir benzerlik seviyesinin altında kalan pikseller sıfıra eşitlenir. Benzer bir şekilde, algoritmanın önceki adımlarında bulunan maske ile kesiştirme yapılarak maske değerleri dışarısında kalan renk değerleri de sıfıra eşitlenir. Bu işlem, geriye izdüşüm görüntüsü üzerinde, gereksiz işlem yükü oluşturabilecek piksellerin sıfırlanması içindir.
5. Bir sonraki adımda geriye izdüşüme bulanıklaştırma ve morfolojik kapama işlemleri uygulanır. Bu işlemler, geriye izdüşümü kenar bulma algoritmasına hazırlar. Şekil 5.16'da geriye izdüşüm örneği görülmektedir. Şekil 5.16 (a)'da

arka planı farklı renklerden oluşan bir ortamda, tenis topunun izlenmesi işleminin orijinal ekran görüntüsü görülmektedir. Şekil 5.16 (b)'de ise bu görüntünün tenis topundan alınan renk histogram bilgisine göre oluşturulan bulanıklaştırma ve morfolojik kapama uygulanmış geriye izdüşüm görüntüsü görülmektedir.



(a)



(b)

Şekil 5.16 Geriye izdüşüm örneği. Orijinal görüntü(a), geriye izdüşüm görüntüsü (b)

6. Geriye izdüşüm üzerinde kenarlar bulunur.

7. Bulunan kenarlar kullanılarak birleşik elemanların en büyüğü bulunur ve bu en büyük eleman ile yeni bir görüntü oluşturulur. Şekil 5.16 (b)'de görülen geriye izdüşüm görüntüsünde, izlenecek nesne ile benzer renk özellikleri gösteren arka planda kalmış cisimler de görülmektedir. Bu cisimlerin nesne izleme algoritmasını etkilememesi ve izlemenin doğru yapılması için, izlenecek olan nesne görüntüdeki en büyük alanı kapsıyormuş gibi varsayılarak birleşik elemanların en büyüğü bulunur. Şekil 5.16 (b) görüntüsü üzerinde birleşik elemanların en büyüğünü bulma işlemi uygulandığında Şekil 5.17'de ki görüntü elde edilir. Bu görüntüde yalnızca en büyük eleman olan izlenecek cisim görülmektedir. Arka plandaki sistemi etkileyebilecek gürültüler elenmiştir.

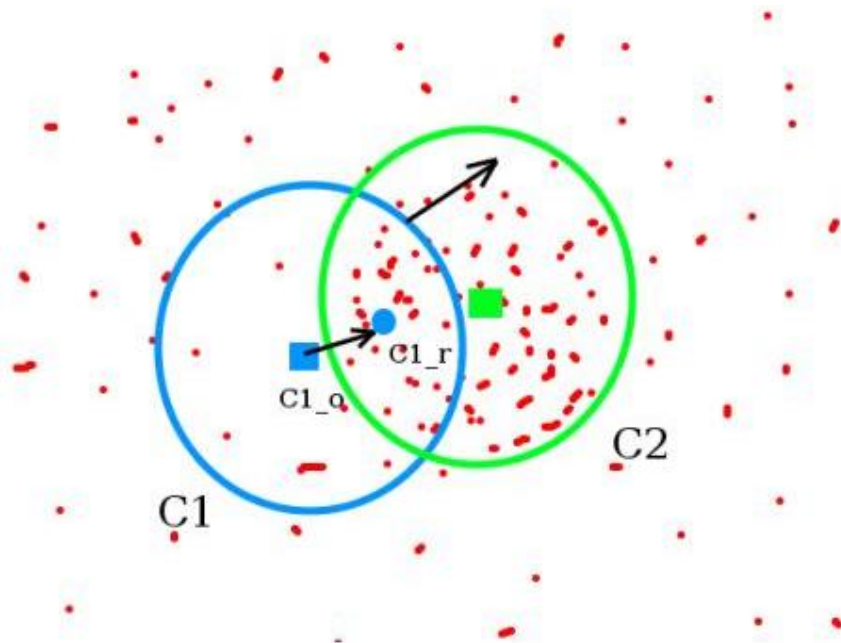


Şekil 5.17 Geriye izdüşüm görüntüsü üzerinde birleşik elemanların en büyüğünün bulunması

8. Oluşturulan görüntüye CAMSHIFT uygulanır.

CAMSHIFT [4] uygulaması Bölüm 3.4.2.1' de bahsedilen, şablon veya yoğunluk esaslı görünüm modeli kullanılan çekirdek izleme yöntemlerinden, Mean-Shift [54], [62] uygulamasının geliştirilmiş halidir. Mean-Shift uygulamasında, rastgele dağılmış noktalar bulunan veri seti içerisinde (histogram geriye

izdüşüm görüntüsü) küçük bir pencere (arama bölgesi) hareket ettirilerek pencere içerisinde maksimum nokta yoğunluğu sağlanmaya çalışılır. Bu işlem Şekil 5.18’de örneklenmiştir. Burada mavi renk ile gösterilen C1 dairesi başlangıç penceresidir. C1 dairesinin merkezi mavi kare (C1_o) ile gösterilmiştir. C1 penceresi içerisinde kalan noktaların ağırlık merkezi olan nokta ise mavi çember (C1_r) ile gösterilmiştir. Bu durumda mavi çember merkezi C1_r pozisyonuna gelecek şekilde kaydırılır ve aynı işlemler dairenin merkezi ile daire içerisinde kalan noktaların ağırlık merkezi çakışincaya kadar devam eder. Son pozisyon yeşil renkli C2 dairesi ile gösterilmiştir [78].



Şekil 5.18 Meanshift örneği [78]

Mean-Shift algoritmasının zayıf kaldığı nokta, arama penceresinin her zaman aynı boyutta olmasıdır. İzlenen cisimlerin boyutu hareketlerine bağlı olarak kameraya yaklaşmaları ve uzaklaşmalarına göre değişir. Meanshift uygulamasının pencere büyüklüğü sabit olduğundan, bu tip hareketli cisimlerin izlenmesinde yanlış sonuçlar oluşabilir (Şekil 5.19).



Şekil 5.19 Meanshift ile izlemede, cismin boyutunun değişimine bağlı problemin oluşumu [78]

Şekil 5.19'da görülen problemi çözmek amacıyla meanshift algoritmaları, sürekli uyarlamalı meanshift (CAMSHIFT) [4] algoritmalarına geliştirilmiştir. CAMSHIFT, algoritmaları başlangıç olarak meanshift algoritmasını uygularlar. Meanshift algoritması ile yakınsama yapıp arama penceresi sabitlendikten sonra, pencerenin büyüklüğü Eşitlik (5.4) kullanılarak güncellenir. Büyüklüğü değiştirilen pencere ile yakınsama gerçekleşene kadar meanshift uygulanır. Şekil 5.19'da karşılaşılan problemin CAMSHIFT çözümü Şekil 5.20'de verilmiştir [78].

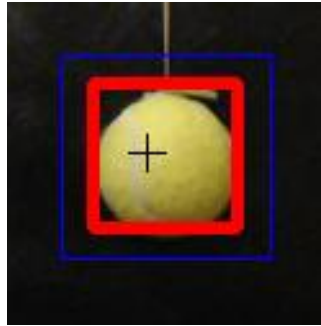
$$S = 2 \sqrt{\frac{\sum x \sum y P(x, y)}{256}} \quad (5.4)$$

$P(x, y)$ geriye izdüşüm görüntüsünü ifade etmektedir.



Şekil 5.20 CAMSHIFT ile izlemede, cismin boyutunun değişimine bağlı problemin çözümü [78]

9. CAMSHIFT uygulanmasından sonra nesnenin bulunup bulunmadığına bakılır. Nesne bulunmuş ise, arama bölgesi bir sonraki çerçevede kullanılmak üzere nesnenin bulunduğu yer ile güncellenir. Kalman düzeltme uygulanarak nesnenin hareketi yumuşatılır. Kalman düzeltmenin sonucunda, nesne ekranda gösterilir. Nesne CAMSHIFT uygulaması ile bulunamamış ise, arama bölgesi bir sonraki çerçevede kullanılmak üzere genişletilir. Eğer nesnenin bulunamadığı ilk çerçeve ise görüntü yakalama adımına dönülerek yeni görüntü üzerinde genişletilmiş arama bölgesi ile arama yapılır. Eğer nesnenin bulunamadığı ilk çerçeve değilse, nesnenin izlenemediği anlaşılır ve kalman kestirimi uygulanır. Kalman kestiriminin sonucu olan konum ekranda gösterilir. Şekil 5.21’de CAMSHIFT uygulaması sonrasında başarılı bir şekilde izlenen nesne görülmektedir. Nesne çevresindeki geniş mavi dörtgen nesnenin arandığı bölgeyi, nesneye tam oturan kırmızı dörtgen ise izleme sonucunu göstermektedir.



Şekil 5.21 Nesnenin CAMSHIFT uygulaması ile izlenmesi. Mavi dörtgen arama bölgesini, kırmızı dörtgen ise izlenen nesneyi göstermektedir.

10. Nesnenin konumunun kalman düzeltmesi veya kalman kestirimi ile belirlenmesinden sonra, nesnenin görüntü merkezinden ne kadar saptığı hesaplanır. Sapma hesaplaması Eşitlik (5.6) kullanılarak yapılır.

$$S(x, y) = N(x, y) - G(x, y) \quad (5.5)$$

Burada $S(x, y)$ x ve y düzlemlerindeki sapmayı, $N(x, y)$ nesne merkezinin x ve y düzlemindeki konumunu, $G(x, y)$ ise görüntü merkezinin konumunu vermektedir. Yakalanan görüntüler 640×480 piksel çözünürlüklü olduğundan; $G(x, y) = (320, 240)$ 'dir.

11. Hesaplanan sapmaya göre, X ve Y eksenleri için hareket yönü ve hızı belirlenerek seri haberleşme ile hedef izleme ünitesine gönderilir.

Şekil 5.22’de 640 x 480 piksel çözünürlük ile elde edilen bir çerçevenin, sistemin hareket yönünün belirlenmesi amacıyla dört bölgeye ayrılmış hali gösterilmektedir. Çerçevenin orijin konumu (320, 240)’dır. İzlenen nesnenin bulunduğu bölgeye göre Eşitlik (5.5) kullanılarak hareketin yönü ve hızı belirlenebilir. Hareketin yönü S_x , x düzlemindeki sapma ve S_y , y düzlemindeki sapma olmak üzere şu şekilde belirlenir;

Cisim I. Bölgede ise;

$$S_x > 0 \text{ ve } S_y < 0$$

Kamera sağa ve yukarıya döndürülmelidir.

Cisim II. Bölgede ise;

$$S_x < 0 \text{ ve } S_y < 0$$

Kamera sola ve yukarıya döndürülmelidir.

