

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DÖRT ROTORLU İNSANSIZ HAVA ARACI İLE SÜREKLİ
UYARLAMALI ORTALAMA KAYMA ALGORİTMASI
KULLANILARAK HAREKETLİ NESNE TAKİBİ

Musab COŞKUN

Yüksek Lisans Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Elektronik Programı
Tez Danışmanı: Yrd. Doç. Dr. Sencer ÜNAL
ŞUBAT-2015

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DÖRT ROTORLU İNSANSIZ HAVA ARACI İLE SÜREKLİ UYARLAMALI
ORTALAMA KAYMA ALGORİTMASI KULLANILARAK HAREKETLİ NESNE
TAKİBİ

YÜKSEK LİSANS TEZİ
Musab COŞKUN
(112113109)

Anabilim Dalı: Elektrik-Elektronik Mühendisliği

Programı: Elektronik

Tezin Enstitüye Verildiği Tarih: 11 Şubat 2015

Tezin Savunulduğu Tarih: 26 Şubat 2015

Tez Danışmanı: Yrd. Doç. Dr. Sencer ÜNAL(F.Ü.)

Diğer Jüri Üyeleri: Yrd. Doç. Dr. Erkan DUMAN(F.Ü.)

Yrd. Doç. Dr. Mehmet ÜSTÜNDAĞ(B.Ü.)

ÖNSÖZ

Tez çalışmam boyunca desteklerini benden esirgemeyen değerli hocam, tez danışmanım Yrd. Doç. Dr. Sencer ÜNAL'a teşekkürlerimi sunarım. Ayrıca hayatım boyunca yanımda olan aileme teşekkürü her daim bir borç bilirim.

Musab COŞKUN

ELAZIĞ – 2015

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	X
KISALTMALAR.....	XI
SEMBOLLER LİSTESİ	XII
1. GİRİŞ.....	1
1.1. Tezin Amacı	4
1.2. Tezin Organizasyonu.....	5
2. NESNE TAKİBİNDE KULLANILAN PARAMETRELER.....	6
2.1. Nesne Gösterimleri.....	7
2.1.1. Noktalar ile Gösterim	8
2.1.2. Geometrik Şekil ile Gösterim.....	8
2.1.3. Nesne Silueti ve Çevre Çizgisi ile Gösterim	9
2.1.4. Eklemlili Şekil Modelleri ile Gösterim	9
2.1.5. İskelet Model ile Gösterim	9
2.2. Özellik Seçimi	11
2.2.1. Renk.....	11
2.2.2. Kenar	12
2.2.3. Optik Akış	13
2.2.4. Doku	13
2.3. Nesne Bulma	15
2.3.1. Nokta Bulucular	16
2.3.2. Arka Plan Çıkarma	17
2.3.3. Bölütleme	19
2.3.3.1. Ortalama Kayma Kümelemesi	19
2.3.3.2. Graf Kesme Kullanarak İmge Bölütleme.....	20
2.3.3.3. Aktif Konturlar	20
2.3.4. Denetimli Öğrenme	21

2.3.4.1.	Uyarlanabilir Destekleme.....	22
2.3.4.2.	Destek Vektör Makineleri	23
3.	NESNE TAKİP YÖNTEMLERİ	24
3.1.	Nokta Tabanlı Nesne Takibi.....	27
3.1.1.	Deterministik Yöntemler	27
3.1.2.	İstatiksel Yöntemler	28
3.1.2.1.	Kalman Filtresi ile Nesne Takibi.....	29
3.1.2.2.	Parçacık Filtresi ile Nesne Takibi	30
3.2.	Çekirdek Tabanlı Nesne Takibi.....	32
3.2.1.	Şablon ve Yoğunluk Tabanlı Görünüm Modelleri Kullanılarak Nesne Takibi	32
3.2.2.	Tek bir Nesnenin Takibi.....	32
3.2.3.	Çoklu Görünüm Modelleri Kullanılarak Nesne Takibi.....	34
3.3.	Siluet Takibi	37
3.3.1.	Şekil Karşılaştırmacılar	38
3.3.2.	Sınır Takibi.....	38
3.3.2.1.	Durum Uzay Modelli Kullanarak Nesne Takibi	38
3.3.2.2.	Sınır Enerji Fonksiyonunun Direk Minimizasyonu Yöntemiyle Nesne Takibi	39
4.	İNSANSIZ HAVA ARAÇLARI.....	40
4.1.	Sabit Kanatlı ve Döner Kanatlı İHA Tipleri	41
4.2.	İHA'ların Özellikleri	42
4.2.1.	Ağırlık	42
4.2.2.	Dayanıklılık ve Menzil	42
4.2.3.	Uçuş Yüksekliği	43
4.2.4.	Kanat Ağırlığı.....	43
4.2.5.	Motor Tipi	44
4.3.	İnsansız Hava Aracı'nın Seçilmesi.....	44
4.4.	AR.Drone	44
4.4.1.	AR.Drone'un Mekanik Yapısı	45
4.4.2.	Motor Tipi	46
4.4.3.	Elektronik Hız Kontrolcüsü.....	47
4.4.4.	Lipo Bataryalar.....	47
4.4.5.	İvmeler, Jiroskoplar ve İşlemciler	47
4.4.6.	Anakart	48

4.4.7.	Kameralar	49
4.4.8.	Ultrasonik Yükseklikölçer.....	50
4.5.	AR.Drone'nun Hareket Kontrolü	51
4.5.1.	Yükselme Hareketi	52
4.5.2.	Yuvarlanma Hareketi	52
4.5.3.	Yunuslama Hareketi	53
4.5.4.	Yönelme Hareketi.....	54
4.5.5.	AR.Drone Haberleşmesi.....	54
5.	SÜREKLİ UYARLAMALI ORTALAMA KAYMA ALGORİTMASI İLE HAREKETLİ NESNE TAKİBİ.....	56
5.1.	OpenCV Yazılımı.....	56
5.2.	AR.Drone ile Bağlantı Kurma.....	56
5.3.	Sürekli Uyarlamalı Ortalama Kayma Algoritması	57
5.3.1.	Renk Uzayı	58
5.3.2.	SUOK Algoritmasının Çalışma Prensibi.....	60
5.3.3.	Uygulamanın Gerçekleştirilmesi.....	64
6.	SONUÇLAR.....	72
7.	KAYNAKLAR.....	74
	ÖZGEÇMİŞ.....	81

ÖZET

Nesne takibi, kamera kullanılarak elde edilen ardışık video dizilerindeki hareketli bir nesnenin görüntülerdeki konum bilgisinin tam olarak belirlenmesi işlemidir. Takip edilecek nesnenin hareketindeki değişiminin anlaşılması, boyutunun değişmesi, ortam ışığının değişmesi ve kameranın hareketli olması sebebi ile nesne takibi zor bir problem olarak karşımıza çıkmaktadır. İnsansız hava araçları, son zamanlarda sivil ve askeri amaçlarla birçok alanda kullanımı artan ve bilimsel olarak önemli bir araştırma alanı oluşturan uygulamalardandır. Farklı modelleri bulunan insansız hava araçlarından dört rotorlu tipi manevra kabiliyeti ve üç farklı ekseninde hareket edebilme gibi belirli avantajlarıyla ön plana çıkmaktadır. Bu tez çalışmasında dört rotorlu insansız hava aracı ile bahsedilen nesne takibi problemlerini çözebilen hareketli nesne takibi uygulaması gerçekleştirilmiştir. Yapılan uygulamada dört rotorlu insansız hava aracı olarak AR.Drone kullanılmıştır. Uygulamanın gerçekleştirilmesinde OpenCV kütüphanesinin 2.4.9 sürümü ile C programlama dili Microsoft Visual Studio 2010 ortamında kullanılmıştır. AR.Drone'un ön kamerasından gerçek zamanlı olarak alınan görüntülerle 640*360 çözünürlükte ve 30 fps hızında hareketli nesne takibi işlemi gerçekleştirilmiştir.

Takip işlemi gerçek zamanlı olarak iki aşamada yapılmıştır. İlk olarak sabit kamera kullanılarak sürekli uyarlamalı ortalama kayma algoritması ile nesne takibi işlemi tamamlandıktan sonra ikinci aşamada ise aynı algoritma kullanılarak takip edilecek nesnenin hareketli bir platform olan AR.Drone dört rotorlu insansız hava aracı ile takibi gerçekleştirilmiştir. Gerçek zamanlı olarak gerçekleştirilen bu yöntem ile takip edilecek hareketli nesne seçilerek takip işlemi başlatılır. Daha sonra hareketli nesnenin AR.Drone'a olan uzaklığından bağımsız olarak takip işlemi sürdürülmektedir. Yapılan çalışma ile mobil bir platform olan dört rotorlu insansız hava aracı ile takip edilen nesnenin boyutunun değişmesine ve ortam ışığının değişmesine karşılık nesne takibi başarılı bir şekilde gerçekleştirilmiştir.

Anahtar Kelimeler: Hareketli Nesne Takibi, Dört Rotorlu İnsansız Hava Aracı, Sürekli Uyarlamalı Ortalama Kayma Algoritması

SUMMARY

TRACKING OF MOVING OBJECT USING CONTINUOUSLY ADAPTIVE MEAN SHIFT ALGORITHM WITH QUADROTOR UNMANNED AIR VEHICLE

Object tracking is the process of generating the trajectory of an object over time by discovering its exact position in every consecutive video frames. It is a challenging problem due to tracked object motion and size changes, illumination changes and egomotion. In this thesis, an approach that is able to cope with those problems using unmanned air vehicle was applied. Unmanned air vehicles has been an increasing field of research in both civilian and military applications. Among its different models, quadrotors have advantages such as high maneuverability and moving in three directions. The unmanned air vehicle used in this thesis is AR.Drone. The application is implemented using version 2.4.9 of OpenCV library with C programming language in Visual Studio 2010 environment. The application is capable of tracking moving objects by using the front camera of AR.Drone with resolution of 640x360 and at a frame rate of 30 fps.

Tracking moving object is carried out in two stages. Firstly, Continuously Adaptive Mean Shift Algorithm on a static camera is applied to track the moving object and secondly AR.Drone front camera was used to carry out tracking a moving object with the same algorithm in real-time. Object tracking process is carried on independently from the distance between AR.Drone and the tracked object. In this thesis, object tracking is carried out successfully with AR.Drone which is a mobile platform despite of changes in object's size and illumination.

Key Words: Moving Object Tracking, Quadrotor Unmanned Air Vehicle, Continuously Adaptive Mean Shift Algorithm

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Nesne gösterimleri	8
Şekil 2.2. Turuncu renkli nesneye ait renk olasılık dağılımı	12
Şekil 2.3. Canny Kenar Bulma yöntemi uygulaması	12
Şekil 2.4. Optik akış yönteminde nesneye ait yön doğrultuları[31].....	13
Şekil 2.5. Filtre yöntemi uygulaması (a) orijinal imge, (b) filtre uygulanmış imge[31].....	14
Şekil 2.6. (a) orijinal imge (b) ortalama kayma bölütlemesi uygulanmış imge (c) normalize edilmiş kesimler yöntemi uygulanmış imge	19
Şekil 3.1. Nesne takibi yöntemlerinin sınıflandırılması [1]	26
Şekil 4.1. Farklı tip İHA'lar	41
Şekil 4.2. AR.Drone gövde yapısı.....	45
Şekil 4.3. AR.Drone'un donanımsal bileşenleri: a) iç ortam kabini, b) dış ortam kabini, c) orta çapraz d) gövde, e) şaftlar, f) Li-on batarya, g) ultrasonik alıcı-verici, h) navigasyon kartı, ı) Anakart, i) motor+EHK devresi j) pervaneler [85]	49
Şekil 4.4. Ultrasonik mesafe ölçerin donanımsal yapısı	50
Şekil 4.5. AR.Drone hareket doğrultuları.....	51
Şekil 4.6. Yükselme hareketi.....	52
Şekil 4.7. Yuvarlanma hareketi	53
Şekil 4.8. Yunuslama hareketi.....	53
Şekil 4.9. Yönelme hareketi	54
Şekil 5.1. RGB renk uzayı.....	58
Şekil 5.2. HSV renk uzayı	59
Şekil 5.3. Arama penceresinin maksimum yoğunluk dağılım bölgesine hareketi	61
Şekil 5.4. Kameradan alınan görüntülerde nesnenin bulunduğu noktaya ait konum bilgisinin tespit edilmesi yöntemi	64
Şekil 5.5. Sabit Kamera ile Sarı top takibi uygulaması sonuçları	65
Şekil 5.6. Sabit Kamera ile yeşil renkli fosforlu kalem takibi uygulaması sonuçları	66
Şekil 5.7. AR.Drone ile hareketli nesne takibi için yapılan uygulama görüntüsü	67
Şekil 5.8. AR.Drone ile hareketli nesne takibi uygulamasına ait akış diyagramı	68

Şekil 5.9. AR.Drone ile geri yayılım imgeleri elde edilmiş hareketli nesne takibi sonuçları	69
Şekil 5.10. AR.Drone ile geri yayılım imgeleri elde edilmiş hareketli nesne takibi sonuçları	70
Şekil 5.11. AR.Drone ile hareketli nesne takibinin dışarıdan izlenmesi	71

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. Nesne bulma kategorileri.....	15
Tablo 3.1. Nesne takip yöntemleri ile ilgili literatürde bulunan örnek çalışmalar	26
Tablo 3.2. Geometrik model tabanlı nesne izleyicilerin karşılaştırılması	36

KISALTMALAR

PTZ	: Pan Tilt Zoom
CAMSHIFT	: Continuously Adaptive Mean Shift
RGB	: Red Green Blue
HSV	: Hue Saturation Value
KLT	: Kanade Lucas Tomasi
SIFT	: Scale Invariant Feature Transform
HMM	: Hidden Markov Model
DVM	: Destek Vektör Makineleri
İHA	: İnsansız Hava Aracı
SDK	: Software Development Kit
Wi-Fi	: Wireless Field
FDAM	: Fırçasız Doğru Akım Motoru
MEMS	: Micro Electro Mechanical System
RAM	: Read Access Memory
CMOS	: Complementary Metal Oxide Semiconductor
QVGA	: Quarter Video Graphics Array
ESSID	: Extended Service Set Identification
DHCP	: Dynamic Host Configuration Protocol
IP	: Internet Protocol
TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol
OK	: Ortalama Kayma
SUOK	: Sürekli Uyarlamalı Ortalama Kayma

SEMBOLLER LİSTESİ

$\mathbf{X}_T$	: Durum bilgisi
$\mathbf{W}^t$	: Beyaz gürültü
$\mathbf{K}$	: Kalman kazancı
$\mathbf{M}$	: Ölçme matrisi
$\mathbf{Q}$	: Nesnenin histogramı
$\mathbf{P}$	: Nesnenin pozisyonu
s	: Nesnenin arama alanı
l	: Nesnenin bulunduğu alanının uzunluğu
w	: Nesnenin bulunduğu alanın genişliği
$\mathbf{M}_{00}$	: Sıfırıncı moment
$\mathbf{M}_{10}, \mathbf{M}_{01}$	: Birinci Momentler
$\mathbf{M}_{20}, \mathbf{M}_{02}$	: İkinci Momentler
x_c	: Arama penceresi merkezinin x koordinatı
y_c	: Arama penceresi merkezinin y koordinatı

1. GİRİŞ

Günümüzde robotik alanında birçok bilgisayarlı görme uygulaması bulunmaktadır. Tıbbi uygulamalarda yüksek doğrulukta cerrahi cihaz tasarımından, askeri uygulamalarda hedef izleme ve gözetim sistemlerine kadar geniş bir uygulama alanında kullanılmaktadır. İnsan yardımı olmadan karmaşık işleri yürüten makinaların yapılması fikri birçok alanda büyük gelişmelere sebep olmuştur. Uzaktan ya da otonom kontrol edilebilen araçlar ile insan için güç olan alanlara müdahale edilebilir. Böylece herhangi bir görev yerine getirilirken can kaybının önüne geçilmiş olur.

Bilgisayarlı görme, robotikte otonomun uygulanması için kullanılan çoğu teknikten biridir. Bilgisayarlı görme bir imgeden ya da bir dizi imgeden önemli bilgilerin çıkartılması için insan duyularını canlandırmayı amaçlamaktadır. İmgelerden çıkartılan bilgi farklı fonksiyonları gerçekleştirmek için kullanılabilir. Bu fonksiyonlar uygulanacak algoritmanın seçilmesi ya da bir aracın kaza yapmadan hangi yönde gideceğinin belirlenmesi olabilir. Bilgisayarlı görme görüntü dizilerinden istenilen bilginin elde edilmesi için çeşitli görüntü işleme yöntemlerini kullanır. Bu yöntemlerle imgeden elde edilen bilgiler genellikle insan görme duyuları ile elde edilmesi mümkün olmayan bilgilerdir. Bu insan yeteneklerini aşan makinaların kullanımına dayandırılabilir.

Gözetim alanında birçok bilgisayarlı görme uygulaması bulunmaktadır. Kapalı devre televizyon sistemleri yaygın olarak havaalanı ve büyükelçilik gibi yüksek güvenlik gerektiren ortamların gözlemlenmesi için kullanılır. Bu sistemler havaalanına giriş yapan kişilerin gözlemlenmesi ve bırakılan sahipsiz bagajların tespiti için bilgisayarlı görme tekniklerini kullanır ve dar bir ortamda kolay bir şekilde uygulanabilir. Ancak doğa koruma alanları gibi geniş alanlarda daha zor olmaktadır. Doğa koruma alanlarının büyüklüğü kapalı devre televizyon sistemlerine benzer gözetim sistemlerinin kullanımını kullanılabilecek donanımların yüksek maliyet, bakımı ve ihtiyaç duyulan insan kaynağından dolayı mümkün kılmamaktadır. Bu sorunun çözülebilmesi için insansız hava araçlarının otonom kullanılmasıyla bu geniş alanlar kısa sürede gözden geçirilebilir. Bilgisayarlı görme teknikleri kullanılarak ortamdaki tehditlerin tespiti yapılabilir. Ayrıca insansız hava araçları tespit edilen tehditlerin önüne geçilmesi için yetkili mercilere yardım etmek amacıyla bu tehditleri takip edebilir.

Nesne takibi bilgisayarlı görme konularından biridir. Literatürü incelediğimiz zaman son otuz yıl içerisinde konuyla ilgili farklı çalışmalar yapıldığını görülmektedir. Şüpheli şahısların teşhisinde, trafik kontrolü ve düzenlenmesinde, navigasyon sistemlerinde ve özellikle insansız gözetim; insansız hava araçları gibi bilgisayarlı görme alanının ilgilendiği önemli uygulamalarda nesne takibinin öne çıktığını görmekteyiz. Nesne takibi; nesneyi temsil eden özelliklere, nesnenin hareketinin, görünümünün ve şeklinin nasıl modellendiğine göre farklılıklar göstermektedir.

Nesne takibi süresince ve öncesindeki adımlarda kullanılabilecek yöntemler, yapılan çeşitli çalışmalarla incelenmiştir. Bu amaçla yapılan çalışmalarda, uygulamaya bağlı olarak seçilmesi gereken nesne gösterimleri, takip için kullanılacak özelliklerin seçimi, nesne bulma yöntemleri ve nesne takibi amacıyla kullanılabilecek yöntemlerden bahsedilmiştir [1].

Literatürde, gerçek zamanlı video görüntüleri üzerinde değil, önceden kaydedilmiş görüntüler üzerinde nesne takibi yapan çalışmalar da mevcuttur. Soon-Yong ve Inho çalışmalarında, takip edilerek bulunan hareketli nesnelerin silinmesini amaçlamışlardır. Kullanılan kameranın hareketli olmasından dolayı art arda gelen görüntü kareleri birleştirilerek bir mozaik oluşturulmuş, böylece kamera hareketi elimine edilmiştir [2].

Uçan bir platformdan alınan kayıtlı video verisi üzerinde nesne takibi yapan çalışmalar da mevcuttur. Isaac ve Gerard tarafından yapılan çalışmada, hareketli nesneleri takip etmek için iki adımlı bir yaklaşım sunulmuştur. Öncelikle gözlemleme platformunun hareketinin neden olduğu görüntü akışı bulunmuş ve elimine edilmiş, daha sonra iki boyutlu şablon kullanılarak hareketli nesne takip edilmiştir [3].

Literatürdeki bazı çalışmalar trafik planlama, düzenleme ve izleme hususunda yapılmıştır. Qi Zang ve Reinhard Klette tarafından yapılan çalışmada trafik düzenleme uygulamalarının performansının yükseltilmesi amaçlanmıştır. Takip edilen nesnenin hareketi Kalman filtresi kullanılarak belirlenmiştir. Geliştirilen sistemde sabit bir kameradan alınan görüntüler üzerinde çalışılmaktadır. Çalışılan görüntüler üzerindeki arka planın sabit olması sayesinde, hareketli nesneler arka plandan ayrılıp izlenmiştir. Sabit kamera ile çalışan bu tarz uygulamalarda aktif bir nesne takibi söz konusu olmamaktadır. Bu sistemler daha ziyade trafik izleme ve düzenleme, kalabalık alanlardaki olağan dışı durumları kontrol etme gibi uygulamalarla bütünleşmiş olarak çalışır [4].

Kim ve Cho tarafından yapılan çalışmada, otonom mobil robot üzerine monte edilmiş kameradan alınan görüntülerle, nesne bulma ve takibi gerçekleştirilmiştir. Bu

sistemde hedef nesnenin ve kameranın hareketli olması, takip işlemini zorlaştırmıştır. Bu problemi aşmak amacıyla kamera hareketinin doğrultusu ve hızı kullanılarak takip edilecek nesnenin hareket doğrultusu ve hızı hesaplanmıştır [5].

Nesne takibi amacıyla kullanılan bir yöntem de kenar takibi algoritmasıdır. Hareketli kameradan alınan video görüntülerini kullanan sistemde, doku ve renk özelliklerinin olasılık dağılım fonksiyonlarını esas alan Bayes çıkarımı kullanılmıştır [6].

Esnek yapılı nesnelerin takibi üzerine de çalışmalar yapılmıştır. Comaniciu ve Meer tarafından yapılan çalışmada, gerçek zamanlı olarak hareketli kameradan alınan görüntüler üzerinde esnek yapılı nesnelerin takibi gerçekleştirilmiştir [7]. Algoritma takip edilecek nesnenin art arda gelen her bir görüntüde Ortalama Değer Kayma (Mean-Shift) iterasyonları ile yerinin belirlenmesine dayanır. Comaniciu ve Meer tarafından yapılan bir başka çalışmada takip işlemi esnek yapılı hedef nesneyi temsil eden bir geometrik şekil ve bu şekil içerisindeki özellik dağılımı kullanılarak gerçekleştirilmiştir. Yapılan çalışmada Bhattacharyya uzaklığı ile benzerlik hesaplanmıştır. Ayrıca optimizasyon için ortalama değer kayma algoritması kullanılmıştır [8].

Bazı çalışmalarda nesne takibinde karşılaşılan problemler üzerinde durulmuş ve bu problemlere çözümler üretilmeye çalışılmıştır. Alper ve Mubarak tarafından yapılan çalışmada, görüntü özelliklerinin değişimine ve hedef nesnenin tamamının görüntülenememesine rağmen, hedef nesne, enerji fonksiyonu kullanılarak takip edilmiştir [9].

Matej ve Martin çalışmalarında insan gözünün takibini HD kamera kullanarak gerçekleştirmiştir [10]. Bu çalışmada gömülü platform üzerinde çalışan Haar özelliği tabanlı Cascade sınıflandırıcı (Haar feature-based Cascade classifier) nesne algılayıcı göz algılama sistemi için kullanılmıştır. Ayrıca, OpenCV kütüphanesinde nesne algılayıcı uygulamasının nasıl yapıldığı anlatılmıştır.

Hedef nesnenin renk ve şekil bilgisine bağlı gerçek zamanlı bir sistem, düzensiz olarak hareket eden bir nesneyi algılamak ve takip etmek için Traun ve Cheolkeun tarafından yapılan çalışmada kullanılmıştır. Bu çalışmada veriler gerçek zamanlı bir kameradan belli oranda resim karesi alınarak işlenmiştir. Daha sonra her bir kare, önerilen metot ile hedef nesneyi art arda gelen her görüntü karesinde algılamak ve izlemek için bazı işlemlere tabi tutulmuştur. Son olarak değişen kontrol işaretine bağlı hedef pozisyonu, PTZ kamerayı kontrol etmek için kullanılmıştır [11].

Fahad ve Pierre-Yves tarafından yapılan çalışmada ise CAMSHIFT algoritması kullanılarak kişi takibi yapılmıştır. Bu çalışmada kişi takibi oklüzyon durumunda bile başarılı bir şekilde gerçekleştirilmiştir [12].

Woo-han ve Dohyung yapmış oldukları çalışmada düşük güç tüketimine sahip taşınabilen cihazlar üzerine yerleştirilen kamera ile nesne takibi yapmışlardır. Hedef nesne seçimi aşamasında sel-taşma tabanlı kesimleme algoritması ve takip aşamasında ise ortalama kayma algoritması kullanılmıştır [13].

İnsansız hava araçlarının görü tabanlı kontrolü son yıllarda önemli bir araştırma alanı olmuştur. Şimdiye kadar yapılan çalışmalarda otonom iniş, kalkış, dengeleme ve navigasyon işlemleri için daha ziyade görü tabanlı kontrol metotları üzerinde durulmuştur. Görsel bilgi, genellikle hedefin bilinen bir modeli ya da ortamı anahtar imgeler, ya da hareket tahmini için özellik noktaları ya da optik akış hesaplamaları kullanılarak elde edilmiştir. Görsel bilginin güvenilirliği görü tabanlı kontrol işleminin iyi anlaşılması açısından önem arz etmektedir. Bu tür bir sistemden beklenen hedef nesnenin hızlı hareketi, oklüzyon, imge gürültüsü, ortam ışığının değişimi, poz değişimleri ve imge bulanıklığı gibi uygun olmayan ortam koşullarına rağmen hedef nesnenin yerinin güçlü bir şekilde çıkartılmasıdır [14, 15,16, 17, 18, 19, 20].

Renk histogramları gibi yoğunluk tabanlı tanımlayıcılar görünüm değişimlerine karşı dayanıklı olması ve düşük hesaplama karmaşıklığından dolayı cazip bir alternatif olmaktadır [21].

1.1. Tezin Amacı

Bu tezin amacı bilgisayarlı görme teknikleri kullanılarak hareketli bir nesnenin takip edilmesi için insansız hava araçlarının uygulanabilir olduğunu test edebilen bir sistemin oluşturulmasıdır. İnsansız hava aracı takip işlemini kendi devre kartı üzerinde bulunan kamerayı kullanarak alınan video görüntülerinin bilgisayar ortamında değerlendirilmesi ile yürütülmüştür. Gerçek zamanlı olarak alınan bu video görüntülerine bilgisayar ortamında görüntü işleme teknikleri uygulanarak elde edilen sonuca göre insansız hava aracına uçuş komutları gönderilmiştir. Burada takip yönteminin uygulanması için öne çıkan kısıtlama, takip yönteminin gerçek zamanlı olarak etkin bir şekilde kullanılmasıdır. Bu tez çalışmasında kullanılacak yöntem ile dört rotorlu insansız hava aracı ile nesne takibi gerçekleştirilebilen bir sistem tasarlanmıştır.

Bu çalışmada ortam koşullarının elverişli olmaması nedeniyle oluşan takip problemi hedef nesnenin anlaşılabilir olduğunu varsaymaktadır. Takip işlemi dört rotorlu insansız hava aracı olan AR.Drone ile gerçekleştirilmiştir. AR.Drone burada ön kamerasından alınan görüntüler yoğunluk tabanlı ortalama kayma algoritması kullanılarak takip edilen hareketli nesnenin oryantasyonu ve ölçeğinin değişmesine rağmen görüntü dizileri boyunca nesnenin yerinin belirlenmesi sağlanmakta ve AR.Drone ile takip edilmesi gerçekleştirilmektedir.

1.2. Tezin Organizasyonu

Tezin ikinci bölümünde nesne takibinde karşılaşılan zorluklar, kullanılan özellikler ve parametreler hakkında bilgi verilmiştir.

Üçüncü bölümde genel olarak nokta, çekirdek ve silüet tabanlı takip yöntemleri hakkında bilgi verilmiştir.

Dördüncü bölümde literatürde mevcut olan insansız hava araçları, onların özellikleri ve uçuş yöntemleri hakkında bilgi verilmiştir. Mevcut insansız hava araçları yanı sıra, tez çalışmasında kullanılacak olan özerk hava aracından da bahsedilmiştir. Farklı insansız hava araçları, uygulamaları ve tez çalışmasında kullanılan dört rotorlu hava aracı incelenmiştir. Dört rotorlu insansız hava aracında kullanılan bileşenler ve bu bileşenlerin işlevselliğinden bahsedilmiştir.

Beşinci bölümde tez çalışması için yapılan uygulamada kullanılan açık kaynak kodlu OpenCV kütüphanesi, bilgisayar ile AR.Drone haberleşmesi ve sürekli uyarlamalı ortalama kayma algoritması ile ilgili bilgiler verilmiştir.

Son bölümde ise sonuçlar tartışılıp sonraki çalışmalar hakkında bilgiler verilmiştir.

2. NESNE TAKİBİNDE KULLANILAN PARAMETRELER

Nesne takibi bilgisayarlı görme alanında önemli bir çalışma konusudur. Güçlü bilgisayarların yaygınlaşması, yüksek kaliteli ve ucuz video kameraların mevcut olması ve otomatik video analizi ihtiyacının artması nesne takibi algoritmalarına olan ilgiyi önemli bir ölçüde arttırmıştır. Video analizinde önemli temel üç aşama vardır. Bunlar; ilginç hareketli nesnelerin tespiti, bu nesnelerin tüm resim kareleri boyunca takibi ve bu nesnelerin davranışını tanımak için takip eden nesnenin analizidir.

En basit ifadeyle nesne takibi, sahnede hareket eden nesnenin imge düzleminde yörüngesinin tahmin edilmesi problemi olarak tanımlanabilir. Bir başka ifadeyle, nesne izleyici bir videonun farklı çerçeveleri boyunca takip edilen nesneye sürekli etiket atamaktadır. Nesne takibi, görüş alanına bağlı olarak, nesne izleyici oryantasyon, alan ya da nesnenin şekli gibi nesne merkezli bilgi gerektirebilir.

Nesne takibi uygulamalarında bilinmesi gereken bazı ölçütler bulunmaktadır. Bu ölçütler takip edilecek nesneye, ortam ışığına ve yapılan uygulamaya göre belirlenmektedir. Bunlar esas itibarıyla nesne gösterimi, takip işlemini gerçekleştirmek amacıyla kullanılan özellik ve kullanılacak metottur.

Nesne takibinde karşılaşılan zorluklar aşağıdaki gibi sıralanabilmektedir [22].

- Ortam ışığının değişmesi
- Nesnenin yüzeyinin değişmesi
- Şeffaflık
- Nesnenin şeklinin değişmesi
- Nesnenin hareketinin uyumlu olmaması
- Nesnenin bulunduğu ortamın karmaşıklığı
- Düşük sahne kontrastı
- Oklüzyon (Takip edilen nesne ile kamera arasına başka bir nesnenin girmesi)
- Kameranın hareketli bir platform üzerinde olması
- Yakınlaştırma/Uzaklaştırma
- Görüntü dizilerinin uzunluğu
- 3 boyutlu uzayın 2 boyutlu imgede yansıtılması sonucu bilgi kaybı
- İmgedeki gürültü
- Gerçek zamanlı işlem gereksinimleri

Nesne görünümü veya hareketi ile ilgili bazı kısıtlamalar yapılarak takip işlemi kolaylaştırılabilir. Örneğin; neredeyse tüm takip algoritmaları nesne hareketinin ani bir değişimi olmaksızın düzgün olduğunu varsaymaktadır. Nesne hareketi sabit bir hızla veya önceki bilgiye bağlı olarak da sabit ivmeli olarak sınırlandırabilmektedir. Ayrıca nesnenin boyutu ve sayısı hakkında önceki bilgi ya da nesne görünümü ve şekli problemi basitleştirmek amacıyla kullanılabilir.

Şu ana kadar yapılan çalışmalarda nesne takibi ile ilgili çok sayıda çözüm önerilmiştir. Yapılan çalışmalara bakıldığında esas itibarıyla birbirinden aşağıdaki sorulara yaklaşım tarzı bakımından farklılık göstermektedir.

- I. Takip işlemi için ne tür nesne gösterimi uygundur?
- II. Hangi imge özellikleri kullanılmalıdır?
- III. Nesnenin hareketi, şekli ve görünümü nasıl modellenmelidir?

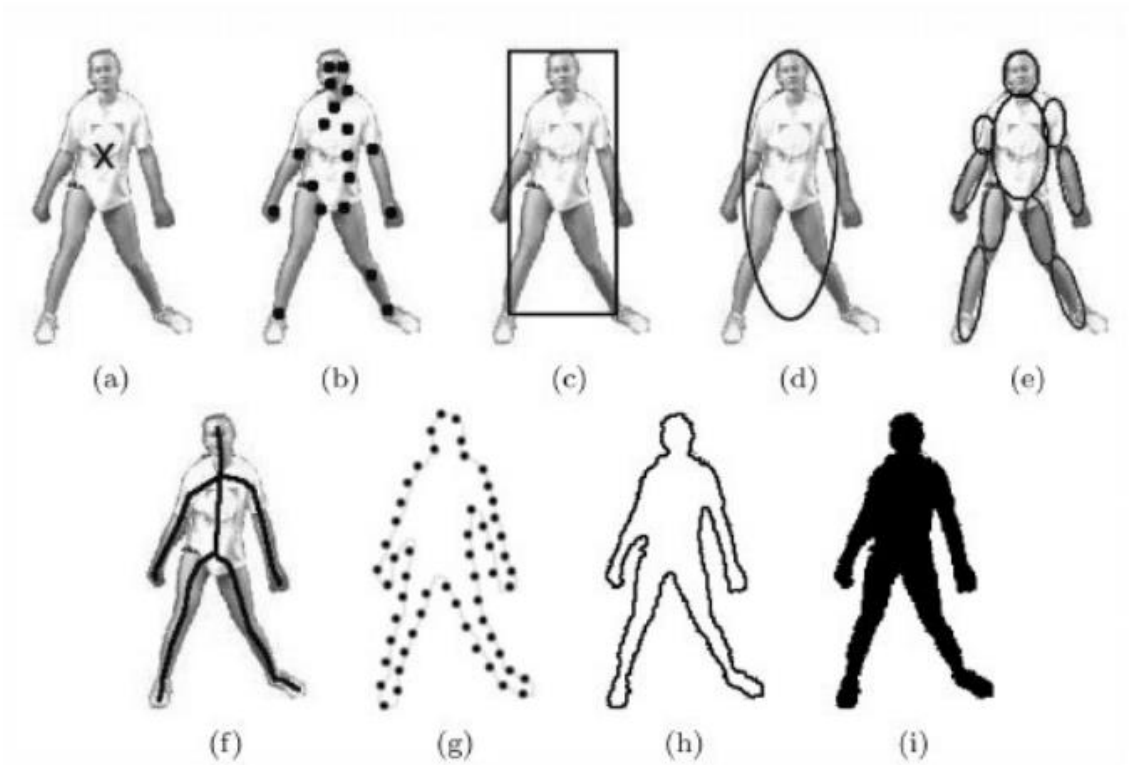
Bu üç temel soruya verilecek cevaplar, takip işleminin nasıl bir ortamda gerçekleştirildiği ve hangi takip işlemine ait bilginin araştırıldığına bağlıdır. Çok sayıda takip metodunda bu sorular çeşitli senaryolar için çözüme kavuşturulmak amacıyla sunulmuştur. İlk olarak nesne için uygun bir gösterim tanımlanmalıdır. Daha sonra nokta, ilkel geometrik şekiller ve nesne hatları gibi genel nesne şekli gösterimleri tanımlanarak nesneyi takip etmek için kullanılacak imge özelliklerinin seçimi yapılmalıdır.

2.1.Nesne Gösterimleri

Nesne takibi aşamasında önemli rol oynayan seçimlerden biri takip edilecek nesnenin sembolize edileceği şekildir. Uygulamanın amacına göre takip edilecek nesne değişiklik göstermekte, takip edilecek nesneye göre de ilgili gösterim seçilmektedir. Örneğin trafikte bir araç, denizde bir gemi, havada bir uçak, yolda bir insan, sahada bir top, akvaryumda bir balık takip etmek için kullanılan nesneler arasında sayılabilir. Nesneyi sembolize etmek için, takip edilecek nesnenin şekil ve hareket yapısına uygun olan bir şekil seçilir.

2.1.1. Noktalar ile Gösterim

Nesne, Şekil 2.1 (a) ve (b)'deki gibi merkezinde yer alan bir nokta ile veya noktalar kümesi ile gösterilir. Nokta gösterimi görüntüde az yer kaplayan nesneleri takip etmek için uygundur.



Şekil 2.1 Nesne gösterimleri

2.1.2. Geometrik Şekil ile Gösterim

Nesne, Şekil 2.1 (c) ve (d)'de görüldüğü gibi dikdörtgen veya elips gibi bir geometrik şekil ile gösterilir. Bu şekillerdeki gösterimler için nesnenin hareketi genellikle ötelemeli, doğrusal veya izdüşümsel olmaktadır. Geometrik şekil ile gösterim genellikle katı, esnek olmayan nesneler için uygun olmakla birlikte, esnek nesneleri takip etmek için de kullanılır.

2.1.3. Nesne Silueti ve Çevre Çizgisi ile Gösterim

Şekil 2.1 (g) ve (h)'deki gibi çevre çizgisi gösterimi nesnenin sınır çizgilerini tanımlamaktadır. Şekil 2.1 (i)'deki çevre çizgisinin içinde kalan bölge ise nesnenin silueti olarak tanımlanır. Çevre çizgisi ve siluet gösterimleri karmaşık ve esnek yapıya sahip nesnelerin takibinde kullanımı uygundur.

2.1.4. Eklemlı Şekil Modelleri ile Gösterim

Eklemlı nesneler birbirine eklem ile bağı olan parçalardan oluşur. Örneğın insan; gövdesi, kolları, bacakları, kafası, elleri ve ayaklarının eklemlerle birbirine bağlanmasından oluşan bir canlıdır. Bu gösterimde parçalar arası ilişkiler kinematik hareket denklemleri ile çıkartılabilir. Eklemlı bir nesnenin gösteriminde, Şekil 2.1 (e)'de olduğu gibi nesnenin eklemler ile bağlanan her bir parçası silindir veya elips şeklinde modellenenebilir.

2.1.5. İskelet Model ile Gösterim

Nesnenin iskeleti, Şekil 2.1 (f)'deki gibi nesnenin silüetine orta eksen dönüşümü uygulanarak çıkartılabilir. Bu model genellikle nesne tanımada şekil gösterimi olarak kullanılır. İskelet gösterimi, hem eklemlı hem de esnek yapılı nesnelerin gösterimi için kullanılabilir.

Nesnelerin görünüm özelliklerini temsil etmek için bir takım yöntemler bulunmaktadır. Burada dikkat edilmesi gereken husus şekil gösterimleri, görünüm gösterimleri ile birleştirilebilmektedir. Nesne takibi bağlamında bazı görünüm gösterimleri aşağıdaki gibi sıralanabilir.

- Nesne görünümünün olasılık yoğunluğu: Olasılık yoğunluğu nesne görünümünün Gaussian ve Gaussian karışımı gibi parametrik olduğunu ya da parzen penceresi ve histogram gibi parametrik olmadığını tahmin etmektedir. Nesne görünümü özelliklerinin olasılık yoğunlukları (renk, doku) şekil modelleri ile belirtilen imge bölgelerinden hesaplanabilmektedir. (elips veya çevre çizgisi sınırları içerisinde kalan bölge)

- Şablonlar: Şablonlar basit geometrik şekiller ya da silüetler kullanılarak sembolize edilirler. Şablon kullanmanın avantajı mekânsal ve görünüm ile ilgili bilgi taşımasıdır. Ancak şablonlar nesne görünümünü tek bir açıdan elde ederek kodlarlar. Bu sebep ile şablonlar sadece takip işlemi boyunca pozisyon önemli derecede değişmeyen nesneleri takip etmek için uygun olmaktadır.
- Aktif görünüm modelleri: Aktif görünüm modelleri nesne şekli ve görünümünün eş zamanlı olarak modellenmesi ile elde edilir. Nesne şekli bir dizi işaret ile ifade edilir. Çevre çizgisi gösterimine benzer olarak, bu işaretler nesne sınırlarında ya da alternatif olarak nesnenin sınırları içerisinde kalan bölge içerisinde kalabilmektedir. Her bir işaret için bir görünüm vektörü renk, doku ya da gradient genliği şeklinde kaydedilir. Aktif görünüm modelleri hem şekil hem de şeklin görünümünün bir dizi örnek kullanılarak öğrenildiği eğitim aşamasına gereksinim duyarlar.
- Çoklu görünüm modelleri: Bu modeller nesneyi farklı açılardan kodlamaktadır. Farklı nesne görüntülerini göstermek için verilen görüntülerden bir alt uzay oluşturulur. Alt uzay yaklaşımları örneğin temel bileşen analizi ve bağımsız bileşen analizi hem şekil hem de görünüm gösterimi için kullanılmıştır.

Özetle, nesne gösterimleri ve takip algoritmaları arasında güçlü bir ilişki vardır.

Nesne gösterimleri genellikle takip işleminin gerçekleştirileceği ortama bağlı olarak seçilmektedir. Bir imgede çok küçük olarak görülen nesnelerin takibi için nokta gösterimi genelde uygun olmaktadır. Örneğin Bettencourt ve Somers tarafından yapılan çalışmada nokta gösterimini hareketli bulaşık dizisinde seedleri takip etmek için kullanmıştır [23]. Benzer şekilde Veenman ve Reinders tarafından yapılan çalışmada uzakta bulunan kuşları takip etmek için nokta gösterimi kullanılmıştır. Geometrik olarak dikdörtgen ya da elipse yakın şekillere sahip olan nesneler için ilkel geometrik şekil gösterimleri daha uygun olmaktadır [24]. Comaniciu ve Ramesh tarafından yapılan bir çalışmada, görünümü modellemek için elips şekli gösterimi kullanılarak bu eliptik bölgeden hesaplanan renk histogramından faydalanılmıştır [8]. Shafique ve Mubarak tarafından yapılan çalışmada görünümün ifade edilmesi için özvektörler kullanılmıştır. Özvektörler dikdörtgensel nesne şablonlarından elde edilmiştir. İnsan gibi karmaşık denilebilecek şekilli nesnelerin takibi için bir kontur ya da bir silüet tabanlı gösterimlerin uygun olduğu görülmüştür [25]. Haritaoglu ve Harwood çalışmasında gözetim uygulamalarında nesne takibi için silüet kullanılmıştır [26].

2.2. Özellik Seçimi

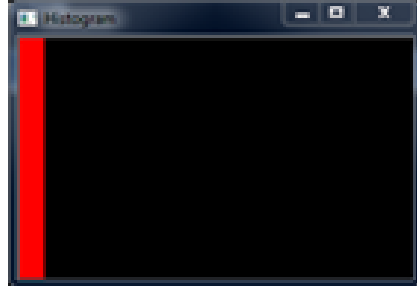
Nesne takibinde özellik seçimi kritik bir rol oynar. Seçilecek özelliğin veya özelliklerin takip edilecek nesneyi ayırt edici kılması arzu edilir. Görsel özelliklerin kullanılmasındaki en önemli sebep her bir nesnenin kendine özgü görsel özellikleri olmasıdır. Özellik seçimi, nesne gösterimi ile yakından alakalıdır. Örneğin renk histogram tabanlı görünüm için bir özellik olarak kullanılır, bunun yanında çevre çizgisi tabanlı gösterimlerde nesne kenarları özellik olarak kullanılmaktadır. Çoğu nesne takibi algoritması bu tarz kombinasyonları kullanmaktadır [1].

2.2.1. Renk

Bir nesnenin rengi ışık kaynağının ışık tayfının güç dağılımı ve nesnenin yüzey yansıma özellikleri olmak üzere iki esas fiziksel faktörden etkilenir. Görüntü işleme çalışmalarında, rengi temsil etme amacıyla genellikle RGB (Red, Green, Blue – Kırmızı, Yeşil, Mavi) renk uzayı kullanılır. Ancak RGB uzayında renkler arasındaki farklılık insan gözü tarafından algılanan renk farklılığı ile örtüşmemektedir. RGB renk uzayı dışında L^*u^*v ve HSV (Hue, Saturation, Value – Renk tonu, Doygunluk, Değer) uzayları da kullanılmaktadır. L^*u^*v renk uzayı da algısal olarak tek düzeli bir renk uzayıdır. HSV için ise yaklaşık olarak tek düzeli renk uzayı denilebilir. Bu renk uzaylarının hepsi gürültüye duyarlıdır. Şekil 2.2 (a)'daki nesne ait renk olasılık dağılımı Şekil 2.2 (b)'de görüldüğü gibi olmaktadır. Nesne takibinde hangi renk uzayının kullanılmasının daha verimli olduğuna dair kesinlik yoktur.



(a)



(b)

Şekil 2.2 Turuncu renkli nesneye ait renk olasılık dağılımı

2.2.2. Kenar

Nesne dış çizgileri görüntü yoğunluğunda genellikle büyük değişimlere yol açar. Kenar bulma bu değişimleri belirlemek için kullanılır. Kenarların önemli bir özelliği renk özellikleri ile mukayese edildiğinde ışık değişimlerine daha az duyarlı olmasıdır. Nesnelerin dış çizgisini takip eden algoritmalar özellik olarak genellikle kenar özelliğini kullanırlar. Sonuçlarının anlaşılır olması ve doğruluğu sebebiyle en çok tercih edilen kenar bulma yaklaşımı Canny Kenar Bulma (Canny Edge Detector) yöntemidir[27]. Şekil 2.3’de orijinal imgeye Canny Kenar Bulma yöntemi uygulandığında elde edilen sonuç gösterilmiştir. Diğer kenar bulma yöntemleri Sobel, Prewitt, Gobar ve Laplace olarak sıralanabilmektedir.



(a) Orijinal resim

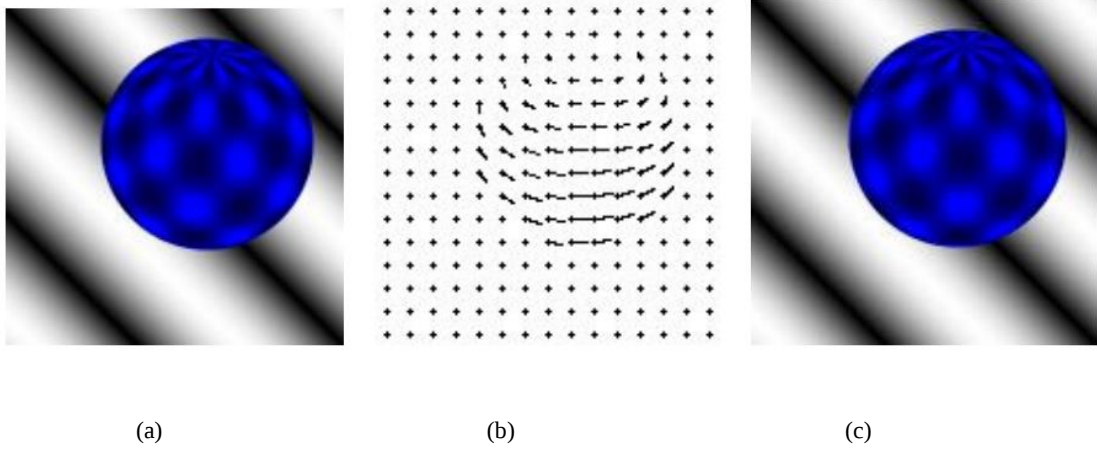


(b) Canny Kenar Bulma yöntemi uygulanmış resim

Şekil 2.3. Canny Kenar Bulma yöntemi uygulaması

2.2.3. Optik Akış

Optik akış, bir bölgedeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Arka arkaya gelen görüntüdeki eş piksellerin parlaklıkları kullanılarak hesaplanır. Optik akış, hareket tabanlı bölütleme ve takip uygulamalarında sıklıkla kullanılan bir özelliktir. Yoğunluk optik akışı hesaplamak için kullanılan popüler yöntemlerden bazıları Lucas Kanade ve Horn-Schunck yöntemleridir [28, 29, 30]. Şekil 2.4-(a)'daki top hareket ederek Şekil 2.4-(c)'deki konuma gelmiştir. Topun hareket değişimi Şekil 2.4-(b)'de verilmiştir.