Cisim III. Bölgede ise;

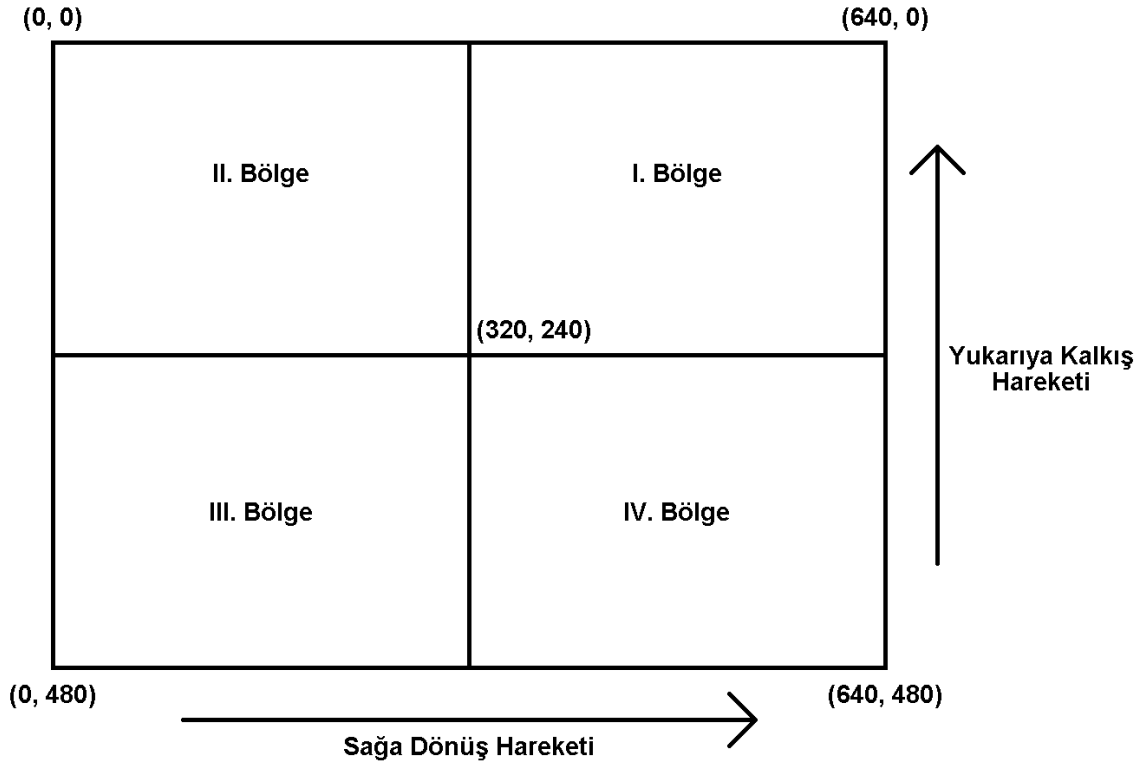
$$S_x < 0 \text{ ve } S_y > 0$$

Kamera sola ve aşağıya döndürülmelidir.

Cisim IV. Bölgede ise;

$$S_x > 0 \text{ ve } S_y > 0$$

Kamera sağa ve aşağıya döndürülmelidir.



Şekil 5.22 640 x 480 piksellik bir çerçevede, sistemin hareket yönünün belirlenmesi için kullanılan bölgeler

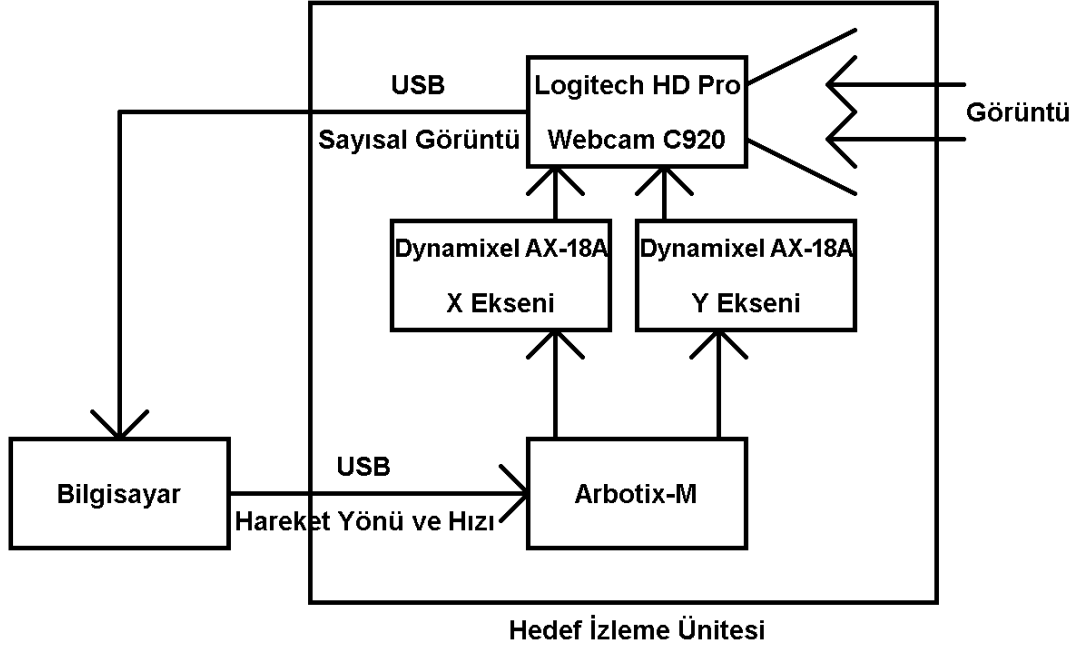
Hareketin hızının belirlenmesi, sapmanın büyüklüğü ile ilgilidir. İzlenen nesne orijinden uzaklaştıkça sapmanın büyüklüğü artar ve sistem hareketin hızını arttırarak nesneyi orijine yaklaştırmaya çalışır. Nesne orijinden ne kadar uzaksa sistem o kadar hızlanır, orijine yaklaştıkça hız azaltılır.

12. Kullanıcı tarafından bölüm değişimi istemi olup olmadığı kontrol edilir. Bölüm değişimi istenmiş ise, sistem bölümü "Lazer İzleme" olarak ayarlar ve izlenecek yeni cismin seçimi için lazer izleme adımını aktif eder.
13. Program son olarak, kullanıcı tarafından programın kapatılma istemi olup olmadığını kontrol eder. İstem varsa program kapatılır. İstem yok ise, program güncellenmiş verileri ile yeniden görüntü yakalama adımına döner ve nesneyi takip eden çerçevelerde izlemeye devam eder.

5.3. Hedef İzleme Ünitesi

Hedef izleme ünitesi, sisteme hareketin kazandırıldığı ünedir. Logitech® HD Pro C920 web kamera, kameraya hareketi sağlayan iki Dynamixel AX – 18A motor ve motor sürücü Arbotix – M bu ünite üzerindedir. Hedef izleme ünitesinin blok şema

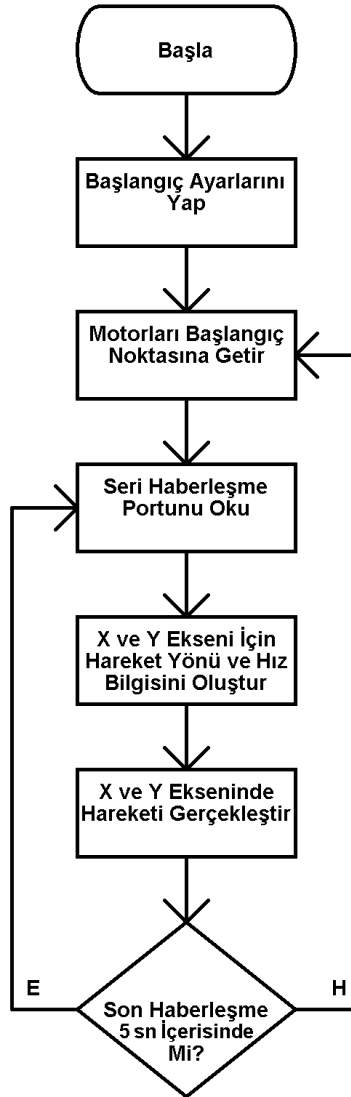
gösterimi Şekil 5.23'de gerçek görünümü ise Şekil 5.24'de verilmiştir. Hedef izleme ünitesi, USB veri yolunu kullanarak yakalanan görüntüleri bilgisayara gönderir. Farklı bir USB veri yolu üzerinden de hareket yönü ve hız bilgisini alır. Ünite, Arduino IDE geliştirme ortamında C++ dili ile yazılmış olan algoritmayı kullanarak sisteme hareket kazandırır. Hedef izleme algoritmasının akış diyagramı Şekil 5.25'de verilmiştir.



Şekil 5.23 Hedef izleme ünitesi blok şema gösterimi



Şekil 5.24 Hedef izleme ünitesi görünümü



Şekil 5.25 Hedef izleme ünitesi akış diyagramı

Şekil 5.25’de verilen hedef izleme ünitesi akış diyagramı algoritmik olarak adım adım incelenirse;

1. Ünite öncelikle motor ve seri haberleşme ayarlarının yapıldığı başlangıç ayarlamalarını yapar.
2. Ayarların yapılmasından sonra, motorları başlangıç konumlarına getirir.
3. Bir sonraki adımda seri haberleşmeden gelecek olan bilgi kontrol edilir.
4. Gelen bilgiye göre, X ve Y eksenini için hareketin yönü ve hızı ile ilgili bilgi oluşturulur.
5. X ve Y eksenindeki hareket gerçekleştirilir.
6. Son 5 saniye içerisinde seri haberleşme ile iletişim kurulup kurulmadığına bakılır. Kurulmuş ise, sistem seri haberleşmeden gelecek olan bilgiyi kontrol ederek hareket yönü ve hızının belirlendiği adımı tekrarlar. İletişim kurulmamış ise, bilgisayar ile bağlantının koptuğuna karar verilir ve motorlar başlangıç pozisyonuna alınarak seri haberleşmenin tekrar kurulması beklenir.

Hedef izleme ünitesi üzerinde kullanılan ve Bölüm 4.5.2’de bahsedilen Dynamixel AX-18A servo motorları pozisyon kontrollü çalıştırıldıklarında 300° çalışma açısına sahiptirler. Bu çalışma açısı motorlara gönderilen $[0, 1023]$ arası, 512 orta nokta olacak şekilde 1024 adım ile belirlenir. Bu durumda her bir adımın karşılığı $300^\circ/1024 = 0,29^\circ$ olur. Bilgisayar, nesne izleme modunda yaklaşık olarak 20 fps ile çalışır. Bu durumda her bir çerçevenin işlenmesi yaklaşık olarak $1000ms/20 = 50ms$ sürer. Bu bilgisayarın, hedef izleme ünitesine her 50 milisaniyede bir yeni hareket yönü ve hızı bilgisi göndermesi anlamına gelmektedir. Hedef izleme ünitesinin hareket hızı, görüntünün ani değişimlerden etkilenmemesi için maksimum olarak 16 adım ile sınırlandırılmıştır. Bu durumda, sistem 50ms süre içerisinde $16 \times 0,29^\circ = 4,64^\circ$ ’lik hareket yapabilir. Bu da 1 saniye içerisinde $92,8^\circ$ ’lik hareket kabiliyeti anlamına gelmektedir.

Hedef izleme ünitesinin y eksenindeki hareket kabiliyeti, ünitenin fiziksel yapısından dolayı kendine zarar vermemesi amacı ile yere paralel olduğu (0°)

durumdan -30° ve $+30^{\circ}$ olmak üzere toplamda 60° hareket edebilecek şekilde yazılımsal olarak sınırlandırılmıştır. Ünitenin x eksenindeki hareketi ise, kameranın sistemin ön tarafına bakmasını sağlamak amacı ile -45° ve $+45^{\circ}$ olmak üzere toplamda 90° hareket edebilecek şekilde sınırlandırılmıştır.

5.4. Sistemin Başarılı Çalışması İçin Gereken Ön Koşullar

Geliştirilen sistemde diğer tüm görüntü işleme tabanlı nesne izleme sistemlerinde olduğu gibi bazı ön koşulların belirlenmesine ihtiyaç duyulmuştur. Bunun nedeni, olabilecek izleme yanılgılarını en aza indirmektir. Söz konusu ön koşullar aşağıda verilmiştir:

- İzlenecek olan nesnenin, ortamın aydınlatma şartlarından fazla etkilenmeyen, ışığı yansıtmayan ve parlama yapmayan mat cisimlerden seçilmiş olması gerekir. Bunun sebebi nesnenin rengine bağlı izleme yapılmasıdır. Ortamın aydınlatma şartlarından fazlaca etkilenen, ışığı yansıtan ve parlama yapan cisimlerin renkleri hareketleri sırasında bu özelliklerinden ötürü bozulabilir. Bu da sistemin izleme sırasında, izlediği nesneyi kaybetmesine neden olabilir.
- Nesne izleme işleminin yapıldığı ortamın aydınlatmasının sabit olması gerekir. Bunun sebebi sistemin izleyeceği nesnenin özelliklerini çıkarma işlemini yalnızca bir kere, izlenecek nesnenin seçilmesi aşamasında yapmasıdır. Başlangıç durumundaki aydınlatma koşulları altında renk özellikleri çıkarılan bir cismin, aydınlatma koşulları değiştikçe görünümü ve renk özellikleri de değişir. Sistemde kullanılan Logitech® HD Pro C920 web kameranın otomatik zayıf ışık düzeltme özelliği ve 20 adımlı otomatik netleme özellikleri, aydınlatma değişimlerinden kaynaklanan problemi büyük ölçüde bastırır da, aydınlatmadaki büyük değişimler sistem performansını olumsuz etkiler.
- Nesne izleme sırasında, kamera merceği üzerine yoğun ışık gelmemesi gerekir. Bunun sebebi, bir üst maddede bahsedilen otomatik netleme özelliğidir. Kamera merceği üzerine yoğun ışık gelmesi durumunda, kamera otomatik olarak kazanç ve pozlama değerini düşürerek görüntüyü karartmaya çalışır. Bu durumda izlenecek olan nesnenin renk ve görünüm özelliklerinin değişmesine ve sistemin performansının olumsuz etkilenmesine sebep olur.