Şekil2.4. Optik akış yönteminde nesneye ait yön doğrultuları [31]

2.2.4. Doku

Doku, düzgünlük ve düzenlilik gibi özellikleri ölçen yüzeyin yoğunluk değişim ölçüsüdür. Renk ile karşılaştırıldığında doku tanımlayıcılarını oluşturabilmek için filtre uygulama gibi ön işlemlere ihtiyaç duyar. Kenar özellikleri gibi doku özellikleri de ışık değişimlerine daha az duyarlıdır. Şekil 2.5 (a)'daki resmin doku özellikleri Şekil 2.5 (b) resmindeki gibidir.



(a)

(b)

Şekil 2.5. Filtre yöntemi uygulaması (a) orijinal imge, (b) filtre uygulanmış imge [31]

Uygulama alanına bağlı olarak çoğu zaman nesne takibi için kullanılan özellikler manuel olarak seçilir. Bunun yanında otomatik özellik seçimi problemi, örüntü tanımada önemli bir ilgi odağı olmuştur. Otomatik özellik seçimi metotları filtre ve kaplama metotları olmak üzere iki ayrı kategoride irdelenmiştir. Filtre metotları özelliklerin ilintisiz olduğu genel bir ölçüte dayalı özellikleri seçer. Kaplama metotları özellik alt kümesi kullanılarak sınıflandırma performansı gibi belirli bir alandaki özelliklerin kullanılabilmesine dayalı olan özellikleri seçer. Temel bileşen analizi özellik indirgemek için filtre metotlarından biridir. Temel bileşen analizi olası ilintili değişkenleri daha düşük miktardaki temel bileşenler diye isimlendirilen ilintisiz değişkenlere dönüştürmektedir. Özel bir sınıf nesnenin takibi için ayırt edici özelliklerin seçimine ait kaplayıcı metodu Adaboost algoritmasıdır. Adaboost kısmi hatalı sınıflandırıcı birleştirmeye dayalı güçlü bir sınıflandırıcının bulunması metodudur. Çok sayıda özelliğin olduğu bir durumda sınıflandırıcı her bir özellik için eğitilebilir. Adaboost algoritması algoritmanın sınıflandırma performansını maksimize eden ağırlıklı bir sınıflandırma birleştirmesi bulur. Özellik ağırlık seviyesi ne kadar yüksek olursa sınıflandırıcının ayırt edici özelliği aynı oranda artar.

Tüm özellikler içerisinde renk takip için en çok kullanılan özellik olmuştur. Comaniciu ve Ramesh tarafından yapılan çalışmada, nesne görünümünün temsil edilmesi için renk histogramı kullanılmıştır [8]. Renk özelliğinin yaygın olarak kullanılmasına rağmen çoğu renk grubu ortam ışığının değişimine karşı oldukça duyarlıdır. Bu etkinin kaçınılmaz olduğu durumlarda diğer özelliklerden nesne gösteriminin modellenmesi için faydalanılır. Cremers ve Schnorr tarafından yapılan çalışmada optik akışı kontur takibi için

özellik olarak kullanılmıştır. Alternatif olarak bu özelliklerin birleştirilmesi takip performansını geliştirmek amacıyla kullanılır [32].

2.3. Nesne Bulma

Tüm takip yöntemleri her görüntü dizisinde ya da nesne bir videoda ilk görüldüğü zaman bir nesne bulma mekanizmasına gereksinim duyarlar. Nesne bulma için genel bir yaklaşım tek bir görüntü dizisindeki verinin kullanılmasıdır. Ancak, bazı nesne bulma yöntemleri hatalı tespitlerin azaltılması için bir dizi görüntüden hesaplanan anlık bilgilerden faydalanır. Bu anlık veri genellikle sıralı görüntülerde değişen bölgeyi vurgulayan görüntü farkı şeklindedir. İmgede nesne bölgeleri belirtilirse nesne izleyici bir görüntüden sonrakine takip işlemini yürütmek için nesne uyumunu uygulama aşamasına geçer.

Tablo 2.1 Nesne bulma kategorileri

Kategori	Örnek Çalışma
Nokta bulucu	Moravec bulucusu Harris bulucu Ölçekten bağımsız öznitelik dönüşümü Affine bağımsız nokta bulucu
Bölütleme	Ortalama kayma Yöntemi Grafik kesim Aktif kontur
Arka plan modelleme	Gaussian Karışımı Öz arka plan Duvar çiçeği Dinamik dokulu arka plan
Denetimli sınıflandırıcı	Destek vektör makinaları Yapay sinir ağları Uyarlanabilir destekleme

Tablo 2.1’de yaygın kullanılan nesne bulma algoritmaları verilmiştir. Nesne bulma üzerinde ayrıca çalışma yapılması gereken konulardan biridir.

2.3.1. Nokta Bulucular

Nokta bulucular imgelerdeki sıralı bölgelerde anlamlı bir dokuya sahip öznitelik/özellik çıkarımının bulunması için kullanılır. Öznitelik çıkarım hareket, stereo ve takip problemleri bağlamında uzun süre kullanılmıştır. Öznitelik çıkarımın arzu edilen niteliği kamera görüş açısı ve ortam ışığındaki değişimlere karşı değişmezliktir.

Öznitelik çıkarımları bulmak için Moravec bulucu yatay, dikey, diagonal ve antidiagonal yönlerde imge parlaklıklarının değişimini hesaplar ve pencere için temsili değerler olarak dört adet değişken içerisinde minimum olanını seçer.

Harris bulucu yönlü yoğunluk değişkenlerini göz önüne almak için x ve y doğrultularında birinci dereceden imge türevlerini hesaplar. Daha sonra, bu değişkeni şifreleyen ikinci bir moment matrisi Denklem (2.1)’deki gibi küçük bir komşulukta her bir piksel için hesaplanır.

$$M = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \quad (2.1)$$

Harris ve Kanade Lucas Tomasi (KLT) benzer ölçümleri kullanarak yoğunluk değişiklikleri üzerinde durmaktadır. Pratik olarak bu iki yöntem ile hemen hemen aynı ilgili noktalar bulunur. Aradaki tek fark bulunan ilgili noktalar arasında önceden tanımlı bir uzaysal uzaklığı ifade eden ek bir KLT ölçütüdür.

Teorik olarak, M matrisi dönme ve dönüşüme karşı değişmez. Ancak, M matrisi affine ya da projektif dönüşümlerden bağımsız değildir. Farklı dönüşümler altında ilgili noktaların güçlü bir şekilde bulunmasını göstermek için Lowe çalışmasında dört adımdan oluşan ölçekten bağımsız özellik dönüşümü (scale invariant feature transform-SIFT) yöntemini önermiştir [33].

İlk olarak bir ölçek uzayı, farklı ölçeklerde Gaussian filtreleri ile imgenin konvüle edilmesiyle oluşturulur. Konvüle edilen imgeler farklı Gaussian imgeleri oluşturmak için kullanılır. Aday öznitelik çıkarımları, ölçekler karşısında farklı gaussian imgelerinin minimum ve maksimum değerlerinden seçilir. Bir sonraki adım, komşu piksellerin kullanılarak renk değerlerinin interpolate edilmesiyle her bir adayın yerini güncellemektir.

Üçüncü adımda, hem düşük kontrastlı adaylar hem de kenar kısımlardaki adaylar elimine edilir. Son olarak, geriye kalan öznitelik çıkarımları bir aday çıkarım çevresindeki küçük bir komşulukta gradyen yönlerinin histogramındaki uç noktalara dayalı oryantasyonlara atanır. SIFT bulucu diğer bulucularla karşılaştırıldığında çok sayıda öznitelik çıkarımı oluşturur. Bunun nedeni farklı ölçeklerde ve farklı çözünürlüklerde öznitelik çıkarımlarının biriktirilmesidir. Deneysel olarak Mikolajczyk tarafından yapılan çalışmada SIFT algoritmasının daha iyi öznitelik çıkarımında bulunduğunu ve imgede oluşan bozulmalara karşı daha dirençli olduğu gösterilmiştir [34].

2.3.2. Arka Plan Çıkarma

Nesne bulma alınan her bir görüntü modelinden sapmaların bulunması ve arka plan modeli diye isimlendirilen sahne gösterimleri oluşturularak elde edilebilir. Arka plan modelinden imge bölgesinde önemli bir değişiklik olması halinde hareketli bir nesnenin olduğu anlaşılır. Değişikliğe maruz kalan bölgeleri oluşturan pikseller daha ileri işlemler için belirtilir. Genellikle nesnelere ait birleşik bölgeleri elde etmek için birleşik bileşen algoritması uygulanır. Bu işlem arka plan çıkarma olarak isimlendirilir.

Arka plan çıkarma Wren ve Azarbayejani tarafından yapılan çalışma ile popüler olmuştur. Bu çalışmada zamanla kademeli değişiklikleri öğrenmek için tek bir üç boyutlu Gaussian ile sabit arka planda her bir pikselin renginin modellenmesi önerilir. Ortalama ve kovaryans parametreleri bir dizi sıralı görüntüdeki renk incelenmesinden öğrenilir. Giriş görüntüsünde arka plan modeli her bir piksel için elde edildiğinde renk benzerliği hesaplanır ve arka plan modelinden ayrılan pikseller ön plan pikselleri olarak sınıflandırılır. Ancak tek bir gauss dış ortamlar için uygun bir model olmamaktadır. Bunun sebebi çoklu renkler tekrarlı nesne hareketi, gölgeler ya da yansılardan dolayı tek bir konumda gözlenebilir. Arka plan modellemede önemli bir gelişme piksel başına arka plan rengini tanımlamak için çok biçimli istatistiksel modeller kullanılarak elde edilir [35]. Stauffer ve Grimson tarafından yapılan bir çalışmada gauss karışımı kullanarak piksel rengi modellenir [36]. Bu yöntemde bir gauss karşılığı bulunana kadar anlık görüntünün modelindeki her bir gauss ile karşılaştırarak arka plan modeline karşı anlık görüntüdeki her bir piksel kontrol edilir. Eğer eşleşme olursa, ortalama değer ve eşlenen gaussun değişimi güncellenir. Aksi halde ortalama değer eşiti yeni bir gauss o anki piksel rengine ve

başlangıç değeri bileşime tanıtılır. Her bir piksel uyumlu dağılımın arka plan işlemini ifade etmesine bağlı olarak sınıflandırılır.

Bir başka yaklaşım sadece renk tabanlı bilgi kullanılması yerine bölge tabanlı sahne bilgisini birleştirmektir. Elgammal ve Harwood tarafından yapılan bir çalışmada piksel başına arka plan modellemek için parametrik olmayan çekirdek yoğunluk tahmini kullanılmıştır [37]. Çıkartma işlemi süresince o anki piksel sadece arka plandaki ilgili piksel ile değil aynı zamanda yakın komşuluktaki pikseller ile de karşılaştırılır. Bu nedenle bu yöntem kamera titreşimleri ve arka plandaki önemsiz hareketlerin üstesinden gelebilir.

Arka plan çıkarma için alternatif bir başka yaklaşım ortamdaki olaylar ile ilgili katı durumlar olarak bir görüntü dizisinde bir pikselin yoğunluk değişimlerini göstermektir. Örneğin otobandaki araçların takibi için imge pikselleri arka plan, ön plan ya da gölge durumlarından seçilebilir. Hidden Markov Modeli (HMM) ile bu üç durumdan birine aitmiş gibi imgenin küçük blokları sınıflandırılır [38]. Stenger ve Ramesh tarafından yapılan bir çalışmada bir odadaki ışığın yanması ve kapanması durumları için HMM arka plan çıkarma için kullanılmıştır [39]. HMM kullanımının avantajı denetimsiz arka plan modelleme yaklaşımları kullanılarak doğru bir şekilde modellenmesi güç olan olayların eğitim örnekleri kullanılarak öğrenilebilmesidir.

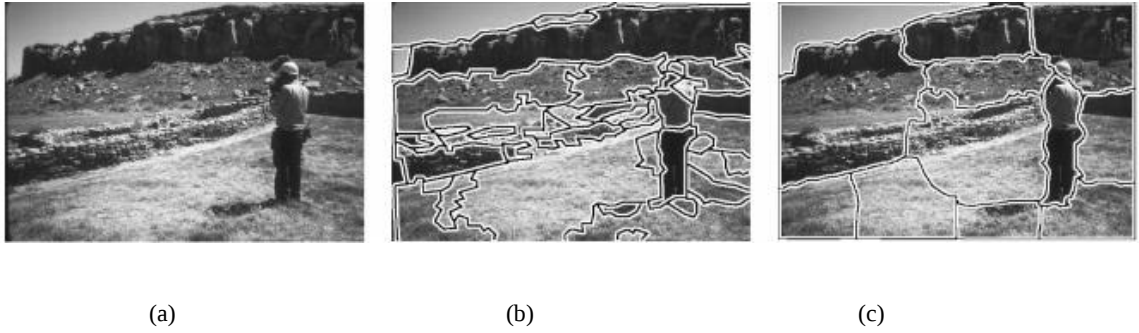
Söz konusu yaklaşımlardaki kısıtlama sabit bir arka plan gereksinimidir. Bu kısıtlama Monnet ve Mittal ile Zhong ve Sclaroff tarafından yapılan çalışmalarda ele alınmıştır. Her iki çalışmada da zamanla değişen arka plana çözüm sunmuşlardır. Bu yöntemler bir sahnedeki hareket örüntülerini tahmin etme ve öğrenme yolu olan özbağımlı hareketli ortalama işlemler olarak imge bölgesini modellerler. Özbağımlı hareketli ortalama işlemi, özbağımlı işlemin önceki değerlerinin ve beyaz gürültü hatasının ağırlıklı toplamı olarak tanımlandığı hareketli ortalama bileşenler ve özbağımlı toplamlarından oluşan bir zaman serisi modelidir [40, 41].

Özet olarak, sabit kameralar için çoğu takip yöntemleri arka plan çıkarma tekniklerini öznitelik çıkarım için kullanırlar[42,43]. Bunun sebebi son zamanlarda kullanılan arka plan çıkarma yöntemleri değişen ortam ışığı, gürültü ve arka planın periyodik olarak hareketinin modellenmesi gibi yeteneklere sahip olmasıdır. Böylece arka plan çıkarma yöntemleri ile değişik şartlar altında nesneler tespit edilebilmektedir. Ayrıca bu yöntemler hesaplama bakımından etkindir. Pratik olarak arka plan çıkarma çoğu durumda eksik nesne bölgeleri oluşturmaktadır. Bunun sebebi nesne içerisinde boşluklar olabilmesi ve böylece arka plan çıkarma işlemi yapılırken nesne özelliklerinin arka

plandan tam olarak ayrışmama sorunu oluşturabilmektedir. Arka plan çıkarma yönteminde en önemli kısıtlama sabit kamera gereksinimidir. Bunun sebebi kamera hareketinin genellikle arka plan modelini bozmasıdır.

2.3.3. Bölütleme

Nesne bölütleme algoritmalarının amacı imgenin benzer bölgelerinin algıya dayalı olarak bölümlenmesidir. Her bölütleme algoritması iyi bir bölümlenme için ölçüt ve etkili bölümlenme gerçekleştirme yöntemi olmak üzere iki problemi ele alır[44].



Şekil 2.6. (a) orijinal imge (b) ortalama kayma bölütlemesi uygulanmış imge (c) normalize edilmiş kesimler yöntemi uygulanmış imge

2.3.3.1. Ortalama Kayma Kümelemesi

İmge bölütleme problemi için Comaniciu ve Meer tarafından yapılan çalışmada ortak bir uzay ve renk uzayında kümelenmelerin bulunması için ortalama kayma yaklaşımı önerilmiştir. Bir imgedeki verilerden varsayılan sayıda rasgele seçilen küme merkezleri ile algoritma başlatılır. Daha sonra her bir küme merkezi küme merkezinde merkezlenen çok yönlü elips içerisinde bulunan veri ortalama kaymasına taşınır. Eski ve yeni küme merkezleri ile tanımlanan vektör ortalama değer kayma vektörü olarak isimlendirilir. Ortalama değer kayma vektörü küme merkezleri konumlarını değiştirmeyene kadar tekrarlı bir şekilde hesaplanır. Ortalama kayma iterasyonları boyunca bazı kümeler birleştirilebilir. Ortalama değer kayma kümelenmesi imge düzenleme, kenar tespiti ve takip uygulamalarına kadar birçok farklı uygulamada kullanılmıştır. Ortalama kayma tabanlı

bölütlemeye daha iyi bir bölütlemenin yapılması için değişik parametrelerin ayarlanması gerekir. Örnek olarak renk ve uzaysal çekirdek bant genişliği seçimi verilebilir. Şekil 2.6 (a)'daki imgeye ortalama kayma yöntemiyle uygulanan bölütleme sonucunda Şekil 2.6 (b)'deki imge elde edilmiştir [45].

2.3.3.2. Graf Kesme Kullanarak İmge Bölütleme

İmge bölütleme graf parçalama problemi olarak sembolize edebilir. İki alt grafik arasında kısaltılan kenarların toplam ağırlığı kesim olarak isimlendirilir. Ağırlık genellikle düğümler arasında renk, parlaklık ya da doku ile hesaplanır.

Shi ve Malik tarafından yapılan bir çalışmada aşırı bölütleme problemi ile başa çıkmak için normalleştirilmiş kesim önerilmiştir. Bu yaklaşımda kesim sadece kesimdeki kenar ağırlıklarının toplamına bağlı değil aynı zamanda grafın tüm düğümlerine ait her bir parçasında düğümlerin toplam bağlantı ağırlıklarının oranına bağlıdır. İmge tabanlı bölütleme için düğümler arasında ağırlıklar uzaysal yaklaşım ve renk benzerliğinin ürünü tarafından tanımlanır. Düğümlerin her bir parçası arasında ağırlıklar hesaplandığında, bir ağırlık matrisi ve diagonal matris oluşturulur. Bölütleme genelleştirilmiş öz sistemin özdeğerleri ve özvektörleri hesaplanarak uygulanır. Daha sonra ikinci en küçük öz vektör imgeyi iki parçaya bölmek için kullanılır. Her bir parça için bu işlem tekrarlı bir şekilde bir eşik değerine ulaşana dek uygulanır. Şekil 2.6 (c)'da normalize edilmiş kesimler yaklaşımıyla elde edilen bölütleme sonucu gösterilmiştir [44].

Normalize edilmiş kesimler tabanlı bölütlemeye büyük imgeler için genelleştirilmiş öz sistemlere çözüm bellek ve işlem gereksinimi açısından maliyetli olabilir. Diğer yandan bu yöntem ortalama kayma yöntemi ile bölütleme ile karşılaştırıldığında daha az manuel olarak seçilen parametrelere gereksinim duyar. Normalize edilmiş kesimler nesne konturları takibi bağlamında kullanılmıştır.

2.3.3.3. Aktif Konturlar

Aktif bir kontur çerçevesinde nesne bölütleme kapalı bir sınırı nesnenin sınırına uygulayarak oluşturulur. Konturun gelişimi varsayılan nesne bölgesine konturun uygunluğunu tanımlayan bir enerji fonksiyonu ile idare edilir.

Kontur tabanlı yöntemlerde önemli bir konu kontur başlatmasıdır. İmge gradyen tabanlı yaklaşımlarda bir kontur genel olarak nesne bölgesi dışına yerleştirilir ve nesne sınırı bulunana kadar daraltılır. Bu kısıtlama kontur nesne sınırına uyması için genişletsin ya da daraltsın diye nesne dışında ya da içinde kontur başlatılabilmesi için bölge tabanlı yöntemler rahatlatılır.

Ancak bu yaklaşımlar önsel nesne ya da arka plan bilgisine ihtiyaç duyar. Referans bir görüntü ya da çoklu görüntüler kullanılarak başlatma bölge öncelikleri oluşturulmadan uygulanabilir.

Enerji fonksiyonu ve ilk yönelim dışında bir başka önemli konu doğru kontur gösterimi seçimidir. Nesne konturu ya dolaylı olarak ya da doğrudan gösterilebilir. Açık bir gösterimde kontrol noktaları arasındaki ilişki eğri denklemleri ile tanımlanır. Kontur gelişimi enerji fonksiyonuna göre değişen grid değerleri ile yönetilir. Grid değerlerindeki değişiklikler yeni sıfır kesişmelerinde dolayısıyla yeni kontur pozisyonlarında sonuçlanır. Dolaylı gösterimin doğrudan gösterim üzerindeki en önemli avantajı topoloji değişikliklerindeki esnekliğidir.

2.3.4. Denetimli Öğrenme

Nesne bulma denetimli öğrenme mekanizması vasıtasıyla bir dizi örnekten alınan farklı nesne görünüşleri otomatik olarak öğrenme ile uygulanabilir. Farklı nesne görünüşlerinin öğrenilmesi bir dizi şablon kaydetmenin gereksinimini ortadan kaldırır. Bir dizi öğrenme örneğinin olması halinde denetimli öğrenme yöntemleri girişleri arzu edilen çıkışlarla eşleştiren bir fonksiyon oluştururlar. Denetimli öğrenmenin standart bir formülasyonu sınıflandırma problemidir. Bu sınıflandırma problemi öğrencinin ya regresyon diye adlandırılan ya da sınıflandırma diye adlandırılan bir şekilde çıkış oluşturularak bir fonksiyonun davranışını yaklaştırır. Nesne bulma bağlamında öğrenme örnekleri nesne özelliklerinin parçalarından ve nesne sınıfına ait çiftlerden oluşur. Bu iki nicelik manuel olarak belirtilir.

Özelliklerin seçimi sınıflandırma performansında önemli bir rol oynamaktadır. Bu nedenle bir sınıfı başka bir sınıftan ayırt edebilen bir dizi özellik kullanmak önemlidir. Nesne alanı, nesne oryantasyonu ve nesne görünümü bir yoğunluk fonksiyonu şeklinde kullanılabilir. Özelliklerin seçilmesiyle nesnenin farklı görünüşlerinden denetimli öğrenme yaklaşımı seçilerek öğrenilebilir. Bu öğrenme yaklaşımları yapay sinir ağları,

uyarlanabilir destekleme, karar ağaçları ve destek vektör makineleri ile yapılabilmektedir. Ve bu öğrenme yöntemleri ile yüksek boyuttaki uzayda bir nesneyi bir başka nesneden ayıran bir hiper yüzey hesaplanır.

Denetimli öğrenme yöntemleri genellikle her bir nesne sınıfından çok sayıda örnek gerektirmektedir. Ek olarak bu örnekler manuel olarak etiketlenmelidir. Manuel etiketlenmiş veri miktarı azaltmaya yönelik olası bir yaklaşım denetimli öğrenme ile birlikte eğitilmesidir. Tümlleşik eğitimin temel fikir özelliklerin her bir sınıflandırıcının bağımsız olarak kullanıldığı az etiketlenmiş küçük veri dizisi kullanılarak iki sınıflandırıcının eğitilmesidir. Eğitim aşaması sonrasında her bir sınıflandırıcı bir başka sınıflandırıcının eğitim dizisine etiketlenmiş verileri atamak için kullanılır. Uyarlanabilir destekleme ve destek vektör makinelerinin nesne takibine uygulanabilir olması nedeniyle kısaca bahsedilecektir.

2.3.4.1. Uyarlanabilir Destekleme

Destekleme, her biri kısmen doğru olan çoğu temel sınıflandırıcının birleştirilerek tekrarlı bir şekilde doğru bir sınıflandırıcının bulunması yöntemidir. Adaboost algoritmasının eğitim aşamasında ilk adım eğitim dizisi üzerinden ağırlıkların başlangıç dağılımını oluşturmaktır. Destekleme mekanizması hatanın yanlış sınıflandırılan verinin ağırlıklarına oranı olduğu en az hatayı veren temel bir sınıflandırıcı seçer. Daha sonra seçilen temel sınıflandırıcı tarafından yanlış sınıflandırılan veriye ait ağırlıklar arttırılır. Bu nedenle Adaboost algoritması bir sonraki iterasyonda yanlış sınıflandırılan verilerde daha iyi bir performans gösteren bir başka sınıflandırıcının seçimini desteklemektedir.

Nesne bulma durumunda zayıf sınıflandırıcılar resimden çıkartılan nesne özelliklerine uygulanan bir dizi eşik gibi basit operatörler olabilir. Viola ve Jones tarafından yapılan çalışmada Adaboost çerçevesi yayaları tespit etmek için kullanılmıştır. Bu yaklaşımda uzaysal ve geçici operatörlerin birleşimi ile çıkartılan imge özellikleri üzerinde eğitilen zayıf sınıflandırıcılar olarak yapay sinir ağırları seçilmiştir. Geçici alandaki işlemciler hareket bilgisinin bazı şekillerini kodlayan görüntü farkı şeklindedir. Görüntü farkı geçici alanda işlemci olarak kullanıldığı zaman hareketin olduğu yerdeki bölgelerde nesne tespitinin uygulanarak yanlış tespitlerin sayısını azaltır [46].

2.3.4.2. Destek Vektör Makineleri

Bir sınıflandırıcı olarak destek vektör makineleri bir sınıfı başka bir sınıftan ayıran maksimum marjinal hiperdüzlemi bularak veriyi iki sınıf halinde kümeler. Maksimuma çıkartılan hiperdüzlem sınırı en yakın veri noktaları ve hiperdüzlem arasındaki uzaklık ile tanımlanır. Hiperdüzlemin sınırlarında bulunan veri noktaları destek vektörleri olarak isimlendirilir. Nesne bulma bağlamında bu sınıflar nesne ve nesne olmayan sınıflara karşılık gelmektedir.

Lineer bir sınıflandırıcı olmasına karşın destek vektör makinesi lineer olmayan sınıflandırıcılar içinde çekirdeği girişten alınan giriş özellik vektörüne uygulayarak kullanılabilir. Lineer olarak ayrılabilen bir dizi verinin çekirdek uygulaması veriyi ayrılabilen yüksek yönlü uzaya dönüştürür. Kullanılan çekirdekler çok terimli çekirdekler ya da radyal tabanlı fonksiyonlardır. Ancak problem için doğru çekirdeğin seçimi manuel olarak kolay değildir. Çekirdek seçilince bir dizi parametre için sınıflandırma performansı test edilir.

Nesne bulmada Papageorgiou ve Oren çalışmalarında yayaların ve yüz seçimi için destek vektör makinelerini kullanmışlardır. Sınıfları birbirinden ayırt etmek için kullanılan özellikler pozitif ve negatif eğitim örnekleri dizisine Haar dalgacıkları uygulanarak çıkartılır. Arama alanının azaltılması için imgede optik akış alanı hesaplanarak geçici bilgi kullanılır. Özellikle optik akış alanındaki süreksizlikler azaltılmış yanlış pozitif eğitim örnekleri sayısı sonucunda olası nesnelerin aranması için kullanılır [47].

3. NESNE TAKİP YÖNTEMLERİ

Görüntü işlemede en popüler konulardan biri nesne takibidir. Literatürde nesne takibi işlevini yerine getirmek için birçok çözüm yöntemi sunulmuştur. Takip edilen nesnenin bulunduğu ortam şartları ve nesnenin farklı fiziksel özelliklere sahip olması sebebi ile önerilen yöntemleri farklı kategorilere ayırmak pek mümkün gözükmemektedir. Yine de belirli ölçütler göz önünde bulundurularak takip yöntemlerini belirli sınıflara ayırmak mümkün olabilmektedir. Nesnenin rengi, şekli ve hareketi gibi özellikler takip yöntemlerini belirleyen özellikler olarak öne çıkmaktadır.

Nesne takip yöntemleri iki esas işlevi gerçekleştirmekten ibarettir [48]. Bunlar takip edilecek nesnenin güncel imge içerisindeki konumunun tespit edilmesi ve imgeler boyunca konumları belirlenen nesneler arasındaki veri bağı ilişkisinin kurulması işlevleridir. Bu iki işlev birbirinden bağımsız olarak gerçekleştirilebileceği gibi birlikte de gerçekleştirilebilir. Takip işlemi bağımsız çalıştırıldığında, öncelikle nesne bulma algoritmaları yardımıyla ardışık imgelerdeki hareketli nesneler yakalanır. Daha sonra, bu nesneler arasında var olan veri bağı ilişkileri elde edilir. Takip işlemi birlikte çalıştırıldığında ise, güncel nesne bilgilerini elde edebilmek için güncel gözlem bilgisi ve önceki nesne bilgisi birlikte değerlendirilir.

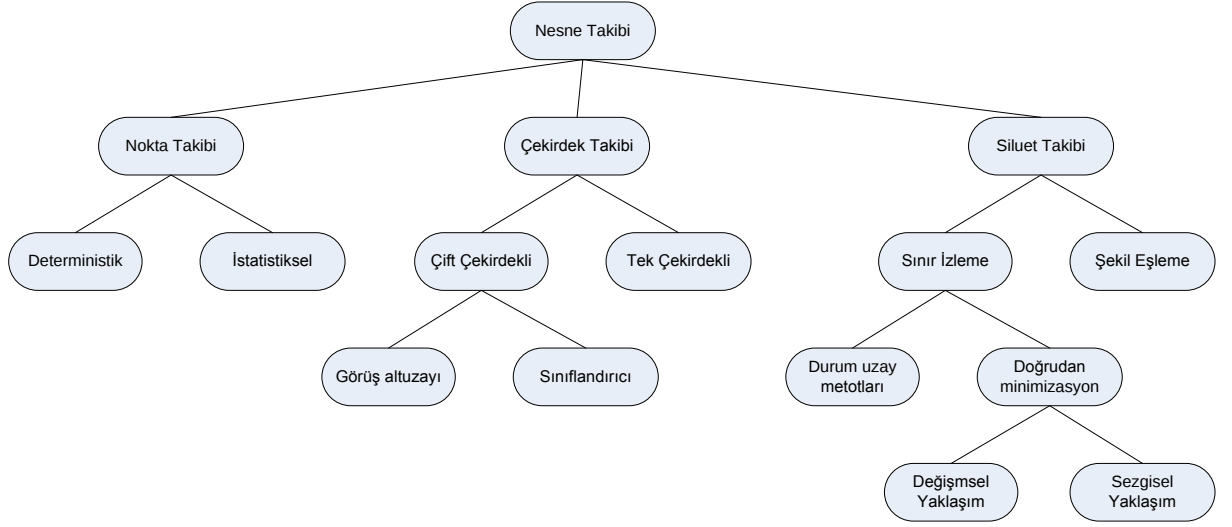
Her iki nesne takip işlevinin yerine getirilmesinde takip edilecek nesne şekil ve görünüm modelleri ile ifade edilir. Nesne şekli, nesnenin yapabileceği hareketi sınırlar. Örneğin; takip edilecek nesne sadece bir nokta ile ifade etmek için basit bir dönüştürme modeli yeterli olabilmektedir. Ancak, nesnenin eliptik bir geometrik şekle sahip olduğu bir durumda, parametrik hareket modelleri (affine veya projective dönüşüm modelleri) kullanılabilir. Bu sunum şekilleri yardımıyla geometrik şekli ve sınırları net olan nesnelerin hareketleri modellenir. Sınırları net olmayan nesnelere gelince siluet veya kenar bilgisi, daha tanımlayıcı bir sunum sağlayabilir. Ayrıca, bu tür nesnelerin hareketlerini modellemek için parametrik veya parametrik olmayan modeller kullanılabilmektedir[49].

Nesne hareketleri veya görünümleri üzerinde belirli kısıtlamalar yapılarak nesne takip problemi basitleşebilir. Neredeyse tüm nesne takip algoritmaları, nesne hareketlerinin ani değişimlere değil de yumuşak hareketlere sahip olduğunu varsaymaktadır. Ayrıca, nesne hareketlerinin sabit hız veya ivmeye sahip olduğu düşünülürse nesne takip işlemi oldukça basitleşmektedir. Buna ek olarak, takip edilen nesne sayısı, büyüklüğü, görünümü

veya sahip olduđu fiziksel řeklin nceden belirtilmesiyle de, nesne takip algoritmalarının karmaşıklığı azaltılmaktadır.

řekil 3.1’de nesne takip yöntemleri takip türüne göre sınıflandırılarak gösterilmiştir. Buna göre üç temel nesne takip yöntemi bulunmaktadır: 1) nokta tabanlı; 2) çekirdek tabanlı; 3) siluet tabanlı. Bu yöntemler ile ilgili literatürde yer alan çalışmalar Tablo 3.1’de listelendiğı gibi yöntemlerin kullanıldığı uygulama alanları da aşağıdaki gibi listelenebilir:

- Hareket tabanlı tanımlama (Örnek: Hareket doğrultusu bilinen bir tümörün akciğer üzerindeki yerinin tespiti [50], yürüyüş tabanlı insan tanıma).
- Otomatik Gözetim (Örnek: İstenmeyen olayları veya şüpheli hareketleri tespit etmek için belli bir alanın gözetlenmesi [42]).
- Video dizinleme (Örnek: Çoklu ortam veritabanında bulunan videoların, içerdığı nesneler ve bu nesnelerin hareketleri şeklinde kodlanarak sisteme kaydedilmesi, nesne adları veya hareketleri kullanılarak otomatik adlandırma yardımıyla video kayıtlarına hızlı bir şekilde ulaşabilmesi).
- İnsan bilgisayar etkileşimi (Örnek: İnsan vücut azaları takip edilerek insanın yürüdüğü veya koştuğı sonucunun üretilmesi [51], mimik tanıma, bilgisayarlara veri girişı için göz bakışı takibi).
- Trafik İzleme (Örnek: Trafik akışına ait istatistiksel verilerin gerçek zamanlı olarak toplanması).
- Araç yolculuğı (Video tabanlı güzergah planlama ve engelden kurtulma yeteneğı).



Şekil 3.1. Nesne takibi yöntemlerinin sınıflandırılması [1]

Tablo 3.1. Nesne takip yöntemleri ile ilgili literatürde bulunan örnek çalışmalar

Kategoriler	Örnek Uygulama yapılan Çalışmalar
Nokta tabanlı nesne takibi	
Deterministik Metotlar	[23] ve [52]
İstatistiksel Metotlar	[51], [53], [54], [55], [56], [57] ve [58]
Çekirdek tabanlı nesne takibi	
Şablon ve yoğunluk tabanlı görünüm modelleri	[45], [59] ve [60]
Çoklu-görüş görünüm modelleri	[61]
Siluet tabanlı nesne takibi	
Sınır Değerlendirme	[62] ve [63]
Şekil Eşleştirme	[64], [65] ve [66]

3.1. Nokta Tabanlı Nesne Takibi

Bu takip yönteminde takip edilen her bir nesne tek bir nokta ile ifade edilir. Bu yöntem ile güncel imgede takip edilecek her bir nesneye bir nokta aktarıldıktan sonra bu noktalar ile önceki imgede tespit edilen noktalar arasındaki veri bağı ilişkisinin doğru bir şekilde oluşturulması beklenir. Bu problemin çözümü için şu iki adım sırayla gerçekleştirilir:

- 1) Güncel imgede nesne yakalama (her biri nokta ile ifade edilir)
- 2) Önceki imgede bir nokta ile ifade edilen yakalanan nesne ile güncel imgede tespit edilen noktalar arasındaki nokta benzerlik değerlerinin hesaplanması.

Nokta benzerliğinin hesaplanması işlemi, özellikle nesne görünümünün kaybolması, yanlış nesne yakalanması, nesnenin imgeye ilk girişi veya çıkışı gibi durumlarda karmaşık bir duruma yol açar. Yılmaz ve Javed tarafından yapılan çalışmada bu alandaki yöntemler deterministik ve istatistiksel olmak üzere iki temel yöntem ile açıklanmıştır [1].

3.1.1. Deterministik Yöntemler

Deterministik yöntemler, ardışık imgelerde bulunan nesnelerin birbirlerine eşleştirilme maliyetlerini tanımlar. Eşleştirme maliyetinin en aza indirgenmesi tümleşik bir optimizasyon problemi olarak ifade edilir. Tüm muhtemel ilişkilendirmeler arasında bire bir eşleştirmenin olduğu bir çözüm olarak optimal atama metotları ile elde edilebilir. Nesneler arasındaki eşleştirme maliyetinin tanımlanması için nesne hareketleri üzerinde aşağıda belirtilen sınırlamalar göz önüne alınır [49]:

- Yakınlık: Bir imgeden diğer bir imgeye geçerken nesne pozisyonlarının önemli ölçüde değişmeyeceğini ima eder.
- Maksimum hız: Nesnelerin hızları üzerinde bir üst sınır değeri tanımlanır ve sadece nesnelerin etrafında dairesel bir komşuluk içerisinde kalan muhtemel nesne adaylarının benzerlik değerleri göz önüne alınır.
- Küçük hız değişimi (yumuşak hareket): Nesnenin hız yönünün önemli bir ölçüde değişmeyeceğini ima eder.

- Genel hareket: Küçük bir komşuluktaki nesnelerin hızlarının benzer olması için sınırlama uygulanır. Bu sınırlama çoklu noktalar ile sunulan nesneler için uygundur.
- Katılık: 3-boyutlu dünyadaki nesneler genellikle katı bir biçime sahiptir. Bundan dolayı güncel bir nesne üzerindeki herhangi iki nokta arasındaki mesafenin değişmeyeceği düşünülür. Bahsedilen sınırlamalar sadece deterministik yöntemlerde değil aynı zamanda istatistiksel yöntemlerde de kullanılabilir.

3.1.2. İstatistiksel Yöntemler

Video algılayıcılarından alınan ölçümler her zaman gürültüye sahiptir. Ayrıca, nesne hareketleri birtakım istenmeyen etkilere maruz kalır. İstatistiksel yöntemler, nesnenin durum (pozisyon, hız ve ivme) tahmini boyunca ölçüm değerlerini ve model belirsizliklerini göz önünde bulundurarak nesne takibi problemini çözmeye çalışır. Bu yöntemler pozisyon, hız ve ivme gibi nesne özelliklerini modellemek için durum uzay yaklaşımını kullanırlar [67]. Ölçümler genellikle bir tespit mekanizması tarafından elde edilen imgelerdeki nesne pozisyonlarını içerir. Bu noktadan sonra istatistiksel yöntemlerin imge içerisindeki nesnelere ait durum tahminini nasıl formüleştirdiği ve hangi çözümleri sunduğu detaylı bir şekilde incelenir. Buna göre problemin tanımı için Kalman filtreleme yöntemi ve parçacık filtreleme yöntemi kullanılır [37].

İmge üzerinde hareket eden bir nesnenin istatistiksel olarak durum tahmin problemi şu şekilde formüle edilebilir [53]. Takip edilecek nesnenin durum bilgisi (örneğin pozisyon) $X^t: t = 1, 2, \dots$ şeklinde bir dizi olarak tanımlanır. Zaman boyunca durum üzerindeki dinamik değişim Denklem (3.1) ile ifade edilir.

$$X^t = f^t(X^{t-1}) + W^t \quad (3.1)$$

$W^t: t = 1, 2, \dots$ gauss gürültüdür. Ölçüm verisi ile durum değişkeni arasındaki ilişki ise Denklem (3.2) ile ifade edilir.

$$Z^t = h^t(X^t) + N^t \quad (3.2)$$

N^t gauss gürültüdür ve W^t 'den bağımsızdır. İstatistiksel yöntemlere dayalı nesne takip edicilerin temel amacı, t anına kadarki tüm ölçüm değerlerini göz önünde bulundurarak X^t durum değişkenini tahmin etmektir.

Bir başka deyişle, posterior olasılık yoğunluk fonksiyonunu $p(X^t | Z^{1,...,t})$ elde etmektir. Teorik olarak en uygun çözüm, problemi iki adımda çözen tekrarlamalı Bayes filtresi yöntemini kullanmaktır. Bu adımlar tahmin ve düzeltme adımlarıdır. Tahmin adımı, dinamik bir eşitlik kullanır ve güncel durumun $t-1$ anındaki önceki olasılık yoğunluk fonksiyonunu $p(X^t | Z^{1,...,t-1})$ hesaplanır. Daha sonra sonraki yoğunluk fonksiyonunun hesaplanabilmesi için güncel ölçümün maksimum olabilirlik fonksiyonunu $p(Z^t | X^t)$ kullanılır. Buna göre sonraki yoğunluk fonksiyonu Denklem (3.3)'te gösterildiği gibi hesaplanır.

$$p(X^t | Z^{1,...,t}) = \frac{p(X^t | Z^{1,...,t-1})p(Z^t | X^t)}{p(Z^t | Z^{1,...,t-1})} \quad (3.3)$$

Denklem (3.3)'teki $p(Z^t | Z^{1,...,t-1})$ normalizasyon katsayısıdır. Ekranda sadece tek nesnenin olması durumunda, nesnenin durum bilgisi bahsedilen iki adım kullanılarak rahatlıkla tahmin edilebilir. Diğer taraftan, ekranda birden fazla nesnenin olması durumunda, elde edilen ölçümler ile ilgili nesneler arasında gerekli bağlantıların kurulma ihtiyacı ortaya çıkar. Bu nedenle çoklu nesnelerin takibi problemi için veri bağı ve durum tahmini problemlerinin birleştirilmiş bir çözümüne ihtiyaç duyulur.

Tek nesnenin olduğu bir durumda, eğer f^t ve h^t fonksiyonları doğrusal ve nesnenin başlangıç durumu X^1 ve gürültü değeri Gaussian dağılımına sahipse, o zaman en uygun çözüm Kalman filtre tarafından elde edilebilir. Eğer nesnenin başlangıç durumu ve sistem gürültüsü Gaussian dağılımına sahip değilse o zaman en uygun çözüm Parçacık filtreleme yöntemiyle elde edilir.

3.1.2.1. Kalman Filtresi ile Nesne Takibi

Kalman Filtresi, durum vektörü bir Gaussian dağılımına sahip lineer sistemlerin durum tahminlerini gerçekleştirmek için kullanılır [58]. Tahmin ve düzeltme gibi iki adımdan oluşur. Tahmin adımı, değişkenlerin yeni durumu tahminini yapabilmek için durum modelini kullanır.

$$\bar{X}^t = DX^{t-1} + W \quad (3.4)$$

$$\bar{\Sigma}^t = D\Sigma^{t-1}D^T + Q^t \quad (3.5)$$

Denklem (3.4) ve (3.5) sırayla $\bar{X}^t$ ve $\bar{\Sigma}^t$, t anındaki durum ve kovaryans tahminleridir. D, t ile t-1 anındaki durum değişkenleri arasındaki ilişkiyi tanımlayan durum dönüşüm matrisidir. Q, W gürültüsünün kovaryansıdır. Benzer şekilde, düzeltme adımı güncel gözlem değerini Z^t nesne durumunu güncellemek için kullanır.

$$K^t = \bar{\Sigma}^t M^t [M^t \bar{\Sigma}^t M^t + R^t]^{-1} \quad (3.6)$$

$$X^t = \bar{X}^t + K^t [Z^t - M\bar{X}^t] \quad (3.7)$$

$$\Sigma^t = \bar{\Sigma}^t - K^t M \bar{\Sigma}^t \quad (3.8)$$

$$v = Z^t - M\bar{\Sigma}^t \quad (3.9)$$

Denklem (3.9)'da v yenilik olarak adlandırılır ve Denklem (3.6), (3.7) ve (3.8)'de verilen M ölçme matrisidir. K , durum modellerinin yayılımı için kullanılan kalman kazancıdır. X^t durumu güncellendikten sonra Gaussian dağılımı olmaya devam eder. Bu durumda f^t ve h^t fonksiyonları doğrusal değildir ve taylor seri açılımları kullanılarak fonksiyonlar lineerleştirilebilirler. Bahsedilen bu filtreleme tekniği literatürde genişletilmiş kalman filtresi olarak kullanılmaktadır.

Kalman filtresi bilgisayarlı görme uygulamalarında takip işlemi için yaygın bir şekilde kullanılmıştır. Kalman filtresi gürültülü imgelerde noktaları takip etmek için kullanılmıştır. Stereo kamera tabanlı nesne takibi çalışmasında, kalman filtresi nesnenin pozisyonunu ve hızını x-z boyutlarında tahmin etmek için kullanılmıştır. Genişletilmiş kalman filtresi iki boyuttaki hareketten nesnenin üç boyuttaki yörüngesini belirlemek için kullanılmıştır [1].

3.1.2.2. Parçacık Filtresi ile Nesne Takibi

Kalman filtresinin bir dezavantajı, durum değişkenlerinin Gaussian dağılımına sahip olma zorunluluğudur. Böylelikle, kalman filtresi gaussian dağılıma sahip olmayan sistemler için zayıf bir tahmin edicidir [9]. Bu kısıtlama parçacık filtreleme yöntemiyle

çözülebilir [55]. Parçacık filtresinde, t anındaki şartlı durum yoğunluğu π ağırlığına sahip N tane parçacık içeren örnek küme ile ifade edilir. Her bir parçacık $\{s_t^{(n)}: n = 1, \dots, N\}$ ve her bir parçacığın ağırlığı ise $n_t^{(n)}$ (örnekleme olasılığı) olarak gösterilir. Ağırlıklar parçacığın önemi veya onun gözlenme frekansı olarak tanımlanabilir. Her bir $(s_t^{(n)}, \pi_t^{(n)})$ 'nin hesaplama maliyetinin azaltılması için katlanmış bir ağırlık $c^{(n)}$ de aynı zamanda hafızaya yüklenir ($c^{(N)} = 1$). t anındaki yeni örnekler $t-1$ anındaki $S_{t-1} = \{s_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}: n = 1, \dots, N\}$ örnekleme şemasıyla elde edilir. En genel örnekleme şeması aşağıda bahsedildiği gibi önem örnekleme şemasıdır. Bu şema, seçme, tahmin ve düzeltme olmak üzere üç temel adımın tekrarlı bir şekilde gerçekleşmesi prensibine dayanır [67].

(1) Seçme adımı: S_{t-1} 'den N tane rastgele $\hat{s}_t^{(n)}$ örnek seçilir. Seçme işlemi yapılırken $r \in [0,1]$ olmak kaydıyla rastgele bir değişken üretilir ve $c_{t-1}^{(j)} > r$ ve $\hat{s}_t^{(n)} = s_{t-1}^{(j)}$ şartlarını sağlayan en küçük j indeksine sahip örnekler seçilir.

(2) Tahmin adımı: Seçilen her bir $\hat{s}_t^{(n)} = f(\hat{s}_t^{(n)}, W_t^{(n)})$ ile yeni bir örnek üretilir. Burada $W_t^{(n)}$ sıfır ortalamaya sahip bir Gaussian hatasıdır ve f negatif olmayan bir fonksiyondur ($f(s)=s$).

(3) Düzeltme adımı: $s_t^{(n)}$ örneklerine ait $\pi_t^{(n)}$ ağırlıkları z_t ölçümleri kullanılarak hesaplanır. Bunun için $\pi_t^{(n)} = p(z_t | x_t = s_t^{(n)})$ eşitliği kullanılır. Burada $p(\cdot)$ gaussian dağılımı ile modellenebilir.

Elde edilen yeni S_t örnekleri $\epsilon_t = \sum_{n=1}^N \pi_t^{(n)} f(s_t^{(n)}, W)$ eşitliğinde kullanılarak yeni nesne pozisyonları hesaplanabilir. Parçacık filtre tabanlı nesne takip edicilerin başlangıç değerleri, örnekleme dizisini kullanan sistemlerin eğitimiyle veya ilk ölçümler ($s_0^{(n)} \sim X_0$) kullanılarak gerçekleşir. Sistem ilk ölçümler kullanılarak başlatılırsa, her bir örneğin ağırlık değeri $\pi_0^{(n)} = 1/N$ şeklinde eşit olarak dağıtılır. Ayrıca en iyi parçacık örneklerinin korunması için düşük ağırlıklı olanların elenmesi gerekir. Bunun için ek bir örnekleme algoritmasına ihtiyaç duyulur. Burada dikkat edilmesi gereken nokta, sonraki yoğunluk fonksiyonunun Gaussian olmak zorunda olmayışıdır.

3.2. Çekirdek Tabanlı Nesne Takibi

Çekirdek takibi görüntü dizileri boyunca ilkel bir nesne bölgesi ile gösterilen nesnenin hareketi hesaplanarak uygulanır. Nesne hareketi genel olarak parametrik bir yapıdadır ya da sıralı görüntülerde hesaplanan yoğunluk akış alanıdır. Bu algoritmalar kullanılan görünüm gösterimi, takip edilen nesne sayısı ve nesne hareketini tahmin etmek için kullanılan metot açısından farklılık gösterir. Bu takip yöntemleri şablonlar, yoğunluk tabanlı görünüm modelleri ve çok yönlü görünüm modelleri olmak üzere üç farklı alt kategoride incelenir.