- İzlenecek olan nesne ile aynı renk özelliklerine sahip başka bir nesnenin veya arka plan görüntüsünün aynı sahnede olması durumunda, ikinci nesnenin izlenecek olan nesneden daha küçük olması gerekir. Bunun sebebi, Bölüm 5.2.3'de bahsedilen geriye izdüşüm üzerinde bulunan kenarlar kullanılarak birleşik elemanların en büyüğünün bulunması ve bu en büyük eleman ile oluşturulan görüntü üzerinde CAMSHIFT uygulanmasıdır. Yapılan bu işlem sayesinde izlenecek olan nesne ile aynı renkte fakat daha küçük olan, arka planda kalmış, nesnelere yok edilir. Ancak izlenecek olan nesne ile aynı renkli fakat daha büyük olan cisimlerin görüntüde bulunması durumunda sistem izlemesi gereken cismi arka plandaki cisim gibi görerek yanlış cismi izler.
- Lazer işaretleme aşamasında, ortam aydınlatmasının çok yoğun olmaması ve parlak cisimlerin kamera görüntüsünde yer almaması gerekir. Bunun sebebi, lazer ışınının düştüğü noktanın bulunması sırasında lazer ışınının parlaklık özelliğinin kullanılmasıdır. Bölüm 5.2.1'de bahsedildiği gibi, normal durumdaki lazer ışını kamerayı doyurarak beyaz çıktı vermesine sebep olur. Bu durumun önlenmesi amacıyla kamera pozlandırma süresi ve kazanç düşürülerek görüntü parlaklık seviyesi belirli bir değerin altına getirilir ve lazer ışınının görüntüdeki en parlak nokta olması sağlanır. Ortam aydınlatmasının çok yoğun olduğu durumda, pozlandırma süresi ve kazanç değerlerinin düşürülmesi görüntünün yeteri kadar karartılmasını sağlayamaz ve lazer ışını dışındaki noktalarında yine beyaza yakın renkli olması mümkün olur. Bu tip bir durumda lazer ışını görüntünün geri kalanından ayırt edilemeyebilir. Bu da sistemin lazer ışını belirleme konusundaki performansını olumsuz etkiler.
- Lazer işaretleme aşamasında, lazer ışını yok olduktan sonra sistemin o noktadaki cismi algılayıp, izleme için gerekli olan bilgileri çıkarması için yaklaşık 1 saniyelik kısa bir süreye ihtiyacı vardır. Cismin bu süre içerisinde yer değiştirmeden sabit kalması gerekir. Bunun sebebi, lazer ışını belirleme sırasında kamera ayarları ile oynanarak görüntünün karartılmasıdır. Lazer ışını yok olduktan sonra kamera ayarları tekrar otomatik haline getirilir. Kameranın görüntüyü netleştirmesi ve izlenecek nesne ile ilgili doğru bilgileri çıkarabilecek hale gelmesi için belirli bir süre geçmesi gerekir. Bu sürenin sonunda, sistem lazer ışınının son düştüğü noktadaki cismin bilgilerini çıkarır. Yaklaşık 1 saniye

süren bu bekleme aşamasında cisim yer değiştirirse, nesne olarak özellikleri çıkarılacak cisim yanlış seçilir ve izleme işlemi yanlış yapılır. Bu durum, lazer ışınının düştüğü son noktanın cismin sınırları dışında kalacak kadar hareket olması halinde gerçekleşir. Bunun dışında gerçekleşen ufak hareketlerin sistem performansına etkisi yoktur.

5.5. Sistemin Performans Analizi

Geliştirilen nesne izleme sistemi, farklı şartlar altında, farklı özelliklerdeki nesnelere izleme başarısına göre performans analizine tabi tutulmuştur. Performans analizindeki farklı şartların her biri ayrı durumlar olarak incelenmiş ve sistemin bu durumlardaki başarısı incelenmiştir. Yapılan incelemelerde, nesne izleme süresince alınan toplam çerçeve sayısı, kaç çerçevede izlemenin gerçekleştirildiği, kaç çerçevede sistemin izlenecek olan nesneyi bulamadığı ve Eşitlik (5.6) kullanılarak elde edilen başarı yüzdesi hesaplanmıştır.

$$\text{Başarı \%} = \frac{\text{İzlemenin Gerçekleştirildiği Çerçeve Sayısı}}{\text{Toplam Çerçeve Sayısı}} \times 100 \quad (5.6)$$

Eşitlik (5.6)'da *izlemenin gerçekleştirildiği çerçeve sayısı* hesaplanırken, sistemin izleme süresince yakaladığı tüm çerçeveler tek tek incelenmiş ve CAMSHIFT algoritması sonucunun, izlenecek olan nesnenin en az 1 piksellik kısmını kapsadığı çerçeveler dikkate alınmıştır. CAMSHIFT algoritmasının renk benzerliği sebebi ile farklı cisimleri izlediği durumlar, *sistemin izlenecek olan nesneyi bulamadığı* durumlar olarak değerlendirilmiştir.

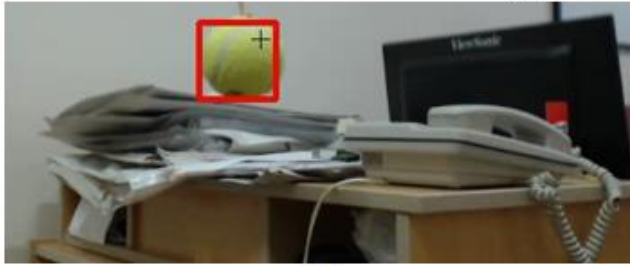
- 1. Durum. Tek renkli veya hafif renk geçişleri olan arka plan önünde rastgele hareketler ile sallanan tenis topunun izlenmesi.

Tek renkli duvar, pano ve masalardan oluşan arka plan önünde yapılan performans analizinde 35 saniyelik izleme gerçekleştirilmiştir. Sistem 35 saniye süresince 744 çerçeve yakalamış ve bu çerçevelerin 744'ünde topu başarılı bir şekilde izleyerek %100'lük bir başarı oranı sağlamıştır. 35 saniye süren izleme sırasında yakalanan 744 çerçeve sistemin saniyede yaklaşık olarak 21.3

çerçeve yakalayarak (21.3 fps) çalıştığını göstermektedir. Bu testten sağlanan izleme ekranı görüntüleri Şekil 5.26'da verilmiştir.



(a)



(b)



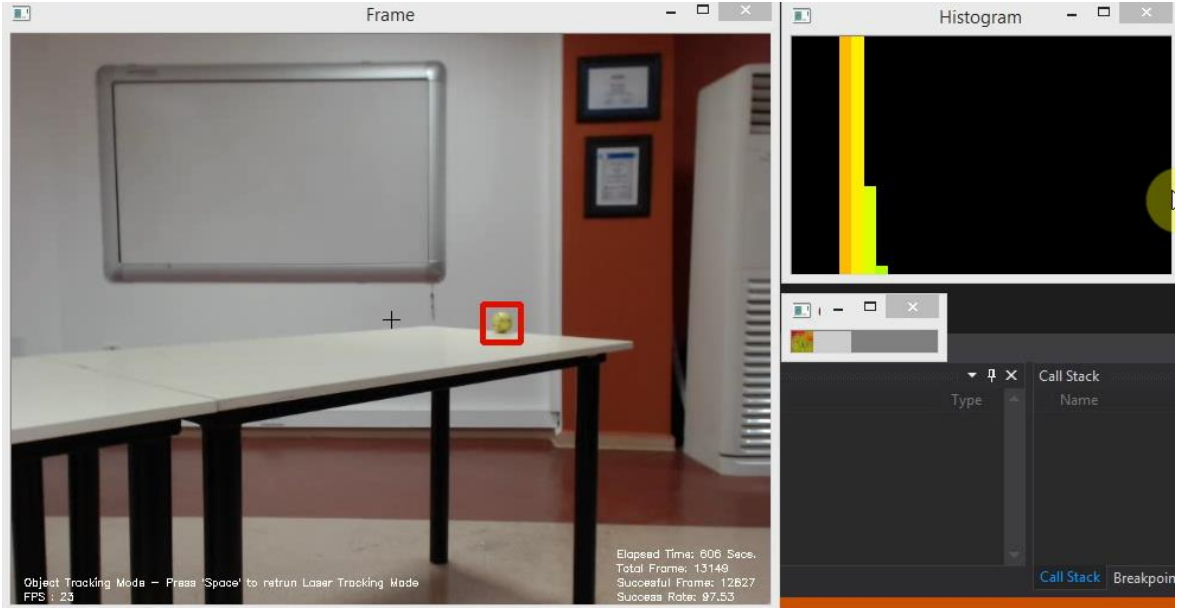
(c)

Şekil 5.26 Tek renkli veya hafif renk geçişleri olan arka plan önünde rastgele hareketler ile sallanan tenis topunun izlenmesi

- **2. Durum.** Tek renkli arka plan önündeki bir masa üzerinde, sarı renkli topun farklı hızlarda yuvarlanması.

Tek renkli duvar önündeki uzun bir masa üzerinde yapılan bu analizde top masa üzerinde farklı hızlarda yuvarlanmıştır. Yapılan testte sistem 1.9 saniye süresince 41 çerçeve yakalamış ve bu çerçevelerin 41'inde topu başarılı bir şekilde izleyerek %100'lük bir başarı oranı sağlamıştır. 1.9 saniye süre izleme sırasında yakalanan 41 çerçeve sistemin saniyede yaklaşık olarak 21.6 çerçeve yakalayarak çalıştığını göstermektedir. Top, 1.9 saniyelik izleme süresince kameraya yaklaşık olarak 2-3 metre uzaklıktaki, 3.7 metre uzunluğunda bir masa üzerinde yuvarlanmıştır. Bu durumda topun ortalama hızı 1.85 m/sn'dir.

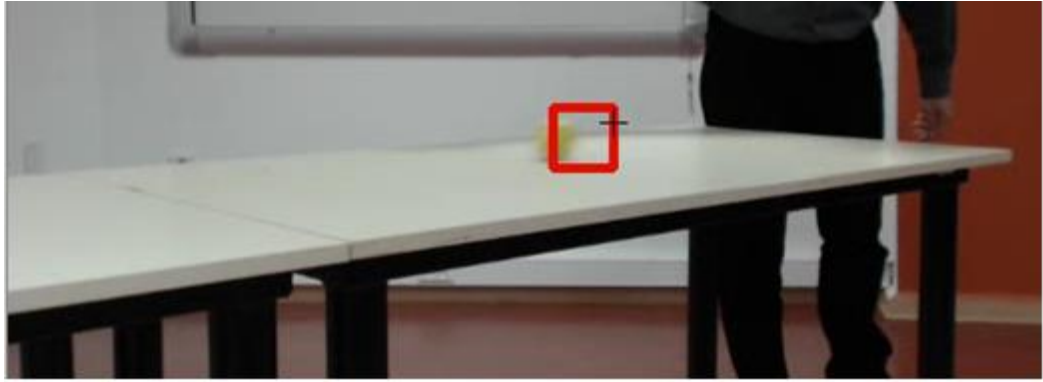
Aynı koşullar altında yapılan başka bir testte, top 3.7 metre uzunluğundaki masayı 1.57 saniye içerisinde kat edecek şekilde ortalama hızı 2.36 m/sn ile yuvarlanmış ve sistem %100 başarı elde etmiştir. Bu testten sağlanan izleme ekranı görüntüsü, izlenecek nesnenin içerisinde kalan en büyük kare alan ve bu alandan oluşturulan histogram ile birlikte Şekil 5.27’de verilmiştir. Topun yuvarlanması sırasında elde edilen izleme ekranı görüntüleri Şekil 5.28’de verilmiştir.



Şekil 5.27 İzleme ekranı, izlenecek olan nesnenin içerisinde kalan en büyük kare alan ve bu alan oluşturulan histogramı



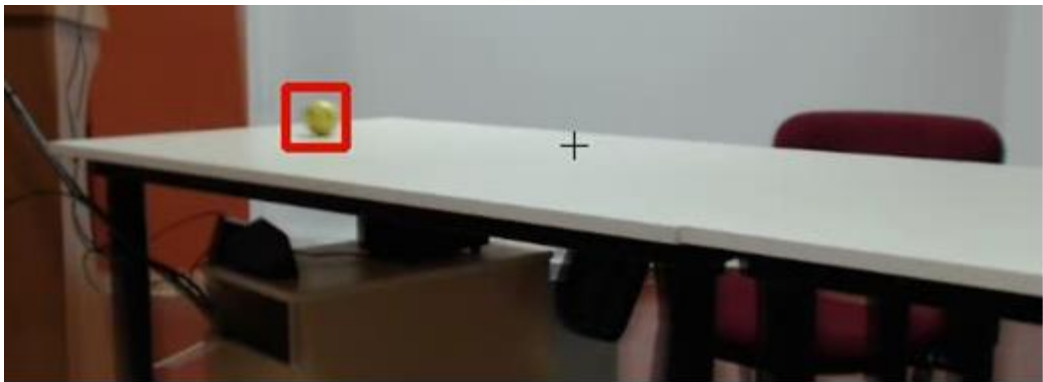
(a)



(b)



(c)



(d)

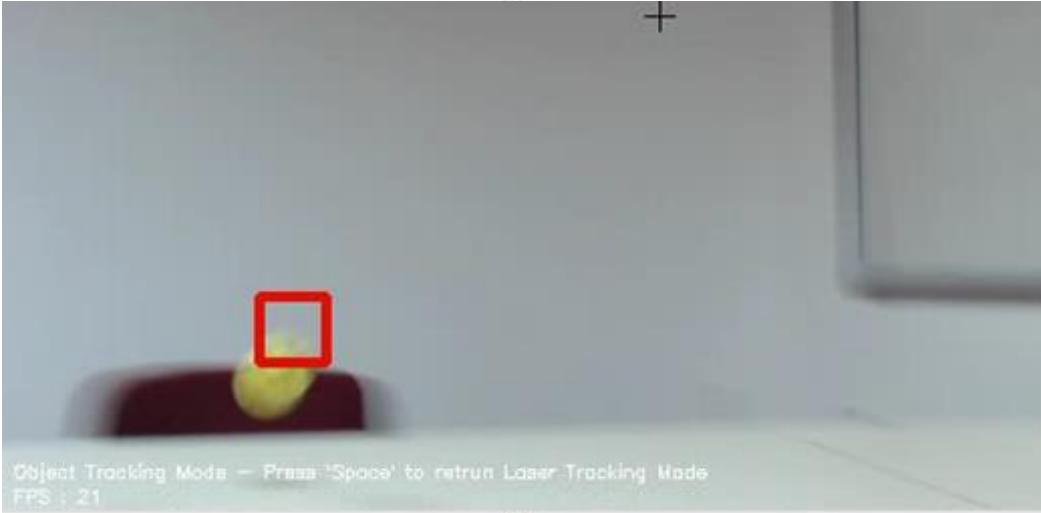
Şekil 5.28 Ortalama hızı 1.85 m/sn olan topun izlenmesi

- 3. Durum. Tek renkli arka plan önündeki bir masa üzerinde, sarı renkli topun sektirilmesi.

Tek renkli duvar önündeki uzun bir masa üzerinde yapılan bu analizde top masa üzerinde sektirilmiştir. Yapılan testte sistem, 2.13 saniye içerisinde 44 çerçeve yakalamış ve bu çerçevelerin 44'ünde topu başarılı bir şekilde izleyerek %100'lük bir başarı oranı sağlamıştır. 2.13 saniye süren izleme sırasında yakalanan 44 çerçeve sistemin saniyede yaklaşık olarak 20.7 çerçeve yakalayarak çalıştığını göstermektedir. Bu testten sağlanan izleme ekranı görüntüleri Şekil 5.29'da verilmiştir.

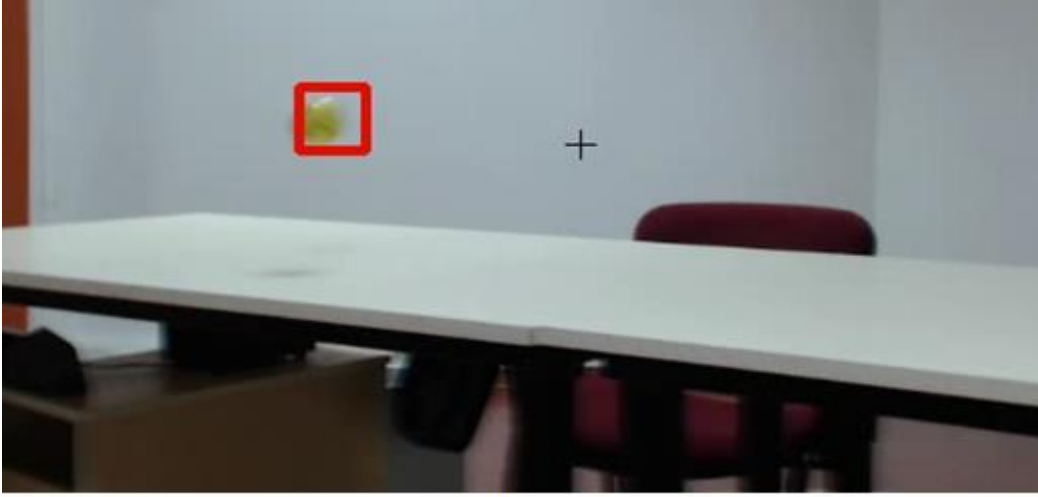


(a)

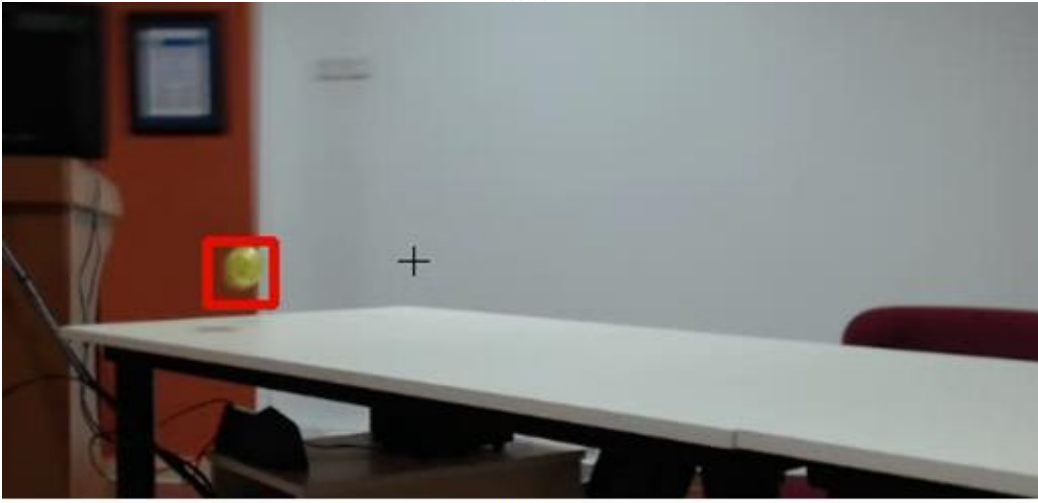


(b)

Şekil 5.29 Masa üzerinde sektirilen topun izlenmesi



(c)



(d)

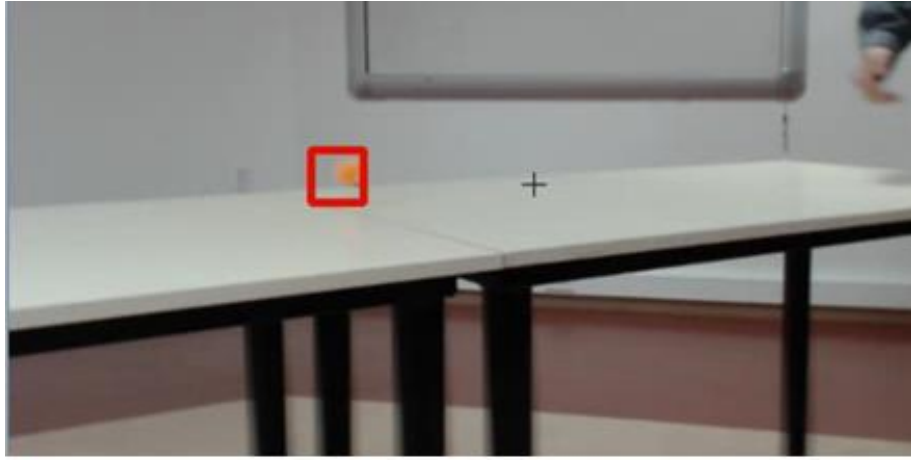
Şekil 5.29 devam

- *4. Durum. Tek renkli arka plan önündeki bir masa üzerinde, pinpon topunun yuvarlanması.*

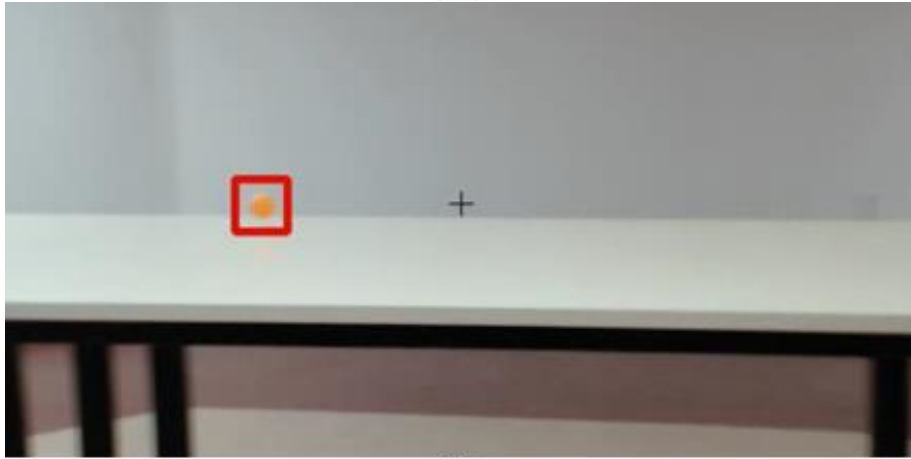
Sabit renkli duvar önündeki uzun bir masa üzerinde yapılan bu analizde renk özellikleri farklı olan bir top masa üzerinde yuvarlanmıştır. Yapılan testte sistem, 3.08 saniye içerisinde 65 çerçeve yakalamış ve bu çerçevelerin 65'ünde topu başarılı bir şekilde izleyerek %100'lük bir başarı oranı sağlamıştır. 3.08 saniye süren izleme sırasında yakalanan 65 çerçeve sistemin saniyede yaklaşık olarak 21.1 çerçeve yakalayarak çalıştığını göstermektedir. Top, 3.08 saniyelik izleme süresince kameraya yaklaşık olarak 2-3 metre uzaklıktaki, 3.7 metre uzunluğunda bir masa üzerinde yuvarlanmıştır. Bu durumda topun ortalama hızı 1.20 m/sn'dir. Bu testten sağlanan izleme ekranı görüntüleri Şekil 5.30'da verilmiştir.