3.2.1. Şablon ve Yoğunluk Tabanlı Görünüm Modelleri Kullanılarak Nesne Takibi

Şablonlar ve yoğunluk tabanlı görünüm modelleri basit olmaları ve düşük hesaplama maliyetlerinden dolayı yaygın olarak kullanılmıştır. Bu kategorideki nesne izleyicileri nesnelerin ayrı ayrı ya da birlikte takip edilmesine bağlı olarak iki alt kategoride incelenir.

3.2.2. Tek bir Nesnenin Takibi

Bu kategorideki en yaygın yaklaşım şablon eşlemedir. Şablon eşleme görüntüde arama yapmak için kaba bir yöntem olarak karşımıza çıkmaktadır. İmge yoğunluğu ve renk özellikleri şablonları oluşturmak için kullanılır. İmge yoğunluğunun ortam ışığından etkilenmesi sebebi ile imge gradyanı da özellik olarak kullanılabilir. Şablon eşlemenin dezavantajı hedef nesnenin imgede kabaca aranması nedeniyle yüksek hesaplama maliyetinin olmasıdır. Hesaplama maliyetinin düşürülmesi için, nesne arama işlemi nesnenin bir önceki pozisyonu çevresinde yapılır. Buna ilaveten şablon eşleme için daha etkili algoritmalar önerilmiştir [68].

Renk histogramı, dikdörtgen ya da eliptik bölgeler içerisinde kalan piksellerin görünümü kullanılarak hesaplanabilen bileşik modeller gibi diğer nesne gösterimleri şablon yerine kullanılabilir. Dikdörtgen nesne bölgesi içerisindeki piksellerin ortalama renk değerini bularak nesne modelleri oluşturulur. Hesaplamayla ilgili karmaşıklığı azaltmak için nesne sekiz komşu bölgede aranır. Nesne modeli ile tahmin edilen nesne pozisyonu arasındaki benzerlik bu ikisinin renk ortalaması oranı ölçülerek hesaplanır [69].

Comaniciu ve Ramesh çalışmalarında nesneyi göstermek için dairesel bir bölgede hesaplanan ağırlıklı histogram kullanmıştır. Nesnenin tam olarak yerinin saptanması için kaba bir yöntemle arama yerine ortalama değer kayması yaklaşımını kullanmışlardır. Bu yöntemle oluşturulan nesne izleyici tekrarlı bir şekilde nesnenin histogramı ile tahmin edilen nesnenin bulunduğu yerin penceresini karşılaştırarak görünüm benzerliğini mümkün olan en yüksek seviyeye çıkarır. Histogram benzerliği Bhattacharya katsayısı ile ifade edilir [60]. Bhattacharya katsayısı;

$$\sum_{u=1}^b P(u)Q(u)$$

Q: Nesnenin histogramı

P: Tahmin edilen nesne pozisyonu

Her bir iterasyonda ortalama kayma vektörü histogram benzerliği arttırılması için hesaplanır. Bu işlem yakınsama gerçekleşene kadar tekrarlanır. Genelde beş ya da altı iterasyonda gerçekleşir. Ortalama kayması algoritmasının şablon eşlemeden avantajlı olması işlem yoğunluğunun az olması, hesaplama maliyetinin düşük olması ve daha az iterasyon gerektirmesidir. Ancak ortalama kayma algoritmasında dikkat edilmesi gereken nokta başlangıçta nesnenin bir kısmının dairesel bölge içerisinde kalmasıdır.

Jepson ve Fleet tarafından yapılan çalışmada hedef nesnenin takip edilmesi için sabit görünüm özellikleri, geçici özellikler ve gürültü işleminden oluşan üç bileşenin kullanılmasını önermiştir. Sabit görünüm bileşeni hareket tahmini için en güvenilir görünüm olan zamanla görünümü değişmeyen nesnenin bölgelerini tanımlar. Geçici bileşen devamlı değişen pikselleri tanımlar. Gürültü bileşeni ise nesne görünümünde gürültüden kaynaklanan aykırı değerleri ele alır. Beklenti maksimizasyonu algoritmasının çevirim içi versiyonu bu üç bileşenin parametrelerini öğrenmek için kullanılmıştır. Görünüm gösterimleri için özellik olarak yönlendirilebilir filtre tepkileri aşaması kullanılmıştır. Nesne şekli elips ile gösterilmiştir. Nesne hareketi görüntü dizileri boyunca takip edilen bölgenin eğitimi açısından hesaplanır. Eğitim dönüşümü öteleme (t_x, t_y) , dönme (a, b) ve ölçek (s) parametrelerinden oluşur [69].

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = s \begin{pmatrix} a & b \\ -b & a \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (3.10)$$

Sabit ve geçici bileşenlerinin ağırlıklı birleşimi eğitim parametrelerini belirlemek için kullanılır. Sabit ve geçici özelliklerin avantajı takip için sabit özelliklere ağırlık

verilebilmesidir. Örneğin, konuşan birinin yüz takibi yapılmak istenirse alın ve burun bölgelerinin seçilmesi bireyin ağız bölgesine nazaran daha iyi bir eşleştirme olmaktadır [1].

Görüntüde belli bir bölgenin takibi için bir başka yaklaşım optik akış yöntemi kullanılarak ilgili bölgenin dönüşümünün hesaplanması için ilkel bir şekil tanımlanır. Optik akış yöntemleri parlaklığın değişmediği kısıtlamaların olduğu şartlarda her bir pikselin akış vektörü hesaplanarak yoğunluk akış alanları oluşturmak için kullanılır.

$$I(x, y, t) - I(x + dx, y + dy, t + dt) = 0 \quad (3.11)$$

Denklem (3.11)'de yapılan hesaplama piksel komşuluğunda ya geometrik olarak ya da cebirsel olarak gerçekleştirilir. Dikdörtgen bir bölgenin dönüşümünü hesaplamak için optik akış yöntemlerinin geliştirilmesi oldukça önemlidir. KLT izleyici bir noktada toplanan bir bölgenin dönüşümünü tekrarlı bir şekilde hesaplar.

$$\begin{pmatrix} \Sigma I_x^2 & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y^2 \end{pmatrix} * \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \Sigma I_x I_t \\ \Sigma I_y I_t \end{pmatrix} \quad (3.12)$$

Bu denklemde ilgili bölgenin yeni yeri elde edildiğinde, KLT izleyici affine dönüşümünü hesaplayarak takip edilen parçanın kalitesini hesaplamaktadır.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = s \begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (3.13)$$

Eğer anlık yama ve hedeflenen yama arasındaki karesel farkın toplamı küçük ise, özellik takip edilmeye devam edilir. Aksi halde özellik elimine edilir [1].

3.2.3. Çoklu Görünüm Modelleri Kullanılarak Nesne Takibi

Şuana kadar bahsedilen nesne takibi yöntemlerinde histogram, şablon gibi görünüm modelleri genellikle çevirim içi oluşturulur. Bu modeller nesne hakkında en son yapılan gözlemlerden elde edilen bilgiyi temsil eder. Nesne farklı açılardan farklı görünebilir. Eğer nesne görünümü takip boyunca önemli ölçüde değişirse, nesnenin görünüm modeli geçerliliğini yitirebilir ve takip edilen nesne kaybedilebilir. Bu problemin üstesinden

gelmek için, nesnenin farklı açılardan görüntüsü çevirim dışı olarak alınıp takip için kullanılabilir.

Caballero ve Merino tarafından yapılan çalışmada alt uzay tabanlı olan vektör uzayı yaklaşımı önerilmiştir. Bu yaklaşım ile öz vektörler kullanılarak oluşturulan imgeye o anki nesne imgesinden affine dönüşümü hesaplanır. İlk olarak bir nesnenin görünümünün alt uzay gösterimi temel bileşen analizi kullanılarak oluşturulur. Daha sonra imgeden öz uzaya dönüşüm hesaplanır. Bu hesaplama işlemi giriş imgesi ile öz vektörler kullanılarak oluşturulan imge arasındaki farkı hesaplayan alt uzay sabit denklemi en aza indirgenerek yapılır. En aza indirgeme işlemi alt uzay katsayıları ve affine parametrelerini hesaplama olmak üzere iki adımda yapılır. İlk adımda, affine parametreleri sabittir ve alt uzay katsayıları hesaplanır. İkinci adımda, yeni alt uzay katsayıları kullanılarak affine parametreleri hesaplanır. Buna bağlı olarak, takip giriş imgesi ile hedeflenen imge arasındaki fark en aza indirgenene dek affine parametrelerini tekrarlı bir şekilde hesaplayarak uygulanır. Benzerlik hesaplaması için öz uzay kullanımı ve normalleştirilmiş korelasyon gibi standart şablon eşleme tekniklerine faydalı bir alternatiftir. Öz uzay tabanlı benzerlik hesaplaması öz şablonların lineer bileşimi ile karşılaştırmaya eşittir. Bu sonuç imgelerdeki aydınlanma değişimlerinden kaynaklanan bozulma gibi şablonlarda da bozulmaya izin verir.

Avidan tarafından yapılan çalışmada takip için destek vektör makine sınıflandırıcı kullanılmıştır. Destek vektör makinaları (DVM) verilen pozitif ve negatif eğitim örnekleri içerisinde en iyi farklı hiperdüzlemi bulan genel bir sınıflandırma yöntemidir. Test aşaması boyunca DVM pozitif sınıfa test verisinin üyelik derecesini belirten test verisini kazandırır. DVM tabanlı izleyiciler için, pozitif örnekler takip edilmesi gereken nesnenin imgelerinden, negatif örnekler takip edilmemesi gereken diğer tüm nesnelerden oluşur. Genel olarak negatif örnekler takip edilen nesneden ayırt edilemeyen arka plandan oluşur. Bu çalışmadaki yöntem, imge bölgelerinden şablonun parlaklık farklılığını en aza indirmek yerine, nesnenin pozisyonunu tahmin etmek için imge bölgelerindeki DVM sınıflandırma puanını maksimuma çıkarır. Bu yöntemin bir avantajı arka plandaki nesneler ile ilgili bilgi açık bir şekilde izleyici ile birleştirilir [70].

Özetleyecek olursak, Nesne izleyicilerin çekirdek tabanlı uygulamalardaki amacı nesnenin hareketini tahmin etmektir. Bölge tabanlı nesne gösterimi, hesaplanan hareket tamamen nesne bölgesine ve bir sonraki görüntüdeki nesne oryantasyonunu tanımlar. Anlık görüntüde nesnenin her bir noktası için nesnenin yeri bir sonraki görüntüde tahmin

edilen hareket modeli kullanılarak belirlenebilir. Bu izleyicilerden hangisinin kullanıldığına bağlı olarak bu özelliklerden sadece biri önem arz eder. Örneğin, nesne yörüngesine bağlı olarak nesne davranışı analizi durumunda nesne hareketi yeterlidir. Ancak nesnenin tanımlanması için nesnenin kapsamış olduğu alan önemlidir. Bu kategorideki nesne izleyicilerin performansını değerlendirmek için, nesne izleyiciden beklentiye bağlı olarak ölçümler tanımlanır. Nesne izleyiciden sadece hareketinin ölçülebildiği durumda, değerlendirme tahmin edilen ve gerçek hareket parametreleri arasındaki uzaklık ölçümü hesaplanarak yapılabilir. Çekirdek tabanlı nesne izleyiciler aşağıdaki durumlar göz önüne alınarak nitelikli bir karşılaştırma yapılabilir.

- Tek bir nesnenin ya da birden fazla nesnenin takip edilmesi
- Takip edilen nesne ile kamera arasına bir başka nesnenin girdiği durumlarla başa çıkabilme yeteneği
- Eğitim gereksinimi
- Hareket modeli tipi
- Manuel olarak nesne takibi işleminin başlatılması

Tablo 3.2’de bu bölümde bahsedilen yöntemlerin nitelikli bir karşılaştırılması verilmiştir.

Tablo3.2. Geometrik model tabanlı nesne izleyicilerin karşılaştırılması #:nesne sayısı M:çoklu nesne S: Tek bir nesne A: Affine ya da homografi T:Dönüşüm hareketi S: Ölçekleme R: Dönme P:Kısmi oklüzyon F: Tam oklüzyon $\checkmark$ ve x sırasıyla nesne izleyicinin eğitim ya da başlatma işlemi gerektirip gerektirmediğini gösterir [1].

	#	Hareket	Eğitim	Oklüzyon	Başlatma
Basit Şablon eşleme	S	T	x	P	$\checkmark$
Ortalama kayma yöntemi	S	T+S	x	P	$\checkmark$
KLT	S	A	x	P	$\checkmark$
Görünüm takibi	S	T+S+R	x	P	$\checkmark$
Katmanlama	M	T+S+R	x	F	x
Bramble	M	T+S+R	$\checkmark$	F	x
Özizleyici	S	A	$\checkmark$	P	$\checkmark$
DVM	S	T	$\checkmark$	P	$\checkmark$

Nesneleri göstermek için kullanılan ilkel geometrik şekiller modern yöntemlerin gerçek zamanlı olarak uygulanmasından dolayı sıklıkla rastlanan bir durumdur. Rijit kısıtlamalardan dolayı bu kategorideki nesne takibi yöntemleri nesnenin parametrik hareketini hesaplar. Bu hareket genellikle dönüşüm, uyumlu affine, affine ya da tahmin şeklindedir. Nesne hareketi önceki ve o anki görüntü arasındaki nesne görünüm benzerliği maksimuma çıkartarak tahmin edilebilir. Tahmin işlemi kaba bir arama şeklinde ya da gradyen değişimi tabanlı uygunlaştırma kullanılarak olabilir. Gradyen değişimi yaklaşımı tabanlı nesne izleyiciler, bir önceki nesne konumundan pozisyonu belirlenen seçilen nesne içerisinde nesnenin bir kısmının görülebilir olmasını gerektirir. Bu tür gereksinimlerin azaltılması için olası bir yaklaşım olarak nesnenin bir sonraki görüntüdeki yerini tahmin edebilen kalman filtresi ya da parçacık filtresi kullanımıdır. Nesne merkezinin ivme ya da hız açısından tanımlanması durumunda bu filtreler nesne merkezinin konumunu tahmin ederler. Bu gereksinim global hareket gereksinimi uygulanması ile nesnelerin kameradan ve kamera hareketinden uzak olduğu varsayılarak karşılanabilir.

Nesne gösterimi için ilkel geometrik şekillerin kısıtlamalarından biri de nesnenin bir kısmı tanımlanan şeklin dışında kalıyor iken arka planın bir kısmının tanımlanan şekil içerisine bulunmasıdır. Bu fenomen hem rijit hem de rijit olmayan nesneler için gözlenebilir. Bu tür durumlarda maksimuma çıkartılmış model benzerliği nesne hareketi için doğru olmayabilir. Bu kısıtlamayı aşmak için, çekirdeğin tüm şekli tamamen içermesinden ziyade nesneyi bulundurmaya zorlamaktır. Bir başka yaklaşım nesne görünümünün renk ya da doku olasılık yoğunluk fonksiyonları ile modellenmesi ve ölçülen renk ya da dokunun şartlı olasılık tabanlı ilkel şekil içerisinde bulunan piksellere ağırlık atanmasıdır.

3.3. Siluet Takibi

Takip edilmesi istenilen nesneler basit geometrik şekiller ile tanımlanamayan el, baş, omuz gibi karmaşık geometrik şekillere sahip olabilir. Siluet tabanlı yöntemler bu nesneler için doğru bir şekil tanımlayıcısı sağlarlar. Siluet tabanlı nesne takip edicilerinin asıl hedefi, önceki imgeler kullanılarak üretilen nesne modelini güncel imge içerisinde bulmaktır [71]. Bu modeller nesnenin renk histogramı, nesne kenarını veya sınır şeklini kullanır. Siluet tabanlı nesne takip edicileri “şekil karşılaştırmacılar” ve “sınır takip ediciler” olarak adlandırılan iki alt kategori altında incelemek mümkündür. Şekil

karşılaştırmalı yaklaşım, güncel imge içerisinde nesne silüetini arar. Diğer taraftan sınır izlemeli yaklaşım ise, bazı enerji fonksiyonlarının en aza indirgenmesi veya durum uzay modellerini kullanarak nesnenin güncel imgedeki yeni yerini belirlemeye çalışır.

3.3.1. Şekil Karşılaştırmacılar

Şekil karşılaştırmacı yöntemlerin çalışma prensibi şablon karşılaştırmalı nesne takip yöntemleri ile benzerdir. Bu yaklaşımda, nesne silüetinin güncel imgeden bir sonraki imgeye aktarıldığı varsayılır. Güncel imgedeki nesne silüeti ve onunla ilişkili nesne modeli bir sonraki imgede aranır. Arama işlemi, bir önceki imgede elde edilen nesne silüeti ve modeli ile güncel imgede yakalanan nesneler arasındaki benzerlik değerleri hesaplanarak gerçekleştirilir. Bu nedenle, silüeti net olmayan (katı olmayan) nesneler doğru ve istikrarlı bir şekilde takip edilemez. Genellikle nesne modeli için kenar haritası kullanılır. Nesne modelinin her imgede nesneyi tam ifade edebilmesi için görünüş değişikliklerini göz önüne alarak yeniden hesaplanması gerekir. Bu güncelleme işlemi, özellikle katı olmayan nesne hareketlerindeki ışık değişimi ve görüş noktası gibi nesne takip problemlerinin üstesinden gelebilmek için gereklidir [65].

3.3.2. Sınır Takibi

Şekil karşılaştırma yöntemlerinin aksine sınır takibi yöntemleri güncel imgedeki sınır çizgilerini başlangıç kabul ederek bir sonraki imgede yeni nesne sınırlarını arar. Sınır takibi algoritmalarının uygun sonuç vermesi için güncel imgedeki nesne ile bir sonraki imgede bulunan nesne alanlarının kesişimi olmak zorundadır. Sınır değerlendirme algoritmaları iki alt kategoride incelenebilir. İlk yaklaşım, sınır biçim ve hareketini modellemek için durum uzay modellerini kullanır. İkinci yaklaşım, gradient azalma gibi minimizasyon tekniklerini kullanarak sınır enerjisini minimize eder [64].

3.3.2.1. Durum Uzay Modeli Kullanarak Nesne Takibi

Nesneye ait durum vektörü, nesne sınırının biçim ve hareketi şeklinde tanımlanır. Nesne sınırına ait sonrasal olasılık yoğunluk değerinin maksimum olması için sınıra ait durum vektörü her bir imgede güncellenir. Sonrasal yoğunluk, öncesel yoğunluğa ve

güncel olabilirliğe bağlıdır. Güncel olabilirlik değeri, nesne sınırları ile gözlenen kenar değerleri arasındaki mesafe kullanılarak hesaplanabilir.

3.3.2.2. Sınır Enerji Fonksiyonunun Direkt Minimizasyonu Yöntemiyle Nesne Takibi

Sınır değerlendirme kontekstinde, bu bölümde işlenen sınır takip ediciler ile bir önceki bölümde incelenen bölütleme yöntemleri arasında bir paralellik bulunmaktadır. Nesne bölütleme ve nesne takip yöntemleri sınır enerji fonksiyonunu greedy veya gradient azaltma yöntemlerini kullanarak minimize eder. Sınır enerjisi iki farklı yöntem ile tanımlanabilir [63]. Bu yöntemler geçici gradient (optik akış) yöntemi ve nesne ile arka plan alanından üretilen görünüm istatistikleri yöntemidir. Geçici görüntü gradientini kullanarak sınır takip yöntemi, optik akış hesaplaması üzerine yapılan kapsamlı bir çalışmayla motive edilir. Optik akış sınırlamaları Denklem (3.14) ile belirtilen sabit parlaklık sınırlaması kullanılarak üretilir:

$$I^{t+1}(x + y) - I^t(x - u, y - v) = 0 \quad 3.14)$$

Denklem (4.14)'te I görüntü, t zaman, (u,v) ise x ve y yönündeki optik akış vektörleridir. Optik akış yöntemine alternatif olarak, bir imgeden diğerine geçerken nesnenin iç ve dış alanları hakkında hesaplanan istatistiksel tutarlılık değeri kullanılarak nesne takibi gerçekleştirilebilir. Bu yaklaşım sınırın güncel imgede başlangıç değerlerinin belirlenmesi gerekliliğine ihtiyaç duyar.

4. İNSANSIZ HAVA ARAÇLARI

Drone olarak da isimlendirilen insansız hava aracı; pilot, kabin ekibi veya yolcu bulundurmeyen hava araçları olarak bilinmektedir. İHA ile ilgili ilk çalışmalar 1. Dünya savaşı sonrasında askeri amaçlı alanlarda yapılmaya başladı. Günümüzde yapılan çalışmaları açısından MIT Aerospace Controls Lab, ETH Flying Machine Arena ve Pennsylvania General Robotics, Automation, Sensing and Perception (GRASP) üniversite laboratuvarları öne çıkmaktadır. Türkiye’de ise Vestel Savunma Sanayi A.Ş. ve Türk Havacılık ve Uzay Sanayii A.Ş. son yıllarda insansız hava aracı ile yapmış oldukları tasarım, geliştirme ve üretimlerle önemli bir seviyeye ulaşmıştır. Amerikan ordusu maliyeti şu anda oldukça yüksek olan Predator Drone ve Global Hawk adında tam otonomun uygulanamadığı iki adet İHA gözlem ve keşif amaçlı kullanmaktadır [72,73].

İHA, bir yer istasyonunda bulunan kontrolör ile iletişim kurabilmekte ve bu kontrolöre sıcaklık ve görüşle ilgili imge verisi, bulunduğu yerin konum bilgileri, yüksekliği ve hızıyla ilgili bilgileri aktarabilmektedir. İstenmeyen bir durumda veya herhangi bir arıza anında doğru hareketi tercih edebilme ve operatörü uyarma yeteneğine sahiptir. Birçok görevi yerine getirmek üzere farklı alanlarda kullanılmıştır. İHA’lar doğal felaketlerde yukarıdan çekim yaparak felaketin ne boyutta olduğunu gözlemlemek için kullanılmıştır. Tarım arazilerinde üstten çekim yaparak istenmeyen bir durumun olup olmadığını kısa bir sürede gözden geçirmek için kullanılmıştır. Ayrıca orman yangını kontrolü, yük taşıma ve film çekimi gibi sivil uygulamalarda da kullanılmıştır [74]. Askeri alanda İHA’lardan geleneksel kullanımın amaçlarından farklı olarak faydalanılmaktadır. Askeri alanda tehlike durumunda İHA kullanımı mürettebat olmaması sebebi ile herhangi bir can kaybına sebep olmamaktadır.

Düşük maliyetli İHA’ları askeri, sivil ve bilimsel alanlarda birçok uygulama için uygun olmaktadır. Düşük irtifada uçan insansız hava araçları bilimsel veri toplama, doğal kaynak izleme, belli bir alanın düzeninin sağlanması amacıyla gözetimi ve askeri keşif amacıyla kullanılabilirler.

Küçük İHA’lar ile ilgili yapılan çalışmalarda karşılaşılan problemler de bulunmaktadır. Küçük İHA’larda düşük güç tüketimine ihtiyaç duyma, insan ara yüzünün kolay bir şekilde uygulanması ve güçlü ve hafif mekanik yapıya sahip olması

istenilmektedir. Küçük İHA'lar otonom özelliği arttırılmış olarak yol bulma, yörünge oluşturma ve takip algoritmalarına ihtiyaç duyarlar.

4.1. Sabit Kanatlı ve Döner Kanatlı İHA Tipleri

İHA'lar sabit ve döner kanatlı olmak üzere iki farklı şekilde incelenebilir. Sabit kanatlı İHA'lar döner kanatlı tiplere göre daha uzak mesafelerde ve daha yüksek hızlarda uçabilme yeteneğine sahiptir. Ancak, sabit kanatlı İHA'lar iniş ve kalkış için piste ihtiyaç duyarlar. Diğer yandan döner kanatlı İHA, bir cismin yüzeyi üzerinde her noktada havanın etki ettiği dikey ve teğetsel kuvvetlerden oluşturmuş olduğu aerodinamik kuvvetin etkisi ile kendisini daha yüksek bir noktaya taşıyabilen, içerisinde mürettebat bulundurmayan, manevra kabiliyeti yüksek ve yük taşıyabilen hava aracıdır. Bu durum döner kanatlı İHA'ların menzilini ve hızını sınırlamaktadır. Ayrıca, döner kanatlı İHA'lar mekanik olarak oldukça karmaşık bir yapıya sahiptir [75]. Şekil 4.1'de farklı tip İHA'lar verilmiştir.