(a)



(b)



(c)

Şekil 5.30 Ortalama hızı 1.20 m/sn olan topun izlenmesi

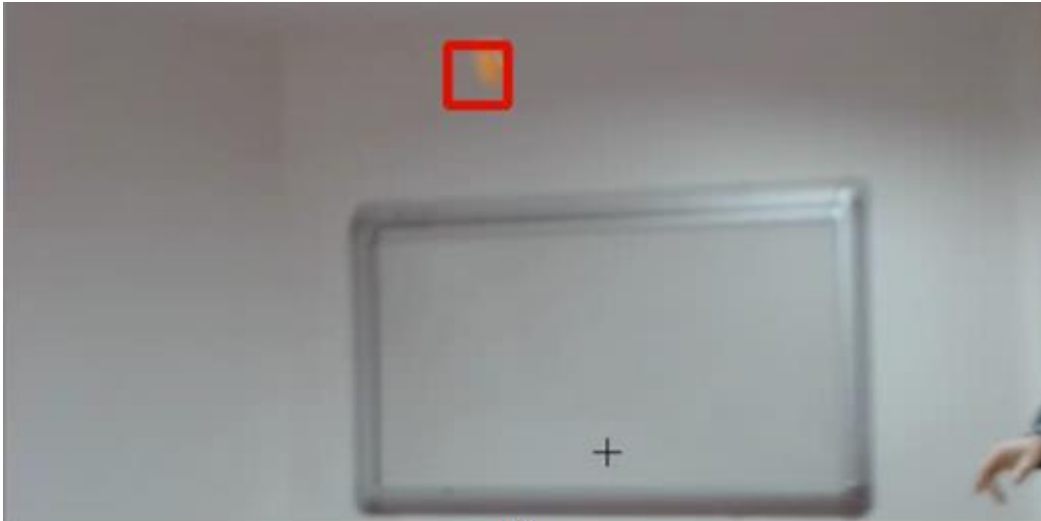


(d)

Şekil 5.30 devam

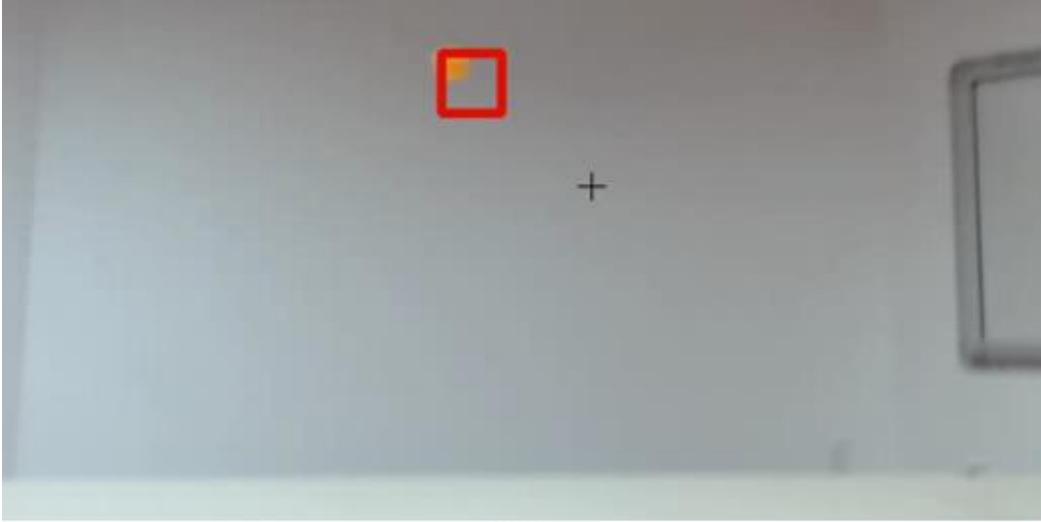
- *5. Durum. Tek renkli arka plan önündeki bir masa üzerinde, pinpon topunun sektirilmesi.*

Tek renkli duvar önündeki uzun bir masa üzerinde yapılan bu analizde renk özellikleri farklı olan bir top masa üzerinde sektirilmiştir. Yapılan testte sistem, 4.13 saniye içerisinde 88 çerçeve yakalamış ve bu çerçevelerin 74'ünde topu başarılı bir şekilde izleyerek %84.1'lik bir başarı oranı sağlamıştır. 4.13 saniye süren izleme sırasında yakalanan 88 çerçeve sistemin saniyede yaklaşık olarak 21.3 çerçeve yakalayarak çalıştığını göstermektedir. Bu testten sağlanan izleme ekranı görüntüleri Şekil 5.31'de verilmiştir.

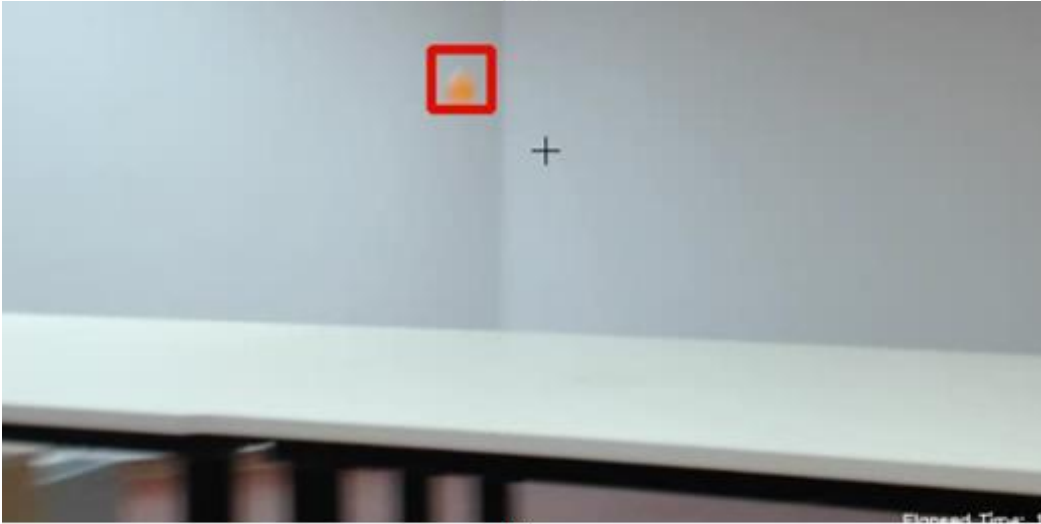


(a)

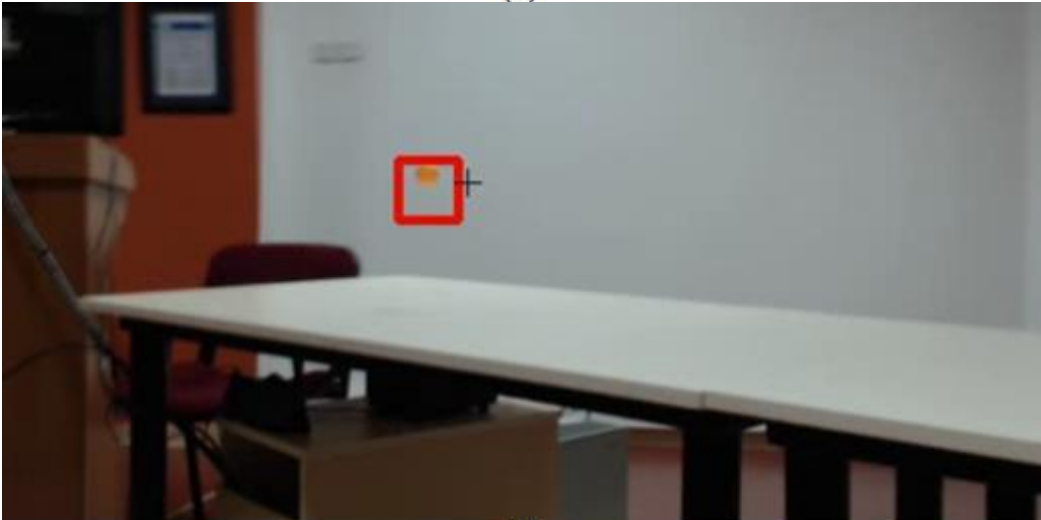
Şekil 5.31 Masa üzerinde sektirilen topun izlenmesi



(b)



(c)



(d)

Şekil 5.31 devam

- 6. Durum. Karışık renkli arka plan önünde, tenis topunun rastgele hareketler ile sallanması.

Renk geçişlerinin ve çeşitliliğinin fazla olduğu bir kütüphane önünde yapılan bu analizde tenis topu rastgele hareketler ile sallanmıştır. Analiz iki test ile gerçekleştirilmiştir. Birinci testte sistem, 37.23 saniye içerisinde 580 çerçeve yakalamış ve bu çerçevelerin 490'ında topu başarılı şekilde izleyerek %84.5'lik bir başarı oranı sağlamıştır. 37.23 saniye süren izleme sırasında yakalanan 580 çerçeve sistemin saniyede yaklaşık olarak 15.6 çerçeve yakalayarak çalıştığını göstermektedir. Aynı koşullar altında gerçekleştirilen ikinci testte, cisim yeniden işaretlenmiş ve bölütleme işlemi tekrarlanmıştır. Sistem, tekrarlanan bölütleme işlemindeki başarının değişmesi sonucunda ortaya çıkan histogram farklılığından etkilenmiş olarak, 92.30 saniye içerisinde 1460 çerçeve yakalamış ve bu çerçevelerin 1456'sında topu başarılı bir şekilde izleyerek %99.7'lik bir başarı oranı sağlamıştır. 92.30 saniye süren izleme sırasında yakalanan 1460 çerçeve sistemin saniyede yaklaşık olarak 15.8 çerçeve yakalayarak çalıştığını göstermektedir. İki testin ortalaması alındığında, sistemin bu şartlar altında saniyede 15.8 çerçeve yakalayarak %95.4 başarı oranı sağladığı söylenebilir. Bu testlerden sağlanan izleme ekranı görüntüleri Şekil 5.32'de verilmiştir.



(a)

Şekil 5.32 Karışık renkli arka plan önünde tenis topunun izlenmesi



(b)



(c)



(d)

Şekil 5.32 devam

- 7. Durum. Karışık renkli arka plan önünde, farklı renklere sahip bir oyuncuğun rastgele hareket ettirilmesi.

Renk geçişlerinin ve çeşitliliğinin fazla olduğu bir kütüphane önünde yapılan bu analizde esas rengi sarı olup, üzerinde farklı renkler de bulunduran bir oyuncak sarı rengi dikkate alınarak izlenmiştir. Yapılan testte sistem, 17.87 saniye içerisinde 327 çerçeve yakalamış ve bu çerçevelerin 290'ında oyuncuğu başarılı bir şekilde izleyerek %88.7'lik bir başarı oranı sağlamıştır. 17.87 saniye süren izleme sırasında yakalanan 327 çerçeve sistemin saniyede yaklaşık olarak 18.3 çerçeve yakalayarak çalıştığını göstermektedir. Bu testten sağlanan izleme ekranı görüntüleri Şekil 5.33'de verilmiştir.