Şekil 4.1 Farklı tip İHA'lar

Döner kanatlı İHA'nın sabit bir pozisyonda durmasını sağlar. Böylece döner kanatlı İHA belirli bir istikamette gidebilir ve istenilen noktaya iniş yapabilir. İnişten sonra döner kanatlı İHA rotorlarını durdurup güç tüketimini azaltarak sensörler ile bilgi almaya devam edebilir. Bu işlem özellikle keşif amaçlı uygulamalarda kullanılır.

Döner kanatlı İHA diğer İHA tiplerine göre hafif olması ve manevra kabiliyetinin yüksek olması, rotor hızlarını değiştirerek 3 boyutlu hareketleri yapabilmeleri, yükselip-alçalmaları, iniş-kalkış yapabilmeleri ve havada asılı kalabilmeleri sebebiyle daha çok tercih edilmektedir. Bu avantajlarından ötürü döner kanatlı İHA'nın birçok modeli tasarlanmış, kontrol algoritmaları yapılmış ve birçok uygulamada yer almıştır. Gözetim sistemlerinde, navigasyon, film çekimi, gerçek zamanlı sistemler, robot teknolojisi ve çevresel haritalamaya kadar geniş bir uygulama alanında kullanılmaktadırlar. Genellikle dış ortamlarda otonom navigasyon için otopilot kullanılmaktadırlar. GPS iletişiminin sağlanmadığı ortam koşullarında bazı konumlandırma tekniklerinden oryantasyon ve pozisyon belirlemek amacıyla faydalanmaktadır [76, 77, 78].

4.2. İHA'ların Özellikleri

Nesne takibi ya da başka bir uygulamada kullanmak üzere İHA seçilirken belli özellikler göz önünde bulundurulmalıdır. Bu özellikler ağırlık, dayanıklılık, menzil, irtifa, kanat yükü ve motor tipi olarak sıralanabilir.

4.2.1. Ağırlık

İHA'lar ağırlık bakımından incelendiğinde 5 kg ağırlığa kadar olan mikro İHA'lardan 11 ton ağırlığa kadar olanları mevcuttur. Bunlara sırayla Dragon Eye ve Global Hawk örnek olarak verilebilir. Hafif olan İHA'lar elektrik motorları kullanırlar. Ağır olan İHA'lar ise jet motorlar kullanırlar [79].

4.2.2. Dayanıklılık ve Menzil

İHA'larda dayanıklılık ve menzil, bir haberleşme ağı içerisinde uçuş hızı, havada kalabildiği süre, çıkabileceği yükseklik ve kat edebildiği mesafe ile ilgilidir. Farklı görevleri yerine getirmek üzere farklı İHA'lar tasarlanmıştır. Bu görevler kalkış pistleri

çevresinde ya da uzak mesafelerde yürütülebilirler. İHA menzili ne kadar aralıklarla yakıt ikmali yapılması gerektiği ya da şarj edilmesi gerektiği ile belirlenir. Çoğu İHA yakıt ikmali yapılması ya da şarj edilmesi için yere iniş yapması gerekir. Global Hawk gibi büyük İHA'lar havada yakıt ikmali yapabilirler ancak bu durum zaman almaktadır. AR.Drone gibi düşük dayanıklılığa sahip İHA'lar 20 dakikaya kadar havada kalabilirler ve bu İHA'lar genellikle kısa mesafede yapılması amaçlanan uygulamalar için kullanılırlar. Yüksek dayanıklılığa sahip İHA'lar 24 saatten fazla havada kalabilir ve 22000 km irtifada uçabilirler [79]. Bu çalışmada kullanılacak İHA'nın gerçekleştireceği takip işlemini performansını görmek için 10 dakika gibi bir uçuş gerçekleştirebilmesi yeterli olmaktadır.

4.2.3. Uçuş Yüksekliği

İHA seçerken göz önünde bulundurulması gereken önemli özelliklerden biri uçabileceği maksimum yüksekliktir. Yüksek irtifada uçabilme özelliği İHA'nın düşman tarafından tespit edilip yok edilmesini zorlaştırmaktadır. Yüksek bir irtifada uçabilmesi görüntülenmesi istenen bölgenin alanını arttırmaktadır. Düşük irtifada uçan İHA'lar için maksimum yükseklik 100 metreye kadar değişmektedir. Örneğin, The General Atomics MQ-9 Reaper 40000 ft irtifaya çıkabilir. İHA'ların yüksek irtifada uçabilmeleri diğer hava araçları ile çarpışma gibi olumsuz durumları oluşturabilir. Bu durumun önüne geçebilmek için İHA'lar çarpışmadan kaçış sistemleri ile programlanabilirler [79].

4.2.4. Kanat Ağırlığı

Kanat ağırlığı hava aracının ağırlığının kanat bölgesine oranı demektir. Kanat ağırlığı hava aracının taşıdığı yük ve bu yükü taşıması için hangi hızda gitmesi gerektiğini belirlemektedir. Hava aracının hızı arttıkça kaldırabileceği yük artmaktadır. Buradan küçük hava araçlarının büyük hava araçlarına oranla daha yüksek hızda gitmesi durumunda aynı yükü taşıyabileceği söylenebilir. Bu durum ağır hava aracının kalkışı ve inişi daha yüksek hızda yaptığını ve düşük manevra kabiliyetine sahip olduğunu gösterir.

4.2.5. Motor Tipi

Hava aracının ağırlığı arttıkça kullanılan motorun gücü de aynı oranda yeterli kaldırma kuvvetinin sağlanması için artmalıdır. Küçük İHA’larda küçük elektrik motorları kullanılırken ağır olanlarda ise pistonlu motorlar (içten yanmalı motorlar) kullanılmaktadır. Ayrıca motor tipi İHA’nın dayanıklılığını ve menzilini de etkilemektedir.

4.3. İnsansız Hava Aracı’nın Seçilmesi

Bahsedilen özellikler dikkate alındığında bu tez çalışmasında hafif ağırlıklı bir insansız hava aracının seçilmesi gerekir. İnsansız hava aracının uygun dayanıklılığa ve menzile sahip olması ve de dar bir alanda çalışabilmesi bu çalışmanın gerçekleştirilmesi için yeterli olmaktadır. Yapılacak olan testlerin uygulanması için en az 10 dakika süreyle uçuş yapabilen döner kanatlı bir İHA’nın seçilmesi gerekir. Düşey olarak iniş-kalkış yapabilen bir sisteme sahip olmalıdır. Taşınabilen, kısa sürede şarj edilebilme ve uçuşun gerçekleştirilmesi için uygun motor tipine sahip bir döner kanatlı İHA olmalıdır. Parrot AR.Drone tüm bu gereksinimleri karşılayan dört rotorlu bir İHA olması sebebi ile bu çalışmada kullanılacaktır.

4.4. AR.Drone

Parrot AR.Drone döner kanatlı bir İHA olup ilk olarak eğlence amaçlı video oyunlarında kullanılmak üzere geliştirildi. 2010 yılında piyasaya sürüldü ve alışlagelmişin dışında gelişim gösterip askeri ve sivil uygulamalarda kullanılmaya başladı. Ayrıca hafif ağırlığı, düşük maliyeti ve manevra kabiliyetinden dolayı akademik amaçlı çalışmalar için de oldukça cazip bir hale geldi. Mobil cihazlarda çalıştırılabilen uygulamalar sayesinde kolay bir şekilde uçurulabilir. Şekil 4.2’de olduğu gibi kapalı ortam ve dış ortamda kullanmak için farklı gövde yapıları bulunmaktadır. AR.Drone’u donanımsal olarak oluşturan bileşenler Şekil 4.3’de verilmiştir.

Üzerinde Wi-Fi bağlantı ve yerleşik 2 kamera bulunmaktadır. Kameralardan ön tarafta bulunan geniş bir görüş açısına sahip olup diğeri ise düşey olarak kullanılan yüksek hızlı bir kameradır. AR.Drone Windows ve Linux tabanlı uygulamalarda kullanılabilir. Bu yazılımlar iniş, kalkış ve havada durabilmesi için bazı otomatik

yöntemler sağlamaktadır. Ayrıca umumi bir yazılım geliştirme kiti (software development kit-SDK) AR.Drone'un kontrolü için bazı özel amaçlı uygulamalar sunar. SDK, AR.Drone ile Adhoc Wi-Fi bağlantısı kurarak komut gönderme (in, kalk, aşağı/yukarı, sağ-sol, vb) alma, şifreleme ve iki kamera ile anlık video görüntüsü alma ve batarya doluluk durumunu gösterme gibi işlemlerin yapılabilmesi mümkün olmaktadır.



(a) İç mekan kullanım



(b) Dış mekan kullanım

Şekil 4.2.AR:Drone gövde yapısı

4.4.1. AR.Drone'un Mekanik Yapısı

AR.Drone'da ileri teknoloji tasarım ürünü olan karbon fiber kabin yapısına sahiptir. Ağırlığı iç ortam kabini ile 420 gr, dış ortam kabini ile 380 gr olması bakımından önemlidir. Ultrasonik sensörler, sıvı geçirmemesi için su geçirmez ince bir tabaka tarafından korunmuştur. AR.Drone'un titreşimlerden etkilenmemesi için donanımsal kısmı köpük ile izole edilmiştir.

Karbon fiber, fiberglas, plastik, kompozit, ağaç gibi çeşitli malzemelerden yapılabilen, değişik shaft sayısı ve boyutlarda tasarlanabilen pervaneler aracın havalanmasını sağlayan itki kuvvetinin oluşmasında büyük öneme sahiptir. Ahşap, fiberglas ve karbon fiber pervaneler en iyi performansı verirler. Plastik pervaneler ise ucuz olmalarının yanında esnektirler ve titreşime neden olarak güç kaybına neden olurlar. AR.Drone üzerinde dört adet pervane bulunmaktadır. Bunlardan ikisi (sağ-sol) "C"

pervaneleri, diğerk ikisi (ön-arka) ise “A” pervaneleridir. C saat yönünde dönecek olan pervaneler, A ise saat yönü tersine dönecek pervaneleri belirtmek amacıyla işaretlenmiştir.

AR.Drone’da kabin düşünülürken öncelikle hangi malzemeden yapıldığı göz önünde bulundurulmuştur. Kullanılan kabin hafif ve dayanıklıdır. Kabin üzerine etkiyen temel kuvvetler yerçekimi kuvveti ve hava basıncıdır. Karbon fiber bir kabin aslında döner kanatlı İHA’nın özelliklerine en iyi hitap edebilecek bir malzemedir. Çünkü karbon fiber gerilmelerin ve esnemelerin ahşap malzemeye göre daha rahat üstesinden gelir ve alüminyumdan da daha hafiftir.

4.4.2. Motor Tipi

AR.Drone her biri bir mikro kontrolör tarafından kontrol edilen dört adet fırçasız doğru akım motoru kullanır. AR.Drone bu dört rotorun çalışıp çalışmadığını veya durduğunu algılayabilmektedir. Bu sayede AR.Drone çalışırken pervanelerde herhangi olumsuz bir durumla karşılaşılması önlenir. Bu tür bir durumla karşılaşılması halinde AR.Drone tüm motorları durdurarak acil iniş yapmaktadır.

Motor yapısını incelediğimiz zaman, her bir pervane için 1 adet fırçasız doğru akım motoru kullanılmıştır. Her bir motorun nominal gücü 14,5 W ve nominal hızı ise 28500 devir/dk’dır. Adından da anlaşıldığı üzere fırçasız doğru akım motorlar komütasyon için fırça kullanmazlar. Fırçasız doğru akım motorlarının gün geçtikçe popülerliği artmaktadır [80].

Fırçasız doğru akım motorları fırçalı doğru akım motorlarına göre birçok üstünlüğe sahiptir. Bunlar:

- Daha iyi hız tork karakteristikleri
- Yüksek dinamik cevap
- Yüksek verimlilik
- Uzun ömür
- Sessiz çalışma
- Yüksek hız aralıklarında çalışabilme özellikleridir.

4.4.3. Elektronik Hız Kontrolcüsü

Fırçasız Doğru Akım Motorları (FDAM), çalıştırılması ve kontrolü oldukça zor motorlardır. Bu motorlar, yapıları gereği elektronik olarak kontrol edilir ve doğru akım komütasyonu için kontrol işleminde rotor pozisyon bilgisi gerektirirler. Elektronik hız kontrolörü, fırçasız motorları kontrol etmek için motorların bobinlerine sırasıyla darbe genişlik modülasyon işaretlerini yollarlar [81].

4.4.4. Lipo Bataryalar

AR.Drone 1000 mAh'lik ve 11.1 V'luk şarj edilebilen batarya ile çalışmaktadır. Batarya şarjı 12.5 V olduğunda tam şarj ve 9 V olduğunda düşük şarj uyarısı vermektedir. AR.Drone bataryanın gerilimini göstermekte ve değerini yüzde olarak göstermektedir. Böylece kullanıcı da AR.Drone'u ne kadar süre daha uçuracağını ve gidebileceği mesafeyi ona göre belirlemektedir. Batarya gerilimi kritik değere ulaşınca AR.Drone hasar görmemek için enerjisini kesip acil iniş yapmaktadır [82].

4.4.5. İvmeler, Jiroskoplar ve İşlemciler

AR.Drone'un donanımında yer alan jiroskoplar, tüm hava araçlarında olduğu gibi kontrol sistemleri için en önemli donanım parçası olup, aracın tam pozisyon ve açı bilgisini elde etmek için kullanıldığı sensörlerdir. AR.Drone sensörlerinin tümü AR.Drone'un merkez gövdesinin altına yerleştirilmiştir. AR.Drone altı serbestlik dereceli, mikro elektromekanik sistem tabanlı küçültülmüş eylemsiz ölçme birimine sahiptir [82]. Bu ölçme birimi yönelme, yuvarlanma ve yunuslama ölçümlü bir yazılım sağlamaktadır. Ve bu ölçme birimi üç eksenli ivmeölçer, iki eksenli yuvarlanma ve yunuslama jiroskopu ve tek eksenli jiroskoptan oluşmaktadır[83]. AR.Drone üzerinde iki kamera, iki işlemci ve sonar bulunduran bir ana kart bulundurmaktadır. Bu işlemcilerden biri giriş ve çıkıştan bilgi almak için kullanılır. Diğerisi ise AR.Drone'un ihtiyacı olan uçuş dengesini sağlamak ve görüntü işlemek amacıyla kullanılan algoritmaların çalıştırılması için kullanılır [84]. İvmeölçer olarak Bosch Sensortec firması tarafından üretilen BMA150 kullanılmıştır. MEMS ivmeölçer ayrı eksenlerde üç katmanı kullanır ve her eksen yönündeki ivmeyi ölçebilir. Yerçekimini ölçmek için ivmeölçerin bu yeteneği tilt sensörü olarak kullanılma

olanağı sağlamaktadır. Jiroskoplar eğimi x,y ve z eksenlerinde saniye başına derece cinsinden açısal hız olarak ölçer.

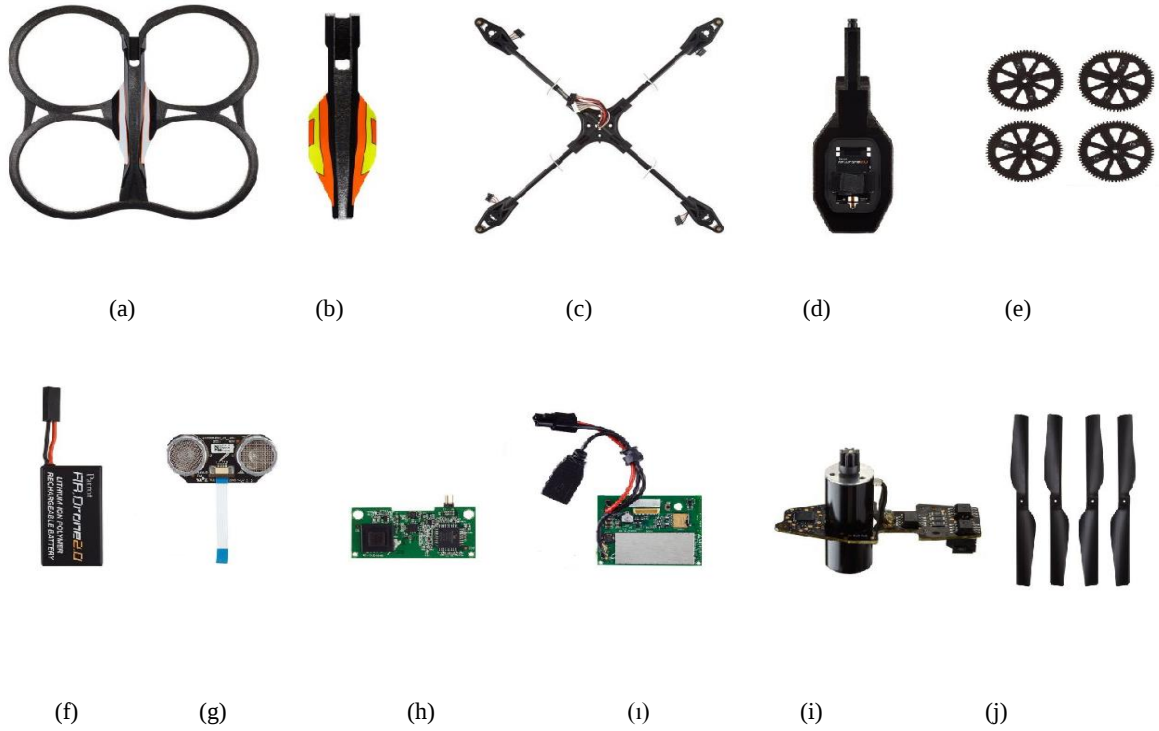
Aracın irtifadaki pozisyon bilgisini elde edebilmek için 3-eksenli ivmeölçer kullanılmıştır. İvmeölçerler İHA'nın her eksen yönündeki (x, y, z) olan hızlanmaları ölçmek amacıyla kullanılırlar. Lineer ivmelenmeleri ölçebilirler. İvmeölçerler atalet ölçüm sistemlerinde jiroskoplar ile birlikte kullanılırlar. Tek başlarına kullanıldıklarında yeterli olmazlar, çünkü ivmeölçerler açısal ivmelenmeleri ölçemezler.

Ayrıca, manyetik alan ölçer, ultrasonik mesafe sensörü ve basınç sensörü bulunmaktadır.

4.4.6. Anakart

AR.Drone ana kartı içerisinde gömülü farklı bileşenler bulundurulur. Bu bileşenler 32 bit 800 MHz'de çalışan ARM Cortex A8 işlemci, Wi-Fi modülü, düşey doğrultuda alta bakan bir kamera ve ön kamerayı çalıştıran bağlantı elemanıdır. İşlemci her iki kameradan da bilgi almakla görevlidir.

ARM Cortex A8 işlemci gerçek zamanlı uygulamalarda C/C++ gibi programlama dilleri ile uyumlu çalışabilmektedir. 200 MHz'de çalışan 1 Gbit DDR2 RAM bulunmaktadır.



Şekil 4.3. AR.Drone'un donanımsal bileşenleri: a) iç ortam kabini, b) dış ortam kabini, c) orta çapraz d) gövde, e) şaftlar, f) Li-on batarya, g) ultrasonik alıcı-verici, h) navigasyon kartı, ı) Anakart, i) motor+EHK devresi j) pervaneler [85]

4.4.7. Kameralar

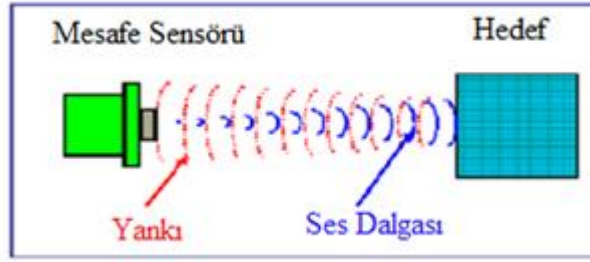
AR.Drone'da iki adet dahili CMOS kamera bulunmaktadır. Bu kameralardan biri alt diğeri ise ön kameradır. Ön kamera canlı olarak 30 fps görüntü almaktadır. Ön kamera 92 derece görüş açılı ve 720 p çözünürlüğe sahiptir. Ön kameranın 92° diagonal geniş açıya sahip olması, düşük gecikmeli veri akışının olması, USB üzerinden ya da Wi-Fi bağlantı ile uzaktan kontrollü cihaz ile veri kaydedebilmesi önemli özellikleridir. Ön kamera çok oyunculu oyunlarda diğer drone'ları bulmak için ve AR.Drone'un kontrol edilmesi için video görüntülerinin alınması için kullanılır. Alt kamera ise 320*240 piksel çözünürlüğe sahip olup 60 fps görüntü alan QVGA kameradır. Alt kameranın bu özelliği sayesinde hareket bulanıklıkları azaltılmakta ve böylece uçuş algoritmaları bu kamera ile iyileştirilmektedir. Her iki kamera da AR.Drone'un uçuş kabiliyeti için önemli rol oynamaktadır.

4.4.8. Ultrasonik Yükseklikölçer

Ultrasonik sensör AR.Drone için yükseklik ölçümlerini otomatik yükseklik dengesinin gerçekleşmesi için sağlamaktadır. Yükseklik ölçümü düşey hız kontrolünün ölçülmesini de desteklemektedir. Ultrasonik yükseklikölçer AR.Drone'un alt tarafına monte edilmiştir ve yer ile arasındaki yüksekliği ölçer. Sensör, göndermiş olduğu sinyalin tekrar yansımalarını bekler ve yankılanan sinyalin almış olduğu süre kullanılarak sensör ile nesne arasındaki uzaklık belirlenir. Aradaki mesafe ses hızı kullanılarak Denklem (4.1) ifadesi ile hesaplanır. Donanımsal olarak Şekil 4.4'teki yapıya benzer bir sistemi içerir.

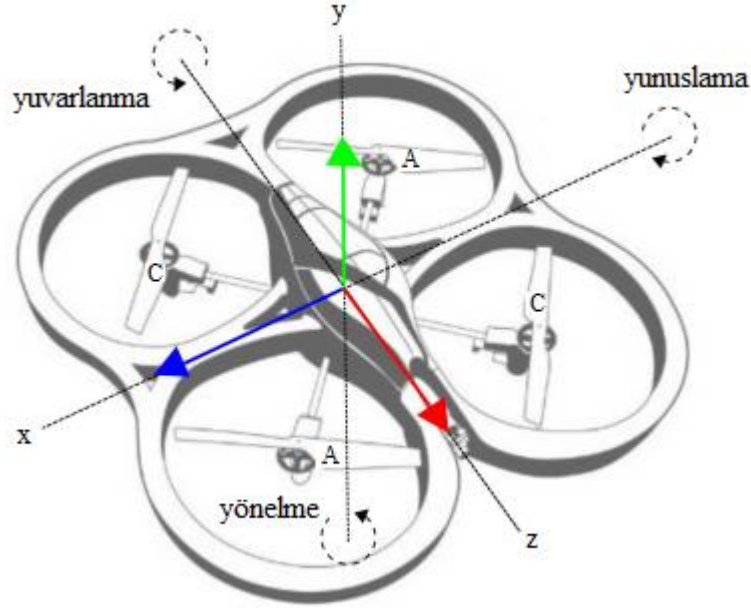
$$d = \frac{c \times t}{2} \quad (4.1)$$

Burada c; 343 m/s olup havadaki ses hızına ve t ise ses dalgasının hedeften yansıyıp sensöre ulaşma süresine karşılık gelmektedir



Şekil 4.4. Ultrasonik mesafe ölçerin donanımsal yapısı

4.5. AR.Drone'nun Hareket Kontrolü



Şekil 4.5.AR.Drone hareket doğrultuları

AR.Drone, dört adet pervanenin çapraz şekilde yerleşiminden oluşur. Bu çapraz şekildeki yapı, hafif olmasına rağmen sistem çatısından daha ağır olan motorları ve onların mekanik bağlantılarını kaldırarak kadar dayanıklı olmalıdır. AR.Drone Şekil 4.5'te olduğu gibi üç farklı eksende yönlendirilebilmektedir. Her pervane bir motora bağlantılıdır. Bütün pervane dönüş eksenleri sabit ve paraleldir. AR.Drone pervane yapısı gereği hava akışı sabit olup yukarı yönlü kaldırma kuvveti sağlamak için hava akışı aşağıya doğru olmaktadır. Bu yaklaşımlar, yapının oldukça rijid ve sadece pervanenin hız değişimine göre ayarlandığını göstermektedir.