(a)



(b)

Şekil 5.33 Karışık renkli arka plan önünde oyuncuğun izlenmesi



(c)

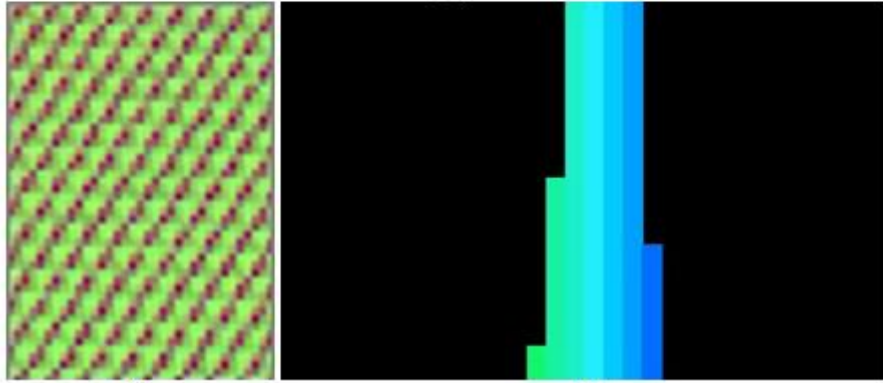
Şekil 5.33 devam

- 8. Durum. Karışık renkli arka plan önünde, farklı renklere sahip bir kitabın rastgele hareket ettirilmesi.

Renk geçişlerinin ve çeşitliliğinin fazla olduğu bir kütüphane önünde yapılan bu analizde karmaşık renklere sahip olan bir kitap, sistemin çoklu renk modu kullanılarak izlenmiştir. Yapılan teste sistem, 50.80 saniye içerisinde 760 çerçeve yakalamış ve bu çerçevelerin 744'ünde kitabı başarılı bir şekilde izleyerek %97.9'luk bir başarı oranı sağlamıştır. 50.80 saniye süren izleme sırasında yakalanan 760 çerçeve sistemin saniyede yaklaşık olarak 15,0 çerçeve yakalayarak çalıştığını göstermektedir. Bu teste izlenecek nesnenin çoklu renk modu ile seçilmesi, seçilen alanın H, S, V görüntüsü ve histogramı Şekil 5.34'de verilmiştir. Kitabın hareketi sırasında elde edilen izleme ekranı görüntüleri Şekil 5.35'de verilmiştir.



(a)



(b)

(c)

Şekil 5.34 Çoklu renk modu izlenecek alanın seçimi (a), seçilen alanın H,S,V görüntüsü (b) ve histogramı (c)



(a)



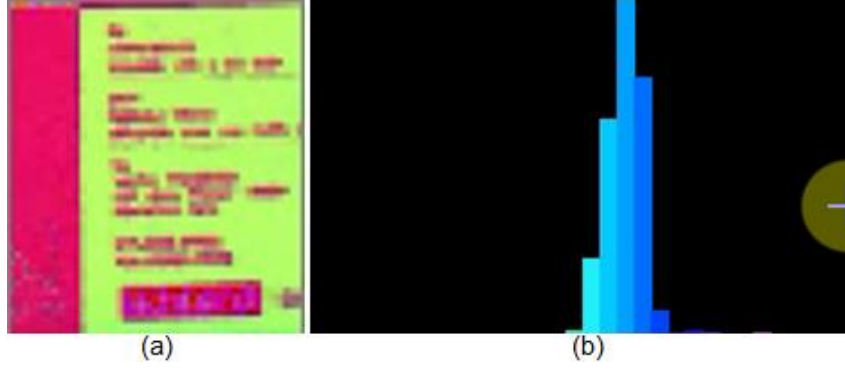
(b)

Şekil 5.35 Karışık renkli arka plan önünde bir kitabın çoklu renk modu kullanılarak izlenmesi

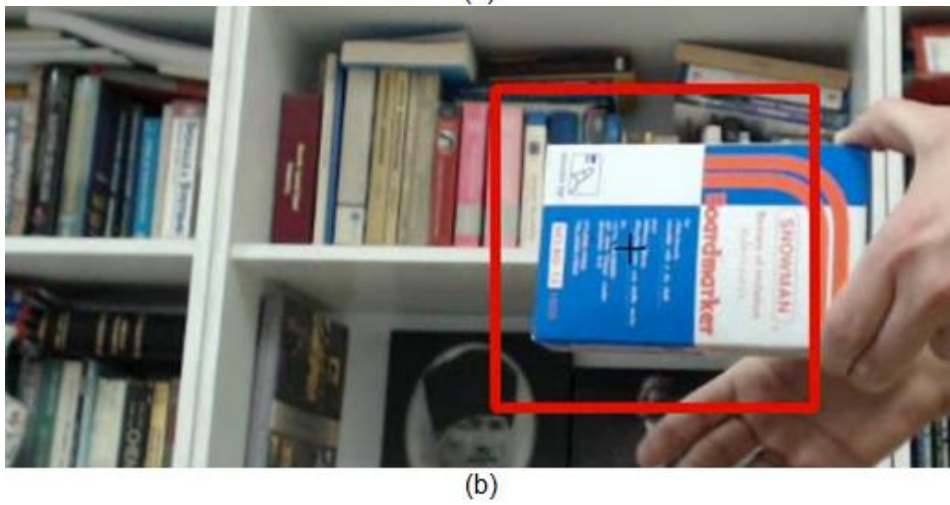
- 9. Durum. Karışık renkli arka plan önünde, farklı renklere sahip bir kutunun rastgele hareket ettirilmesi ve döndürülmesi.

Renk geçişlerinin ve çeşitliliğinin fazla olduğu bir kütüphane önünde yapılan bu analizde karmaşık renklere sahip olan bir kutu, sistemin çoklu renk modu kullanılarak izlenmiştir. Yapılan testte sistem, 65.03 saniye içerisinde 968 çerçeve yakalamış ve bu çerçevelerin 916'sında kutuyu başarılı bir şekilde izleyerek %94.6'lık bir başarı oranı sağlamıştır. 65.03 saniye süren izleme

sirasında yakalanan 968 çerçeve sistemin saniyede yaklaşık olarak 14.9 çerçeve yakalayarak çalıştığını göstermektedir. Bu testte izlenmek üzere seçilen alanın H, S, V görüntüsü ve histogramı Şekil 5.36'da verilmiştir. Kutunun hareketi sırasında elde edilen izleme ekranı görüntüleri Şekil 5.37'de verilmiştir.



Şekil 5.36 Çoklu renk modu ile seçilen alanın H,S,V görüntüsü (a) ve histogramı (b)



Şekil 5.37 Karışık renkli arka plan önünde bir kutunun çoklu renk modu kullanılarak izlenmesi



(c)



(d)

Şekil 5.37 devam

5.6. Bulgular ve Tartışma

Tez çalışması kapsamında gerçekleştirilen sistemin amacı, karmaşık renklerin olduğu, nesne ve arka plan renk dağılımlarının yakın olabileceği ortamlarda, gerçek zamanlı ve yüksek performanslı bir hareketli cisim izleme sistemi geliştirmektir.

Yapılan çalışmanın başarısını ölçmek amacıyla, 9 farklı durum altında testler gerçekleştirilmiştir. İzleme sonucunda elde edilen kare bölge ile izlenecek nesnenin en az 1 pikselde kesişmesi durumunda izlemenin başarılı kabul edildiği test

analizlerinden elde edilen ortalama başarı %95.6'dır. Sistem bu testlerde saniyede ortalama 18.9 çerçeve işleyerek gerçek zamanlı olarak çalışmıştır (Çizelge 6.1).

Geliştirilen sistemin çalışması sırasında algoritma adımlarının çeşitli yerlerinde kullanılan ve performansı doğrudan etkileyebilecek, önceden tanımlanmış parametreler kullanılmaktadır. Sistemin genel olarak çalıştırılacağı ortam koşullarına uygun olan denemeler sonucunda bulunan bu parametreler, farklı ortamlarda kullanım veya bazı özelliklerin dengelenmesi amacı ile programlama sırasında değiştirilebilir. Söz konusu parametreler algoritma içerisindeki kullanım sırasına göre incelenirse;

- “Lazer İzleme” bölümünde, kırmızı rengin bulunması adımında kullanılan $R > 180$ ve $G + B < 280$ koşulu sistemin kırmızı olarak kabul edeceği renk tonlarının belirlenmesini sağlar. $R > 180$ adımıdaki 180 değerinin büyütülmesi, sistemin daha kırmızı noktaları lazer ışını olarak algılamasını sağlarken, lazer ışınının mat cisimler üzerine düşmesi veya uzak mesafeden gönderilmesi sebebiyle renginin turuncu görünmesi durumunda, sistem tarafından fark edilmesini zorlaştırır. 180 değerinin küçültülmesi ise, sistemin daha zayıf kırmızı renkleri lazer ışını olarak algılamasını sağlarken, lazer olmayan parlamaların da lazer gibi görülme olasılığını artırır. $G + B < 280$ adımı ise, beyaza yakın olan renklerin kırmızı lazer ışını olarak algılanmasını önlemek için kullanılmıştır. 280 değeri büyütülürse, sistem daha geniş bir renk yelpazesini ve beyaz renkli parlamaları lazer ışını olarak değerlendirebilir. Değerin düşürülmesi ise, yalnızca saf kırmızının lazer olarak değerlendirilmesine sebep olur.
- “Lazer İzleme” bölümünde kullanılan, görüntünün ortalama parlaklık değerinin 15'in altına düşürülmesi ve görüntüdeki en parlak nokta değerinin %88'i altında kalan piksellerin sıfırlanması adımları birbirleri ile bağlantılıdır. Ortalama parlaklık değerinin 15 olarak seçilmesinin nedeni, bu değer normal aydınlatılan bir odada elde edilebilecek kadar yüksek ve lazer ışınını ortaya çıkarabilecek kadar düşük bir parlaklık değeri olmasıdır. Bu değer sistemin kullanıldığı ortamın aydınlanmasına göre değiştirilebilir. En parlak noktanın %88'i altında kalan noktaların sıfırlanması ise, sistemin lazer ışınına göre daha zayıf olan parlamaları yok sayması içindir. Bu değerın yükseltilmesi, belirlenen

noktaların aranan lazer ışını olma şansını artırır fakat ışının üzerine düştüğü cisim ışını soğursa lazer ışını belirlenemez. %88'değerinin düşürülmesi ise, ışının soğurulduğu durumlarda da belirlenmesini sağlarken, lazer ışını olmayan parlamalarında lazer olarak değerlendirilmesine sebep olur.

- “Lazer İzleme” bölümünde kullanılan, lazer ışınının 120 pikselden büyük olmaması durumu, farklı parlamaların lazer olarak belirlenmesini engellemek içindir. 120 piksellik bir alan yaklaşık olarak 1 metre mesafeden tutulan lazer ışınının yayılımı için yeterli bir mesafe olmaktadır. Daha yakından tutulan lazer ışınlarında 120 pikselden daha büyük yayılım olmakta ve sistem lazeri algılamamaktadır. Bu değerın büyütülmesi durumunda sistem yakın mesafeden tutulan lazer ışınlarını algılayabilir ancak, cisimlerin yüzeylerindeki ışık yansımalarının da lazer ışını olarak belirleme tehlikesi ortaya çıkar. Değerin düşürülmesi ise sistemin yansımalara karşı olan dayanıklılığını artırırken, lazer ışınının daha uzak bir mesafeden tutulması gerekliliğini doğurur.
- “Lazer İzleme” bölümünde sistemin lazer ışını olabilecek noktanın devamlılığını 20 çerçeve boyunca beklemesi, sistemin çalışma hızı ile ilişkilendirilmiştir. Saniyede ortalama 20 çerçeve yakalayan sistem bu şekilde lazer ışını 1 saniye süresince görmeyi bekler. Bunun altındaki değerler anlık parlamalar olarak nitelendirilir. Beklenecek çerçeve sayısının artırılması sistemin anlık parlamalara olan dayanıklılığını arttıracakken, hedef seçmek amacıyla yapılan işaretlemenin de süresinin uzamasına sebep olur. Beklenecek çerçeve sayısının düşürülmesi ise, sistemin sahnedeki hareketler nedeniyle oluşabilecek anlık parlamaları lazer olarak değerlendirmesine sebep olur.
- “Nesne Belirleme” bölümünde kullanılan, nesne içerisine oturan en büyük kare alanın bulunması algoritmasında 10 x 10 büyüklüğünde kare ile başlanması sistemin yeterince küçük nesnelere belirleyebilmesini sağlar. Daha büyük kare kullanımı nesne belirleme adımını hızlandırır ancak 10 pikselden küçük nesnelere bölütleme problemleri doğurur. Cismin çok büyük olduğu durumlarda, 10 x 10'luk karenin büyütülerek cismin içine oturması daha uzun süren bir işlem haline alır. Bu gibi kullanımlarda, karenin başlangıç boyutu daha büyük seçilebilir.

- “Nesne İzleme” bölümünde maske oluşumunda kullanılan $(H, S, V)_{min}$ ve $(H, S, V)_{max}$ değerleri elde edilirken standart sapma 3 tamsayısı ile çarpılmıştır. Bu değer in büyütülmesi durumunda, izleme sisteminin renk özellikleri değişimine olan duyarlılığı ve izlenecek nesne dışındaki cisimleri takip edebilme olasılığı artar. Değer in düşürülmesi durumunda, sistemin renk özellikleri değişimine olan duyarlılığı ve izlenecek nesne dışındaki cisimleri takip edebilme olasılığı azalır.

Sistemin görüntü işleme algoritmasında yapılabilecek bu değişikliklerin yanında, hedef izleme ünitesinin kamerayı hareket ettirme hızı da kullanım şartlarına göre değiştirilebilir. Geliştirilen sistemin kamerayı hareket ettirme hızı, kamera görüntüsünün aşırı hızdan dolayı bozulmaması ve kameranın yeni konum bilgisi gelene kadar hedefi geçerek kararlı duruma gelememesi sebebiyle sınırlandırılmıştır. Bu sınırlandırma ile kamera x ve y düzlemlerinde 1 saniye içerisinde maksimum 92.8° dönebilmektedir.

6. SONUÇLAR

Bu tez çalışmasında, lazer ışını ile işaretlenmiş olan hareketli nesnelere izleyen bir sistem tasarımı yapılmıştır. Sistem, izlenecek olan nesnenin seçilmesi için gerekli olan lazer ışınının belirlenmesi ile çalışmaya başlar. Lazer ışınının görüntü üzerinde düştüğü noktanın konumu, ışının belirlendiği andan itibaren ışın yok olana kadar her çerçevede hesaplanır. Lazer ışını kaybolduktan sonraki ilk çerçevede, ışının bilinen son konumunun olduğu noktadaki cisim bölütlenerek görüntünün geri kalanından ayrılır. Ayrılmış olan bu cisim sistem tarafından izlenecek nesne olarak belirlenir ve nesnenin renk histogramı oluşturulur. Takip eden çerçevelerde nesne renk histogramı kullanılarak geriye izdüşüm görüntüleri elde edilir. Geriye izdüşüm görüntüleri üzerinde CAMSHIFT uygulanarak nesne izleme gerçekleştirilir.

CAMSHIFT uygulamalarında, izlenen nesne ile aynı renk özelliklerine sahip başka nesnelerin veya arka planların sahnede bulunması durumunda nesne izleme hataları gözlenmektedir [79], [80]. Bu hatalar çoğunlukla, renk tabanlı izleme yapan algoritmalarda görülür. CAMSHIFT algoritması da, bazı durumlarda nesnenin başarılı bir şekilde izlenmesini sağlamaya yetmeyecek bir bilgi olan, renk bilgisini kullanan bir izleme algoritmasıdır. Geliştirilen sistemde bu probleme çözüm olarak, histogram geriye izdüşüm görüntüsü üzerinde birleşik elemanların en büyüğü bulunmaktadır. Bu işlem izlenecek olan cismin, sahnede benzer renk özelliklerine sahip en büyük cisim olacağı varsayımına dayanır. Bu yöntem kullanılarak geliştirilen nesne izleme algoritmasında, izlenecek olan nesneden daha büyük ve benzer renk özelliklerine sahip bir cisim sahneye girmediği, başarı oranının iyileştiği gözlenmiştir.

Yapılan araştırmalarda, CAMSHIFT kullanılan izleme sistemlerinde izlenecek nesne seçiminin, kameradan elde edilen görüntüler üzerinde fare imleci ile veya sisteme önceden öğretilmiş nesne görüntüleri ile yapıldığı görülmüştür. Geliştirilen sistemde ise, izlenecek nesnenin seçimi bilgisayar ile hiçbir etkileşim gerekmeksizin, lazer ışını ile işaretleme yapılarak gerçekleştirilmektedir. Bu yolla yapılan işaretlemede sistemin renk histogramını çıkaracağı bölgeyi otomatik olarak seçmesi gerekir. Sistem bu işlemi özel olarak geliştirilen ve nesne içerisine oturan en büyük kare alanı bulmayı sağlayan özgün bir algoritma ile gerçekleştirmektedir. Öte yandan literatürdeki bazı uygulamalarda [3], CAMSHIFT kullanılarak gerçekleştirilen izleme

sırasında kullanılan maske değerlerinin, kullanıcı tarafından izlenecek nesnenin ve arka planın renk değerlerine göre girildiği görülmüştür. Geliştirilen sistemde bu ayarlamalar tamamen otomatik olarak yapılmaktadır. Sistem, izlenecek nesnenin lazer işaretleme ile seçilmesinden sonra başka bir veri girişine ihtiyaç duymaksızın izleme işlemine başlar.

Geliştirilen sistem, verilen ön koşulların sağlandığı durumlarda izleme işlemini gerçek zamanlı olarak başarılı bir şekilde gerçekleştirebilmektedir. 9 farklı durum altında gerçekleştirilen testlerde elde edilen, izleme başarı yüzdeleri ve sistemin bir saniyede işleyebildiği çerçeve sayıları Çizelge 6.1'de verilmiştir. Çizelge 6.1 incelendiğinde, sistemin sabit veya renk geçişlerinin sınırlı olduğu arka planlar önünde izleme yaptığı 1, 2, 3 ve 4. Durumlarda başarı yüzdesi %100'dür ve sistem saniyede yaklaşık olarak 21-22 çerçeve yakalamaktadır. Aynı arka plan önünde izleme yapılan 5. Durumda hedef olarak kullanılan pinpon topunun renk özellikleri, bölütleme ve histogram çıkarımının başarısını etkilemiş ve sistemin bu durumdaki başarı yüzdesi benzer diğer durumlara göre daha düşük çıkmıştır. Karışık renkli arka plan önünde yapılan rastgele hareket testlerinde ön koşul özelliklerine uyan tenis topu kullanılan 6. Durumun başarı yüzdesi, ön koşulu zorlayan oyuncak kullanılan 7. Duruma göre daha yüksek çıkmıştır. 6 ve 7. Durumlarda saniyede yakalanan çerçeve sayılarının önceki durumlara göre düşük olmasının nedeni, hedef bulunamadıkça büyütülen arama çerçevesinin sistemde oluşturduğu ekstra işlem yüküdür. 8 ve 9. Durumlarda, 6 ve 7. Durumlarda kullanılan arka plan önünde, çoklu renk modu kullanılarak daha karışık renkli nesnelere izlenmiştir. İki farklı nesne için başarı yüzdeleri ve hız birbirlerine oldukça yakındır. Bu iki durumdaki hız düşüşünün nedeni, çoklu renk modunda daha büyük histogramların geriye izdüşüm görüntüsünün çıkarılmasıdır.

Gerçekleştirilen testler sonucunda, izlenecek olan hedefin hareket hızının görüntü işleme tekniklerinin başarısını etkilemediği görülmüştür. Renk tabanlı yapılan izlemelerde izlenecek cismin, kamera tarafından renginin algılanamayacak kadar hızlı hareket ettiği durumlarda performans etkilenmesi görülür.

Çizelge 6.1 Sistemin performans analizi sonuçları

Durumlar	Başarı Yüzdesi (%)	Saniyedeki Çerçeve Sayısı (FPS)
1. Durum	100	21.3
2. Durum	100	21.6
3. Durum	100	20.7
4. Durum	100	21.1
5. Durum	84.1	21.3
6. Durum	95.4	15.8
7. Durum	88.7	18.3
8. Durum	97.9	15.0
9. Durum	94.6	14.9
Ortalama	95.6	18.9

6.1. Gelecek Çalışma Planları

Gerçekleştirilen tez çalışması sonucunda elde edilen bulgulara ek olarak, aşağıda sayılan konularda yapılabilecek araştırmalar, gelecekte konuyla ilgili yeniliklerin önünü açacaktır.

- Mevcut sistemde merkezi işlem birimi (CPU) üzerinde çalıştırılmakta olan görüntü işleme algoritması, grafik işlem birimi (GPU) üzerinde paralel işlem yapacak şekilde çalıştırılarak büyük performans artışları sağlanabilir [6].
- SIFT (ölçekten bağımsız öznitelik dönüşümü) [50] ve SURF (Hızlandırılmış dayanıklı öznitelikler) [81] gibi öznitelikleri kullanan algoritmalar sisteme dahil edilerek, daha karmaşık şekil ve renk özelliklerine sahip nesnelerin başarılı bir şekilde izlenmesi gerçekleştirilebilir. Sistemin bu alanda geliştirilmesi ile ortam aydınlatmasına bağlı ön koşulların gereksinimi ortadan kaldırılabilir.
- Geliştirilen algoritmanın “Lazer İzleme” bölümünde yapılabilecek değişiklikler ile lazer ışınının daha zorlu şartlar altında belirlenmesi sağlanabilir. Bu işlem lazer

ışını belirleme konusundaki farklı arařtırmaların [14] sisteme eklenmesi ile gerekleřtirilebilir.

- Blm 5.6'da bahsedilen ve sistemin performansına dođrudan etkisi bulunan parametrelerin, izlenecek nesne ve izleme yapılacak ortama gre otomatik olarak ayarlanacađı bir optimizasyon adımıının algoritmaya eklenmesiyle, farklı kořullar altında gerekleřtirilen izleme iřlemlerinin performansı arttırılabilir.

KAYNAKLAR LİSTESİ

- [1] N. Yıldız, "Görüntü İşlemenin Dünü, Bugünü ve Geleceği," *Elektrik Mühendisliği*, vol. 48, no. 440, s. 11-16, 2010.
- [2] U. M. Cahn von Seelen and R. Bajcsy, "Adaptive Correlation Tracking of Targets with Changing Scale," University of Pennsylvania Technical Reports (CIS), Philadelphia, PA., 1996.
- [3] J. Liu, H. Sun, H. Yang, W. Li and L. Wu, "CamShift Based on Multi-feature Fusion and Kalman Prediction for Real-time Visual Tracking," *Information Technology Journal*, vol. 13, no. 1, s. 159-164, 2014.
- [4] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technology Journal*, vol. 2, no. 2, s. 13-27, 1998.
- [5] P. Hidayatullah and H. Konik, "CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking," in *Electrical Engineering and Informatics (ICEE) International Conference*, Bandung, 2011.
- [6] D. Exner, E. Bruns, D. Kurz, A. Grundhöfer and O. Bimber, "Fast and Robust CAMShift Tracking," in *Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference*, San Francisco, CA, 2010.
- [7] J. G. Ellis, K. Kramer and S. C. Stubberud, "Image Correlation Based Video Tracking," in *Systems Engineering, 21st International Conference*, Las Vegas, NV, 2011.
- [8] A. Jog and S. Halbe, "Multiple Objects Tracking Using CAMShift Algorithm and Implementation of Trip Wire," *Image, Graphics and Signal Processing*, vol. 5, no. 6, s. 43-48, 2013.
- [9] H. Rami, M. Hamri and L. Masmoudi, "Objects Tracking in Images Sequence using Local Binary Pattern (LBP)," *International Journal of Computer Applications*, vol. 63, no. 20, s. 19-23, 2013.
- [10] G. Nebehay and R. Pflugfelder, "TLM: Tracking-Learning-Matching of Keypoints," in *Distributed Smart Cameras, Seventh International Conference*, Palm Springs, CA, 2013.
- [11] G. Bradski and A. Kaehler, "Tracking and Motion," in *Learning OpenCV Computer Vision with the OpenCV Library*, O'Reilly Media, Inc., 2008, s. 316-370.
- [12] M. Meško and Š. Toth, "Laser Spot Detection," *Journal of Information, Control and Management Systems*, vol. 11, s. 35-41, 2013.