AR.Drone'nun hareketi doğrudan pervanelerin hızlarıyla ilgilidir. Sağ ve sol pervaneler saat yönünde dönerken, ön ve arka pervaneler saat yönünün tersine dönüş yapar. Bu karşılıklı grupların zıt düzenleşimi, helikopterler için zorunlu olan kuyruk rotorunu ortadan kaldırır. AR.Drone'un, havada sabit kalabilmesi için de tüm pervanelerin aynı hızda olmaları gerekmektedir. Tüm pervaneler aynı hızda döndüğünde, yerçekimi etkisiyle oluşan ivmelenme dengelenmiş olur. Bu yüzden AR.Drone, sabit bir uçuş yapar ve etki eden kuvvetler ve torklar konumunu değiştiremez. Eğer tüm pervaneler aynı yönde döndürülecek olursa her bir rotorun oluşturduğu döndürme momenti birbirini

destekler biçimde olacaktır ve AR.Drone havada sabit kalmak yerine düşey merkez eksenini etrafında dönmeye başlayacaktır.

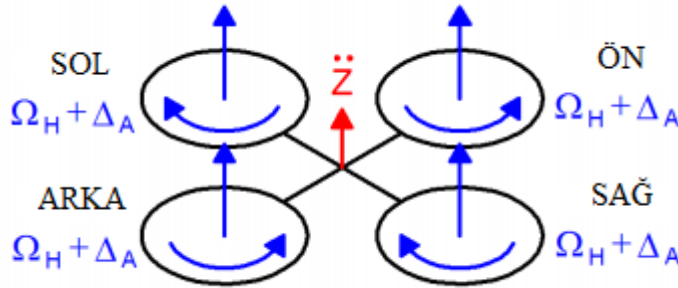
AR.Drone'nun hareket kontrolü yapılırken belirli bir yükseklik ve açısal pozisyona ulaşmasını sağlayan dört temel hareket göz önünde bulundurulmalıdır.

4.5.1. Yükselme Hareketi

Yükselme hareketi Şekil 4.6'dan görüldüğü gibi tüm pervanelerin hızlarını aynı miktarda artırıp azaltmak için kullanılır. Hava aracını yükseltecek veya alçaltacak şekilde, ana gövde çatısına dikey bir kuvvet uygular. Eğer araç yatay pozisyonda ise, eylemsizlik çerçevesinin dikey yönü ile sabit ana gövde çerçevesi çakışır. Aksi takdirde, elde edilen itki eylemsizlik çatısında hem dikey, hem yatay ivmelenme üretir. Pervane hızları;

$$\Omega_H \mp \Delta$$

ifadesi ile belirlenmektedir. Burada Ω_H motorun sabit hızını, Δ hız değişim miktarını ifade eder. Pervane hızlarının artması durumunda Δ , Δ_A olup, azalması durumunda ise Δ_B olmaktadır.

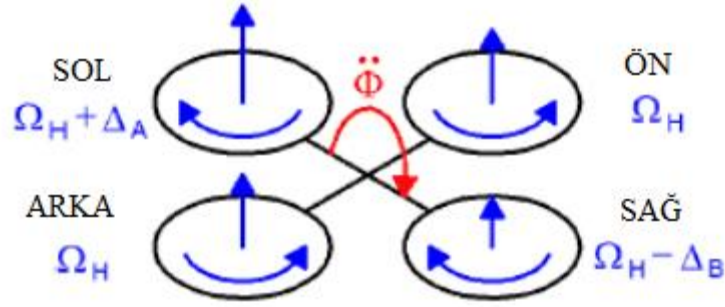


Şekil 4.6.Yükselme hareketi

4.5.2. Yuvarlanma Hareketi

Bu hareket ile sağ pervanenin hızı azaltılırken, sol pervanenin hızının artırılmasıyla elde edilir. Benzer şekilde sağ pervanenin hızı artırılırken, sol pervanenin hızının azaltılmasıyla elde edilir. AR.Drone'u x-ekseni boyunca döndürecek şekilde bir tork ile uygulanır. Tüm dikey itki kuvvetlerinin, havada sabit tutunma pozisyonunda olduğu gibi, eşit değerde iken bu hareket, Şekil 4.7'de olduğu gibi sadece yuvarlanma

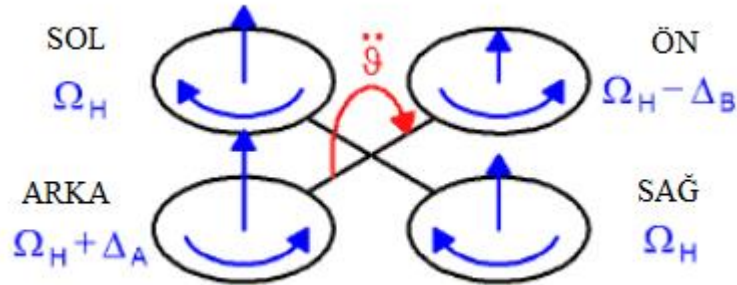
açısının ivmelenmesine neden olur. Hava aracı üzerine etki eden bu hareket yuvarlanma hareketinin nasıl yapıldığını gösterir. Pozitif değişken Δ_A dikey itki kuvvetini değiştirmeyecek şekilde seçilir.



Şekil 4.7. Yuvarlanma hareketi

4.5.3. Yunuslama Hareketi

Bu hareket, yuvarlanma hareketiyle çok benzerdir ve arka pervanesinin hızı azaltılırken (arttırılırken), ön pervane hızının arttırılıp (azaltılmasıyla) elde edilen hareketi tanımlar. AR.Drone'u Şekil 4.8'de görüldüğü gibi y-ekseni boyunca döndürecek şekilde bir tork uygulanır. Tüm dikey itki kuvvetlerinin, havada sabit tutunma pozisyonunda olduğu gibi, eşit değerde iken bu hareket, sadece yunuslama açısının ivmelenmesine neden olur. Şekil 4.8'de hava aracı üzerine etki eden yunuslama hareketi gösterilmektedir.

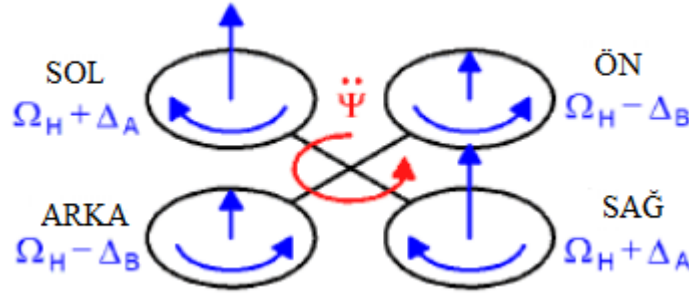


Şekil 4.8. Yunuslama hareketi

4.5.4. Yönelme Hareketi

Bu hareket, ön ve arka pervanelerin hızları arttırılırken (azaltılırken), sağ-sol pervane çiftinin hızlarının arttırılmasıyla (azaltılmasıyla) elde edilir. Dört rotorlu hava aracını z-ekseni boyunca döndürecek şekilde bir tork ile uygulanır. Sapma hareketi, sağ-sol pervanesinin saat yönünde dönerken, ön- arka pervane saat yönünün tersine dönmesiyle Şekil 4.9'daki gibi elde edilir. Bu yüzden, bütün torklar dengelenmemiş durumda iken, AR.Drone kendi etrafında z-ekseni boyunca döner. Toplam dikey itki kuvvetlerinin, havada sabit tutunma pozisyonunda olduğu gibi, eşit değerde iken bu hareket, sadece sapma açısının ivmelenmesine neden olur.

AR.Drone'nun dört pervanesi eşit hızlarda ve çiftler arasında zıt yönde olmak üzere döndürüldüklerinde eşit ve zıt yönde torklar üretirler. Bunun sonucu olarak toplam tork sıfır olur ve sabit bir yönelme açısı elde edilmiş olur. Çiftlerden birindeki pervanelerin hızı diğer çiftte ki pervanelere göre artırıldığı zaman net tork artık sıfır olmayacak ve araç yönelme hareketine başlayacaktır. Aracın yönelmesi hızı yüksek olan rotor çiftinin tersi yönünde olacaktır. Söz konusu manevra şekil 4.9'da gösterilmiştir.



Şekil 4.9.Yönelme hareketi

4.5.5. AR.Drone Haberleşmesi

AR.Drone haberleşmesi Wi-Fi yöntemi ile gerçekleştirilmektedir. Wi-Fi bilindiği üzere farklı bilgisayarları veya cihazları kablosuz olarak ortak bir ağ üzerinden haberleştirmeyi sağlar. Wi-Fi radyo vericisi gibi radyo dalgalarını kullanır söz konusu bu

dalgaların frekansı 2.4GHz, 3.6GHz, ya da 5GHz seviyelerindedir. Wi-Fi adaptörleri iletişim için dijital kodları radyo sinyallerine dönüştürürler ve gönderilen bilgiyi yorumlamak için ise radyo sinyalleri dijital kodlara çevrilirler. Wi-Fi ile haberleşmenin avantajları, diğer uzaktan kumanda metotlarında olduğu gibi herhangi bir kablo gerektirmez ve iletişim kablosuz olarak gerçekleştirilir. Wi-Fi birden çok cihaza bağlantı sağlar ve çok noktadan kontrol sağlanır. Ayrıca bu iletişim metodunda güvenlik duvarı olduğundan dışarıdan gelecek parazitler veya yabancı alıcılar engellenmiş olur ve daha güvenli bir kumanda işlemi gerçekleştirilmiş olur. Dezavantajları ise, verici noktadan uzaklaştıkça bağlantı hızı ve doğruluğu oldukça fazla bir şekilde etkilenmektedir. Wi-Fi ile çalışan cihazlar diğer haberleşme teknolojileri ile çalışan cihazlara nazaran daha fazla güç tüketirler böyle olunca hem güç tüketimi hem de batarya boyutları büyük olurken aktif çalışma süresi de o denli küçük olmaktadır.

5. SÜREKLİ UYARLAMALI ORTALAMA KAYMA ALGORİTMASI İLE HAREKETLİ NESNE TAKİBİ

5.1. OpenCV Yazılımı

OpenCV farklı takip ve tanıma algoritmalarını çalıştırmak için seçilen açık kaynak kodlu bilgisayarlı görme ve makine öğrenmesi yazılımı kütüphanesidir. OpenCV bilgisayarlı görme ve görüntü işleme tekniklerinin uygulanabileceği birçok algoritmayı barındıran bir kütüphanedir. Bilgisayarlı görme uygulamalarında gerçek zamanlı görüntü işlemenin uygulanması amaçlanmıştır ve bu gereksinimlerin karşılanması için optimize edilmiştir. OpenCV kütüphanesindeki algoritmalar nesneleri tanıma, videolardaki insan hareketlerini sınıflandırma, kamera hareketlerini takip etme, yüz algılama ve nesnelerin üç boyutlu modellenmesi gibi algoritmalar sayılabilir.

OpenCV'nin bu çalışma için MATLAB gibi algoritmalar yerine seçilmesinin farklı nedenleri vardır. OpenCV diğer programlama dillerine göre makine diline daha yakın olan C/C++ ile yazılmıştır. C/C++ programlama dili algoritmanın hızlı bir şekilde çalıştırılmasını sağlar. Görüntü işleme için daha fazla işlem döngüsü bırakılarak çalıştırılma ihtiyacı duyan kod yorumlama miktarını azaltmaktadır. OpenCV gerçek zamanlı olarak çalıştırılması için 70 Mb'lık RAM'e ihtiyaç duyar. Windows, Linux ve Mac OS tarafından desteklenmektedir.

5.2. AR.Drone ile Bağlantı Kurma

AR.Drone ile haberleşmenin yapılması için Parrot yazılım geliştirme kiti (SDK) 2.0 kullanılmıştır. AR.Drone bilgisayara bağlanmak için Wi-Fi bağlantı kullanmaktadır. AR.Drone Wi-Fi bağlantı desteği olan herhangi bir cihaz ile kontrol edilebilir. AR.Drone çalıştırıldığında Wi-Fi ağ oluşturmaktadır. Wi-Fi ağ ile ardrone_aaa formatında ESSID tanımlanır. Burada aaa rasgele bir sayı dizisidir. Daha sonra AR.Drone ağ üzerinde sabit bir IP adresi tanımlar. Bir sonraki aşamada istemci cihaz ile tanımlanan bu ağ ile bağlantı kurulur. İstemci cihaz akıllı telefon ya da bilgisayar olabilir. AR.Drone'nun bir sunucu gibi çalışması nedeniyle istemci cihaz AR.Drone'nun DHCP sunucusundan talep eder. AR.Drone'nun DHCP sunucusu istemci programa bir IP adresi tahsis eder. Bu IP adresi

AR.Drone'un IP adresi ve 1 ile 4 arasında deęişen bir rasgele sayıdır. Daha sonra istemci cihaz AR.Drone komut gönderebilir ve servis portu üzerinden veri almaya başlar.

AR.Drone yazılım geliştirme kiti 3 ana servis kullanılarak yapılmaktadır. Bunlar uçuş kontrolü ve konfigürasyon servisi, navigasyon veri servisi ve video görüntüleri servisi.

AR.Drone'un konfigürasyon ve uçuş kontrolü AT komutları yardımıyla yapılır. AT komutlar 5556 portu üzerinden UDP paketleri ile AR.Drone'a gönderilir. AT komutları AR.Drone'a düzenli bir şekilde gönderilir. AR.Drone yaklaşık olarak saniyede 30 AT komutu alır.

AR.Drone'un navigasyon veri servisi droneun durumu ile ilgili bilgi sağlamaktadır. Bu bilgi AR.Drone'un konumu, çok yönlü hızı, oryantasyon, motor devir sayısı ve yükseklik bilgilerini içermektedir. Navigasyon verisi AR.Drone'un istemci cihazına 5554 portu üzerinden UDP paketleri ile gönderilir.

Video görüntüleri servisi 5555 portu üzerinden TCP paketleri ile AR.Drone'un istemci cihazına gönderilir. AR.Drone'un kullandığı video sıkıştırma teknięi H264(MPEG4.10 AVC)'dir. Bu sıkıştırma teknięi yüksek kalitede video görüntülerinin alınması ve kaydedilmesi için kullanılır. Video görüntülerinin belirli parametreleri isteęe baęlı olarak uyarlanabilir. Saniyede alınan görüntü sayısı 15 FPS olarak ayarlanabilir. Bit oranı 250 kbps ile 4Mbps arasına ayarlanabilir ve çözünürlük ise 640x360 ya da 1280x720 olarak ayarlanabilir. İsteęe baęlı olarak AR.Drone görüntüleri kaydedebilir. Kaydedilen bu görüntüler 5553 portu üzerinden TCP paketleri ile AR.Drone istemcisine gönderilir. Video dizileri kaydetmek için H264-720 piksel çözünürlüklü görüntüler gönderir. AR.Drone kaydetme işlemini sonlandırdığında video görüntülerinin alımı durur.

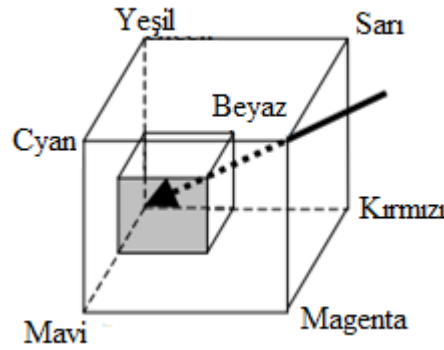
5.3. Sürekli Uyarlamalı Ortalama Kayma Algoritması

SUOK, sürekli uyarlamalı ortalama kayma (Continuously Adaptive Mean Shift Algorithm-CAMSHIFT) algoritmasının kısaltılmış şekilde ifade edilmesidir. SUOK algoritması ortalama kayma algoritması (OK-Mean Shift) esasına dayanan bir algoritmadır. OK algoritması olasılık dağılımlarını işleyen bir algoritmadır. Görüntü dizilerindeki renkli nesneleri takip etmek için renkli imge verisi olasılık dağılımı olarak temsil edilmesi gerekmektedir. Bu amaç doğrultusunda renk histogramları kullanılmaktadır. Görüntü dizilerinden alınan renk dağılımları zamanla deęişmesi sebebi ile OK algoritması takip

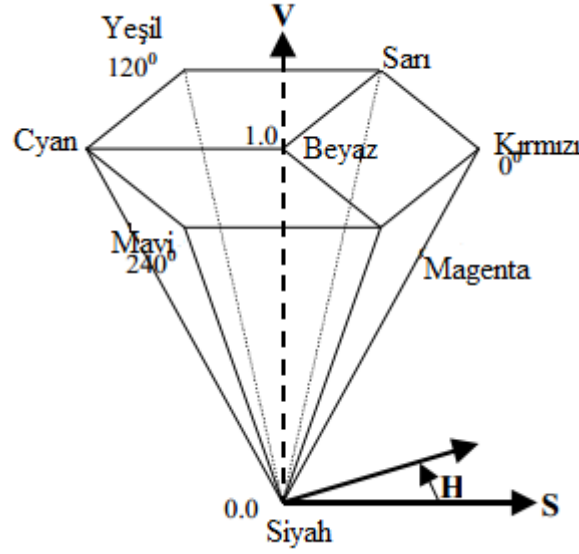
ettiği olasılık dağılımına dinamik olarak uyarlanabilmesi için düzeltilme ihtiyacı duymaktadır. SUOK algoritması tüm bu gereksinimleri karşılamaktadır. Nesne takibi için SUOK nesneyi temsil eden renk olasılık dağılımının alanı ve x,y koordinatları hesaplanmaktadır.

5.3.1. Renk Uzayı

Bir video dizisindeki renkli nesnelerin SUOK algoritması ile takibi işlemi arzu edilen rengin olasılık dağılımı imgesi oluşturularak yapılabilmektedir. Bu işlemin gerçekleştirilmesi amacıyla ilk önce renk histogramı kullanılarak arzu edilen renk özüne ait model çıkartılır. Şekil 5.1'deki standart RGB renk uzayına karşılık gelen Şekil 5.2'deki HSV renk uzayı kullanılır. HSV renk uzayı renk özü özelliğini yoğunluk ve parlaklıktan ayırabilmektedir. HSV uzayında 1 boyutlu histogramı alınarak renk modelleri oluşturulur.



Şekil 5.1. RGB renk uzayı



Şekil 5.2. HSV renk uzayı

Takip işlemi boyunca kaydedilen renk histogramı alınan görüntü dizilerini nesnenin ilgili olasılığına dönüştürüp model olarak kullanılır. Bu işlem tüm video kareleri için yapılmaktadır. Bu yöntem kullanılarak olasılıklar ayırık adımlarla sıfırdan (olasılık 0.0) maksimum (olasılık 1.0) değere kadar değişmektedir. 8 bitlik renk değerleri için bu aralık 0 ile 255 arasında değişmektedir. Yapılan işlemler doğrultusunda hedef nesnenin olasılığı ile SUOK algoritması kullanılarak takip işlemi gerçekleştirilmektedir.

Ayrık piksel değerli gerçek kameralar ile görüntü alınırken HSV uzayı kullanıldığında bazı sorunlara neden olabilmektedir. Parlaklığın düşük iken doygunlukta düşük olursa renk özü gürültülü olmaktadır. Bu durum renk değerlerinde aşırı değişimlere yol açmaktadır. Bunu düzeltmek için düşük parlaklığa sahip renk pikselleri ihmal edilir. Mat sahneler için parlaklık arttırılmalı aksi halde takip işlemi için uygun şartlar sağlanmamaktadır. Aşırı parlaklığın olduğu ortamlarda ise üst eşik değeri belirlenmesi gerekmektedir. Düşük doygunlukta renk tam olarak belirlenememesi sebebi ile düşük doygunlukta renk pikselleri de göz ardı edilmektedir. RGB renk uzayının ortam aydınlığının değişmesine karşı hassas olması sebebi ile HSV renk uzayı tercih edilmiştir.

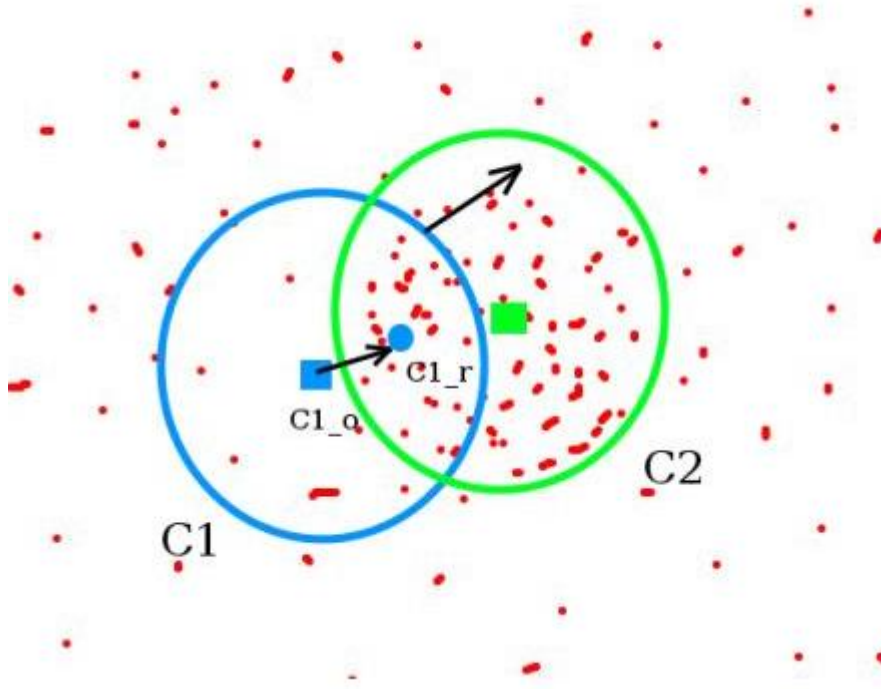
5.3.2. SUOK Algoritmasının Çalışma Prensipleri

Nesne ve yüz takibi için kullanılan bir yöntemdir. OK algoritmasında arama penceresi boyutu, nesnenin kameraya yaklaşması ve kameradan uzaklaşması durumunda aynı kalmaktadır. Bu istenilen bir durum değildir. Çözüm olarak SUOK algoritması sunulmuştur. OK algoritmasının uyarlanabilir bir bölgenin boyutlandırması adımı ile birleştirilmesiyle oluşturulmuştur. SUOK algoritması ise anlık görüntü karesinin dağılımının sıfırıncı momentine bağlı olarak bir sonraki görüntü karesi için arama penceresinin boyutunu tekrar ayarlayabilen dinamik dağılımları kullanabilen bir algoritmadır [86].

OK algoritması parametrik olmayan istatistiksel bir yöntem olup renkli bir nesnenin merkezini bulmaya yarayan olasılık yoğunluk dağılımının maksimum noktasının tespit edilmesi için kullanılır. Temel olarak, piksel dağılımını gösteren bir dizi nokta olsun. Bu noktalar dizisi histogram geri izdüşümü gibi bir piksel dağılımı olabilir. Öncelikle bu piksel dağılımı üzerinde küçük bir pencere belirlenir ve algoritmanın amacı bu pencereyi şekildeki gibi maksimum piksel yoğunluğunun olduğu alana taşımaktır.

Başlangıç penceresi Şekil 5.3'te olduğu gibi mavi renkli C1 çemberi seçilir. C1 çemberinin yarıçapı C1_o olarak belirlenir. C1 penceresi içerisindeki piksellerin yoğunluğuna bakılırsa merkez C1_r olarak alınır. Dolayısıyla arama penceresinin merkezi C1_o noktasından C1_r noktasına kaydırılır. Yeni penceredeki piksellerin ortalaması tekrar hesaplanıp merkez bulunur. Bu işlem belirli iterasyonda maksimum piksel yoğunluğunun olduğu C2 merkezli C2 arama penceresine ulaşılır. Böylece, histogram geri izdüşümü imgesi ve nesnenin başlangıç yeri sürekli değişerek maksimum yoğunluğun olduğu yere ulaşılır.

Genel olarak takip işlemine başlamadan önce takip edilecek nesne seçilerek histogram geri-yayılım imgesi ve başlangıç konumu belirlenir. Histogram geriye izdüşüm imgesinde nesnenin hareketi izlenir. Ve sonuç olarak, OK algoritması arama penceresini maksimum yoğunluğun olduğu bölgeye doğru hareket ettirir.