- [13] H.-A. Le, M.-T. Doan and M.-T. Tran, "Webcam-Based Laser Dot Detection Technique In Computer Remote Control," *Journal of Science and Technology*, vol. 48, no. 4, s. 73-81, 2010.
- [14] D. Krstinić, A. K. Skelin and I. Milatić, "Laser Spot Tracking Based on Modified Circular Hough Transform and Motion Pattern Analysis," *Sensors*, vol. 14, no. 11, s. 20112-20133, 2014.
- [15] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of Imaging Understanding Workshop*, s. 121-130, 1981.
- [16] B. Cornett, "10 Benefits of PTZ Security Cameras," EZWatch, 2/07/2014. <http://www.ezwatch.com/security-cameras/10-benefits-of-pan-tilt-zoom-security-cameras>. Erişim Tarihi: 25/11/2015.
- [17] W. Y. Kim, "Pan/tilt Camera". United States Patent US 6,503,000 B1, 07/01/2003.
- [18] A. Dankers, "Realtime Stereo Active Vision," The Australian National University, http://users.cecs.anu.edu.au/~rsl/rsl_active.html. Erişim Tarihi: 25/11/2015.
- [19] D. Murray and A. Basu, "Motion Tracking with an Active Camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, s. 449-459, 1994.
- [20] J. Y. Aloimonos, I. Weiss and A. Bandyopadhyay, "Active Vision," *International Journal of Computer Vision*, s. 333-356, 1988.
- [21] S.-N. Lim, A. Elgammal and L. S. Davis, "Image-Based Pan-Tilt Camera Control In A Multi-Camera Surveillance Environment," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference*, 2003.
- [22] K. Kurihara, S. Hoshino, K. Yamane and Y. Nakamura, "Optical Motion Capture System with Pan-Tilt Camera Tracking and Realtime Data Processing," in *IEEE International Conference on Robotics & Automation*, Washington, DC, 2002.
- [23] S. Stolle and R. Rysdyk, "Flight Path Following Guidance For Unmanned Air Vehicles With Pan-Tilt Camera For Target Observation," in *Digital Avionics Systems Conference*, Indianapolis, IN, USA, 2003.
- [24] M. Paralic, "Fast Connected Component Labeling in Binary Images," in *Telecommunications and Signal Processing (TSP) International Conference*, Prague, 2012.
- [25] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, s. 35-45, 1960.

- [26] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third Edition ed., Pearson Education International, 2008.
- [27] "Color Palettes, Bit Depth, and Dithering," The University off Oklahoma, http://www.ou.edu/class/digitalmedia/articles/ColorPalettes_Dithering_BitDepth.html. Eriřim Tarihi: 19/01/2016.
- [28] G. H. Joblove and D. Greenberg, "Color Spaces for Computer Graphics," in *Computer Graphics (SIGGRAPH Proceedings)*, Atlanta, 1978.
- [29] İ. Yılmaz, M. Güllü, T. Baybura and O. A. Erdoğan, "Renk Uzayları ve Renk Dönüşüm Programı (RDP)," *Afyon Kocatepe Üniversitesi Fen Bilimleri Dergisi*, vol. 2, no. 2, s. 19-35.
- [30] K. Jack, "HSI, HLS, and HSV Color Spaces," in *Video Demystified*, Elsevier, 2007, s. 27-28.
- [31] N. A. Ibraheem, M. M. Hasan, R. Z. Khan and P. K. Mishra, "Understanding Color Models: A Review," *ARPN Journal of Science and Technology*, vol. 2, no. 3, s. 265-275, 2012.
- [32] A. R. Smith, "Color Gamut Transform Pairs," in *SIGGRAPH*, 1978.
- [33] W. Chen, Y. Q. Shi and G. Xuan, "Identifying Computer Graphics Using HSV Color Model and Statistical Moments of Characteristic Functions," in *Multimedia and Expo, IEEE International Conference*, Beijing, 2007.
- [34] B. Bayram, "Yıldız Teknik Üniversitesi İnşaat Fakültesi Jeodezi ve Fotogrametri Mühendisliği Bölümü," Yıldız Teknik Üniversitesi, <http://www.yildiz.edu.tr/~bayram/sgi/saygi.htm>. Eriřim Tarihi: 09/10/2015.
- [35] U. Sinha, "AI Shack Tutorials: Mathematical Morphology," AI Shack, <http://aishack.in/tutorials/mathematical-morphology-composite-operations/>. Eriřim Tarihi: 20/01/2016.
- [36] "MathWorks: Morphological Reconstruction," MATLAB, <http://www.mathworks.com/help/images/morphological-reconstruction.html>. Eriřim Tarihi: 01/12/2015.
- [37] A. Şengür, İ. Türkođlu and C. M. İnce, "Eđitici-siz Yapay Sinir Ağları ile Görüntü Bölütleme Uygulamaları," in *IEEE 13th Signal Processing and Communications Applications (SIU)*, Kayseri, 2005.
- [38] K. Hanbay, F. M. Talu and A. Karcı, "Dalgacık Dönüşümü ve Yapay Arı Koloni Algoritması ile Renkli Doku Görüntülerinin Bölütlenmesi," in *Signal Processing and Communications Applications Conference (SIU)*, 20th, Mugla, 2012.

- [39] L. G. Roberts, "Machine Perception of Three-Dimensional Solids," Massachusetts Institute of Technology, 1963.
- [40] J. M. S. Prewitt, "Object Enhancement and Extraction," in *Picture Processing and Psychopictorics*, New York, Academic Press, 1970, s. 75-149.
- [41] I. Sobel and G. Feldman, "History and Definition of the so-called "Sobel Operator", " 20/08/2015.
https://www.researchgate.net/publication/281104656_An_Isotropic_3x3_Image_Gradient_Operator. Erişim Tarihi: 21/01/2016.
- [42] M. Topaloğlu and A. Gangal, "Watershed Dönüşümü Kullanılarak Corpus Callosumun Bölütlenmesi," in *Uluslararası Radyo Bilimleri Birliği 3. Bilimsel Kongresi*, Ankara, 2006.
- [43] A. S. Keçeli and B. A. Can, "Beyaz Cevher Lezyonlarının Otomatik Bölütlenmesi," in *Signal Processing and Communications Applications Conference*, Kocaeli, 2009.
- [44] S. Eddins, "MathWorks: The Watershed Transform: Strategies for Image Segmentation," MathWorks, 2002.
<http://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html>. Erişim Tarihi: 21/01/2016.
- [45] A. Yılmaz, O. Javed and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.
- [46] "Colorimetry, 3rd Edition," International commission on Illumination, 2004.
- [47] H. P. Moravec, "Towards Automatic Visual Obstacle Avoidance," in *5th International Joint Conference on Artificial Intelligence*, Massachusetts, 1977.
- [48] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Fourth Alvey Vision Conference*, 1988.
- [49] J. Shi and C. Tomasi, "Good Features to Track," in *Computer Vision and Pattern Recognition, IEEE Computer Society*, Seattle, 1993.
- [50] D. G. Lowe, "Object Recognition From Local Scale-Invariant Features," in *International Conference on Computer Vision*, Kerkyra, 1999.
- [51] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Computer Vision and Pattern Recognition*, Fort Collins, 1999.

- [52] N. Oliver, B. Rosario and A. Pentland, "A Bayesian Computer Vision System for Modelling Human Interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, s. 831-843, 2000.
- [53] A. Monnet, A. Mittal, N. Paragios and V. Ramesh, "Background Modelling and Subtraction of Dynamic Scenes," in *Computer Vision*, Nice, 2003.
- [54] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, s. 603-619, 2002.
- [55] C. Berge, "The Theory of Graphs and Their Applications" (Fransızca'dan çeviri), Wiley, 1962.
- [56] C. Çığla and A. A. Alatan, "Bölge Tabanlı Çizge Kesik Görüntü Bölütlemesi," in *Signal Processing, Communication and Applications Conference, 2008*, Aydın, 2008.
- [57] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, vol. 1, no. 4, s. 321-331, 1988.
- [58] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, s. 273-297, 1995.
- [59] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning*, 1992.
- [60] İ. Çayıroğlu, "Kalman Filtresi ve Programlama," *Fen ve Teknoloji Bilgi Paylaşımı*, 2012.
- [61] P. P. Dash, D. Patra, S. K. Mishra and J. Sethi, "Kernel based Object Tracking using Color Histogram Technique," *International Journal of Electronics and Electrical Engineering*, vol. 2, no. 4, s. 28-35, 2012.
- [62] Z.-Q. Wen and Z.-X. Cai, "Mean Shift Algorithm and Its Application in Tracking of Objects," in *Fifth International Conference on Machine Learning and Cybernetics*, Dalian, 2006.
- [63] D. Conte, P. Foggia, G. Percannella and M. Vento, "A MultiView Appearance Model for people re-identification," in *8th International Conference on Advanced Video and Signal-Based Surveillance*, Klagenfurt, 2011.
- [64] S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 24, s. 509-522, 2002.

- [65] G. Panin, A. Ladikos and A. Knoll, "An Efficient and Robust Real-Time Contour Tracking System," in *Computer Vision Systems*, 2006.
- [66] G. Bradski and A. Kaehler, "What Is OpenCV?," in *Learning OpenCV*, O'Reilly Media, Inc., 2008, p. 1.
- [67] "OpenCV About," Open Source Computer Vision Library, <http://opencv.org/about.html>. Erişim Tarihi: 01/12/2015.
- [68] "OpenCV Introduction," Open Source Computer Vision Library, <http://docs.opencv.org/2.4/modules/core/doc/intro.html>. Erişim Tarihi: 01/12/2015.
- [69] "Arduino: Introduction," Arduino, <https://www.arduino.cc/en/Guide/Introduction>. Erişim Tarihi: 02/12/2015.
- [70] "Arduino: Environment," Arduino, <https://www.arduino.cc/en/Guide/Environment>. Erişim Tarihi: 02/12/2015.
- [71] "Logitech HD Pro Webcam C920," Logitech, <http://www.logitech.com/tr-tr/product/hd-pro-webcam-c920>. Erişim Tarihi: 04/12/2015.
- [72] "Vanadium Labs: Arbotix," Vanadium Labs, <http://www.vanadiumlabs.com/arbotix>. Erişim Tarihi: 04/12/2015.
- [73] "Trossen Robotics: Arbotix - M Robocontroller," Trossen Robotics, <http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx>. Erişim Tarihi: 04/12/2015.
- [74] "Robotis: Dynamixel," Robotis, http://en.robotis.com/index/product.php?cate_code=101010. Erişim Tarihi: 04/12/2015.
- [75] "Trossen Robotics: Dynamixel AX-18A," Trossen Robotics, <http://www.trossenrobotics.com/dynamixel-ax-18A-robot-actuator.aspx>. Erişim Tarihi: 04/12/2015.
- [76] M. J. Swain and D. H. Ballard, "Indexing Via Color Histograms," in *Computer Vision*, Osaka, 1990.
- [77] E. Yuan, "Eric Yuan's Blog : Continuously Adaptive Mean Shift," Eric Yuan's Blog, 18 9 2013. <http://eric-yuan.me/continuously-adaptive-shift/>. Erişim Tarihi: 19/01/2016.
- [78] "OpenCV Tutorials: Meanshift and Camshift," Open Source Computer Vision Library, http://docs.opencv.org/master/db/df8/tutorial_py_meanshift.html#gsc.tab=0. Erişim Tarihi: 18/01/2016.

- [79] C. Zhang, Y. Qiao, E. Fallon and C. Xu, "An Improved CamShift Algorithm for Target Tracking in Video Surveillance," in *9th IT&T Conference*, Dublin, 2009.
- [80] N. M. Artner, "A Comparison of Mean Shift Tracking Methods," in *Spring Conference on Computer Graphics*, Budmerice, 2008.
- [81] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, s. 346-359, 2008.