Şekil 5.3. Arama penceresinin maksimum yoğunluk dağılım bölgesine hareketi

İlgili bölgedeki renk dağılımını temsil etmek için literatürde parametrik ve parametrik olmayan farklı yöntemler kullanılmaktadır. Histogram en sık kullanılan parametrik olmayan olasılık tahmin yöntemidir. Histogram, ilgili bölgedeki piksellerin renk dağılımı bulunarak oluşturulur. Bu sayede benzer renkler histogramda aynı grupta yer alır ve aynı olasılık değerine sahip olur. Oluşturulan bu histogram modeli aşağıdaki formülde olduğu gibi hesaplanmaktadır.

$$\hat{q} = \sum_{i=1}^n \delta[c(x_i) - u] \quad (5.1)$$

$$c: R^2 \rightarrow \{1...m\} \quad (5.2)$$

Denklem (5.1)'de n nesnenin bulunduğu bölgedeki toplam piksel sayısını, δ kronecker delta fonksiyonunu ve $c(x_i)$ fonksiyonu ise x_i pikselinin piksel değerini ifade etmektedir. Özellik olarak kullanılan nesnenin renk dağılımı, takip edilen nesnenin kameraya olan uzaklığının artıp azalmasıyla farklılık göstermemektedir. Olasılık yoğunluk dağılımının görüntüsü $I(x,y)$, histogram yardımıyla elde edilir. Bu işlem giriş görüntüsündeki piksel değerlerini o pikselin renk değerine ait histogram değeri ile

değiştirir. Bu sayede olasılık görüntüsündeki her bir pikselin değeri o pikselin nesne modelinde yer alma olasılığı ile temsil edilmiş olmaktadır. Olasılık değerlerinin 0 ile 255 arasındaki değerleri alabilmesi için histogram değerleri Denklem (5.3)'teki gibi normalize edilmektedir.

$$p_u = \min\left(\frac{255}{\max(\hat{q})} \hat{q}_u, 255\right) \quad (5.3)$$

SUOK Algoritmasının uygulanması

1. Takip edilmesi gereken arama penceresinin yeri seçilir. Bu arama penceresi OK arama penceresi olmaktadır. Burada takip edilecek nesnenin renk özü ayarlanır.

2. OK penceresinin merkezinde toplanan bölgenin seçiminin olasılık dağılımı hesaplanır ve böylece nesneyi ifade eden renklerin histogramı olarak temsil edilir. Histogramda düşeyde verilen her bir değer seçilen bölgede renk özüne sahip piksellerin toplamını ifade eder.

3. Olasılık imgesinin merkezini bulmak için OK algoritması tekrar edilir. Merkez noktası sıfırcı moment olarak kullanılır. Böylece arama penceresinin yeni merkezi belirlenir. Arama penceresi hesabı 0. Momentlerin bir fonksiyonu olarak bulunur.

4. Sonraki tüm görüntü kareleri için arama penceresi yeni merkeze taşınır ve işlemler 2. adımdan itibaren tekrar ettirilir.

$I(x,y)$ arama penceresi içerisinde x,y koordinatlarındaki ayrık olasılık dağılımının parlaklığını ifade eder. Sıfırcı moment aşağıdaki Denklem (5.4)'deki gibi bulunur.

$$M_{00} = \sum_x \sum_y I(x, y) \quad (5.4)$$

Arama penceresinin merkez koordinatlarını hesaplamak için M_{10} ve M_{01} ilk moment değerleri Denklem (5.5) ve (5.6)'da olduğu gibi hesaplanır.

$$M_{10} = \sum_x \sum_y xI(x, y) \quad (5.5)$$

$$M_{01} = \sum_x \sum_y yI(x, y) \quad (5.6)$$

Arama penceresinin yeri aşağıdaki denklemlere göre bulunur.

$$x_c = \frac{M_{10}}{M_{00}} \quad (5.7)$$

$$y_c = \frac{M_{01}}{M_{00}} \quad (5.8)$$

Olasılık dağılımının iki boyutlu oryantasyonu SUOK algoritmasının işlenmesi sürecinde takip edilecek nesneye ait ikinci momentler kullanılarak elde edilir. İkinci momentler Denklem (5.9) ve (5.10)'daki gibi hesaplanır.

$$M_{20} = \sum_x \sum_y x^2 I(x, y) \quad (5.9)$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y) \quad (5.10)$$

θ , nesne oryantasyonu Denklem (5.11)'deki gibi hesaplanır.

$$\theta = \frac{\arctan \left(\frac{2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2 \right) - \left(\frac{M_{02}}{M_{00}} - y_c^2 \right)} \right)}{2} \quad (5.12)$$

Olasılık dağılımının uzunluğu ve genişliği, Denklem (5.13), (5.14) ve (5.15)'de verilen a, b ve c ifadeleri kullanılarak sırayla Denklem (5.16) ve (5.17)'de olduğu gibi SUOK algoritması tarafından hesaplanır.

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad (5.13)$$

$$b = 2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right) \quad (5.14)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \quad (5.15)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (5.16)$$

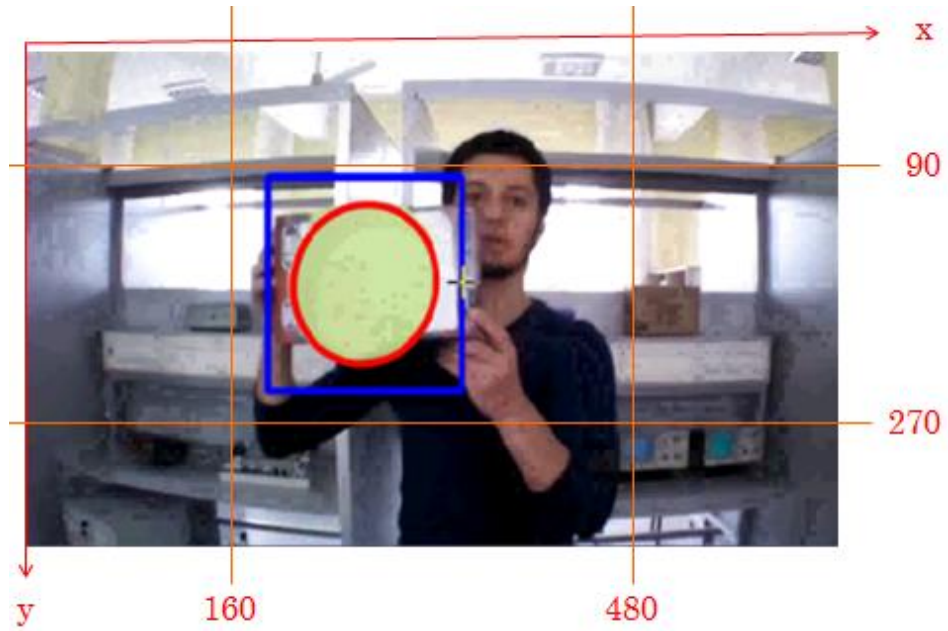
$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (5.17)$$

$$s = 2 * \sqrt{\frac{M_{00}}{256}} \quad (5.18)$$

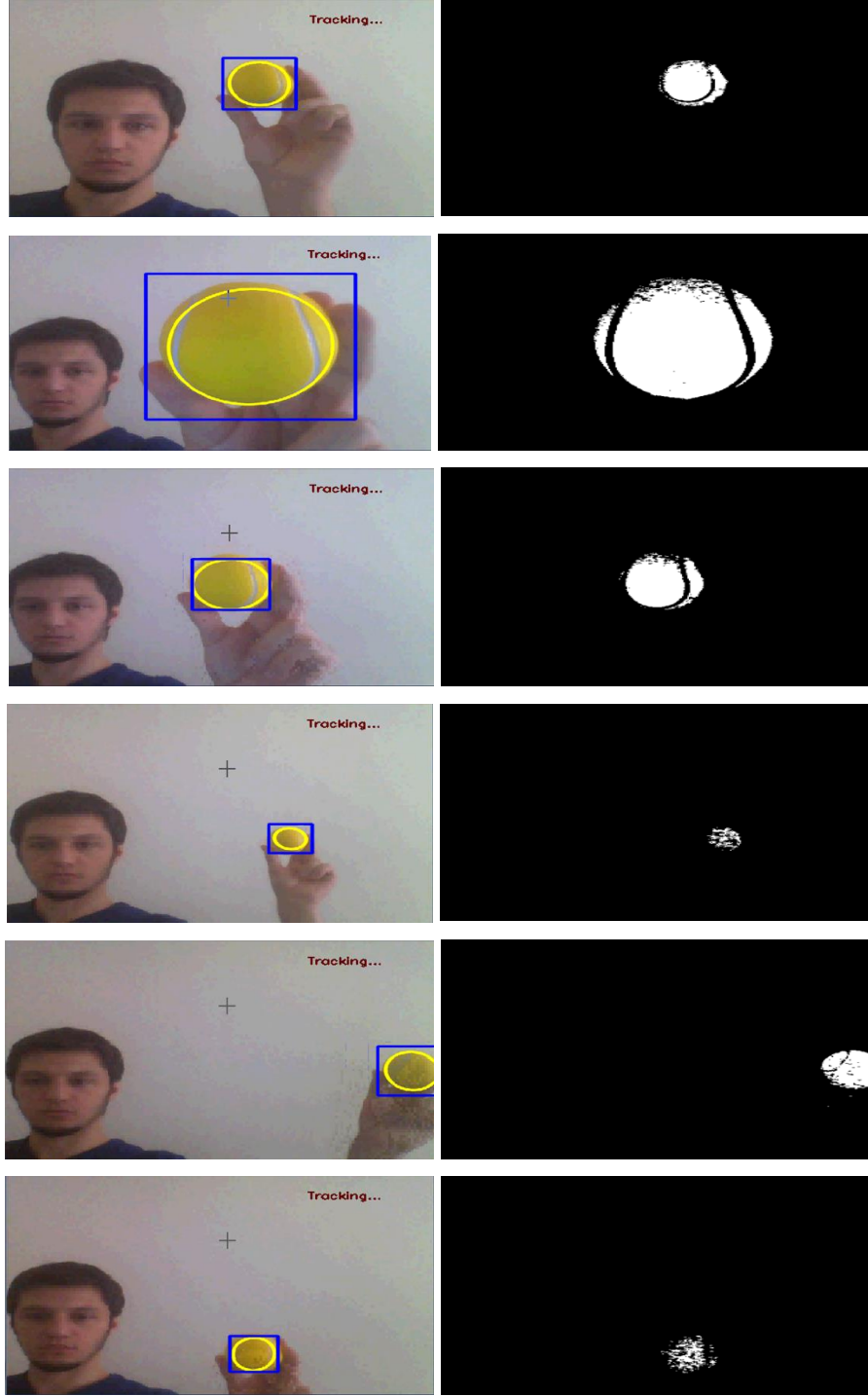
Denklem (5.18)'de görüldüğü gibi arama penceresi adaptif olarak ayarlanması 0. momentin bir fonksiyonu olarak bulunur. Histogramda dağılıma bakıldığında maksimum değer 256 olarak görünmektedir. Bu nedenle 0. moment 256 sayısına bölünür. İki boyutlu renk olasılık dağılımının tek boyuta indirgenmesi için bu değer karekökü alınır ve arama penceresinin alanı bulunur.

5.3.3. Uygulamanın Gerçekleştirilmesi

Bu tez çalışmasında ilk olarak sürekli uyarlamalı ortalama kayma algoritması ile sabit kamera kullanarak farklı renkli nesnelerin takibi gerçekleştirilmiştir. Şekil 5.5 ve 5.6’da görüldüğü gibi hareketli nesnenin kameraya yaklaşma, uzaklaşma ve oryantasyon değişimleri durumunda başarılı sonuçlar alınmıştır. Şekil 5.5’te takip edilen nesnenin bazı kısımları farklı renkten oluşmaktadır. Buna rağmen algoritma nesnenin olasılık yoğunluk dağılımını tatmin edici düzeyde bulup nesneyi takip edebilmektedir.

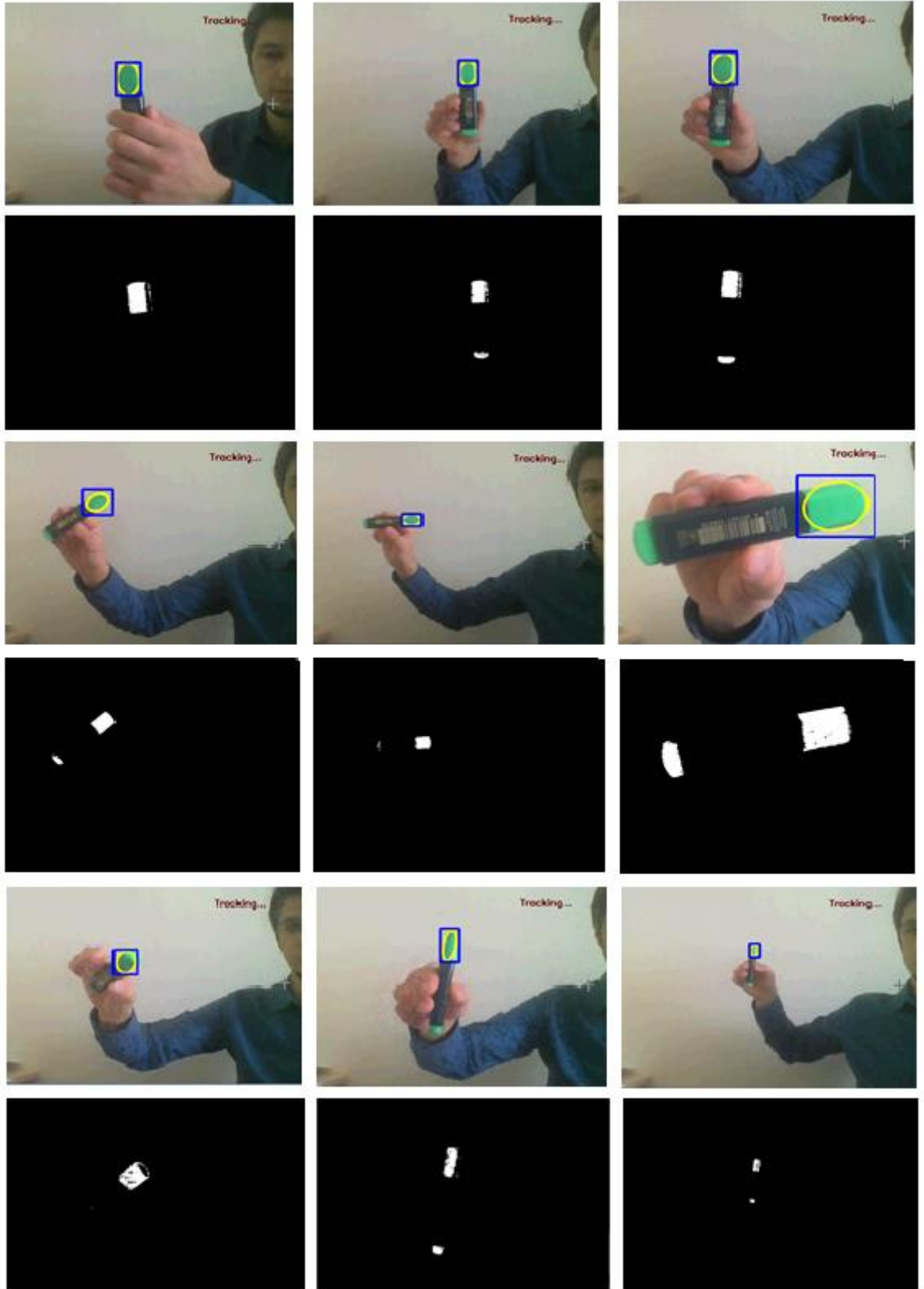


Şekil 5.4. Kameradan alınan görüntülerde nesnenin bulunduğu noktaya ait konum bilgilerinin tespit edilmesi yöntemi



Şekil 5.5. Sabit Kamera ile Sarı top takibi uygulaması sonuçları

Şekil 5.6’da ise takip edilen nesnede aynı renge sahip iki farklı bölge bulunmaktadır. Buna rağmen kamera sürekli olarak renk olasılık dağılımının en yüksek olduğu alanı takip etmektedir.



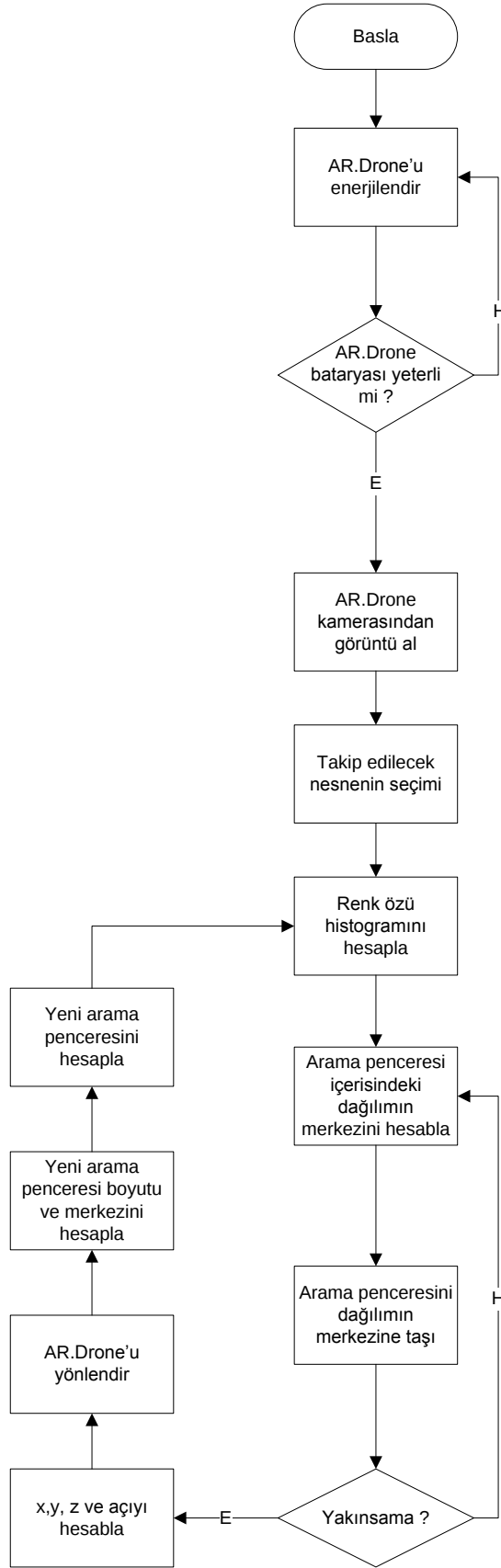
Şekil 5.6. Sabit Kamera ile yeşil renkli fosforlu kalem takibi uygulaması sonuçları

İkinci aşamada ise AR.Drone'un ön kamerasından alınan görüntüler sürekli uyarlamalı ortalama kayma algoritması kullanılarak takip edilen nesne AR.Drone ile takip edilmiştir. Yapılan uygulamanın kullanıldığı ortam, hareketli nesne ve AR.Drone ve PC Şekil 5.7'de gösterilmiştir. Takip işlemi takip edilecek nesne seçilmesi ile başlamaktadır. Seçilen alanın renk olasılık dağılımından yola çıkarak takip edilecek nesnenin arama alanı belirlenir. Bu arama alanında nesnenin her bir iterasyonda konum bilgisi tespit edildikten sonra nesneyi takip için AR.Drone pozisyon kontrolü yapılmaktadır. Bu işlemi gerçek zamanlı olarak uygulanabilirliğini görmek amacıyla Şekil 5.7'de görülen uygulama ile bazı testler yapılmıştır.



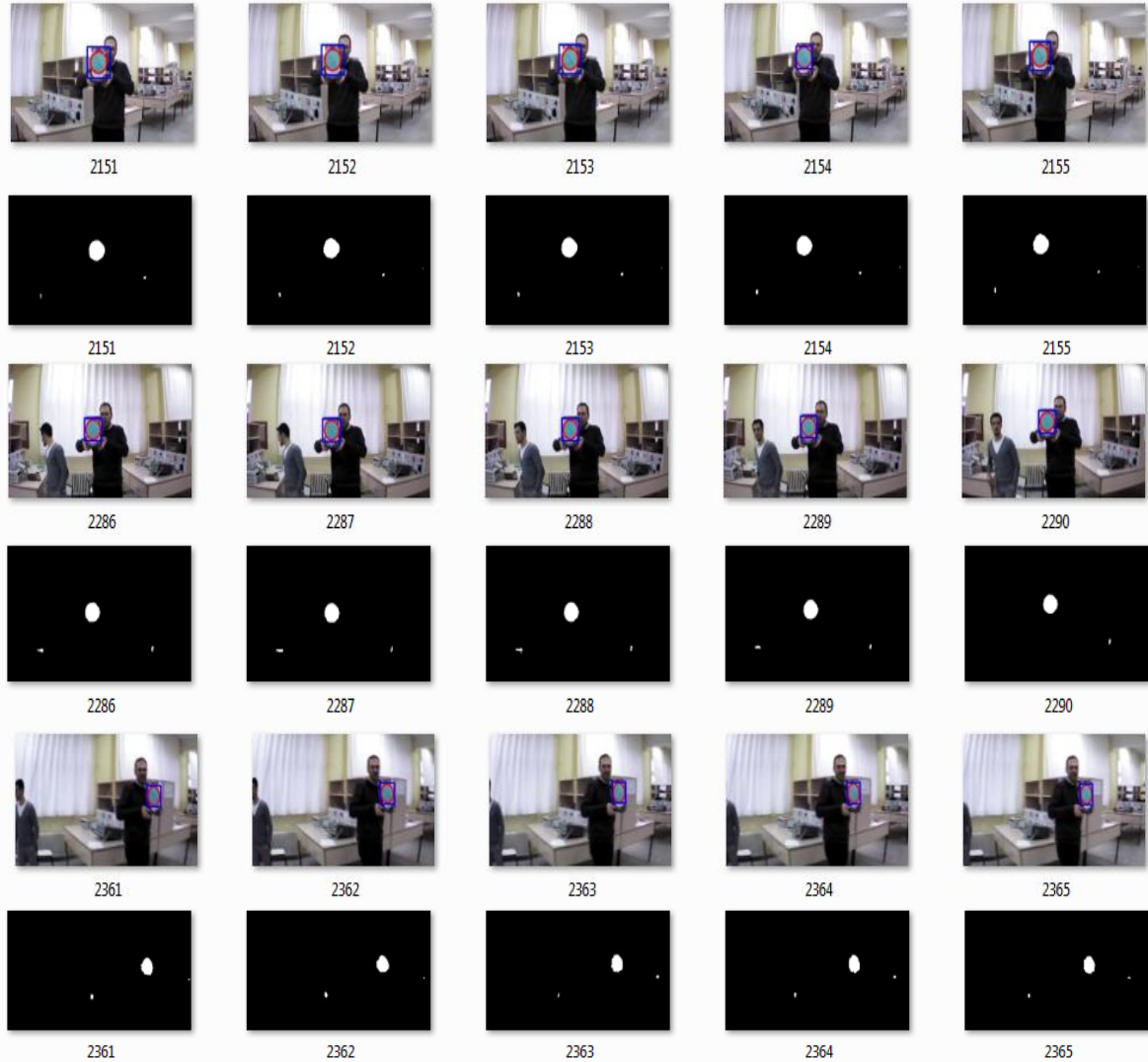
Şekil 5.7. AR.Drone ile hareketli nesne takibi için yapılan uygulama görüntüsü

Takip edilen nesnenin hareketli olması sebebi ile kameradan alınan görüntülerde yeri devamlı olarak değişmektedir. Hareketli nesne aşağı, yukarı, sağa, sola, ileri ve geri olmak üzere farklı hareketlerde bulunabilmektedir. Hareketli nesnenin hareketi sırasında kameradan uzaklaşması veya kameraya yaklaşması halinde kameradan alınan görüntülerde nesnenin boyutu azalmakta veya artmaktadır. Böylece takip işlemi boyunca hareketli nesneyi takip etmekte kullanılan pencerenin boyutu dinamik olarak değişmesi gerekmektedir. Dolayısıyla burada SUOK algoritmasının bu çalışmada kullanılmasının önemi görülmektedir. AR.Drone ile hareketli nesne takibi uygulamasına ait akış diyagramı Şekil 5.8'de verilmiştir. Ayrıca AR.Drone ön kamerasından alınan görüntüler Şekil 5.9 ve 5.10'da görüldüğü gibi arka planda renkli ve geri yayılım imgeleri ile kaydedilmektedir. Yapılan uygulamada kayıt işlemi 1290. imgeden sonra başlatılmıştır.



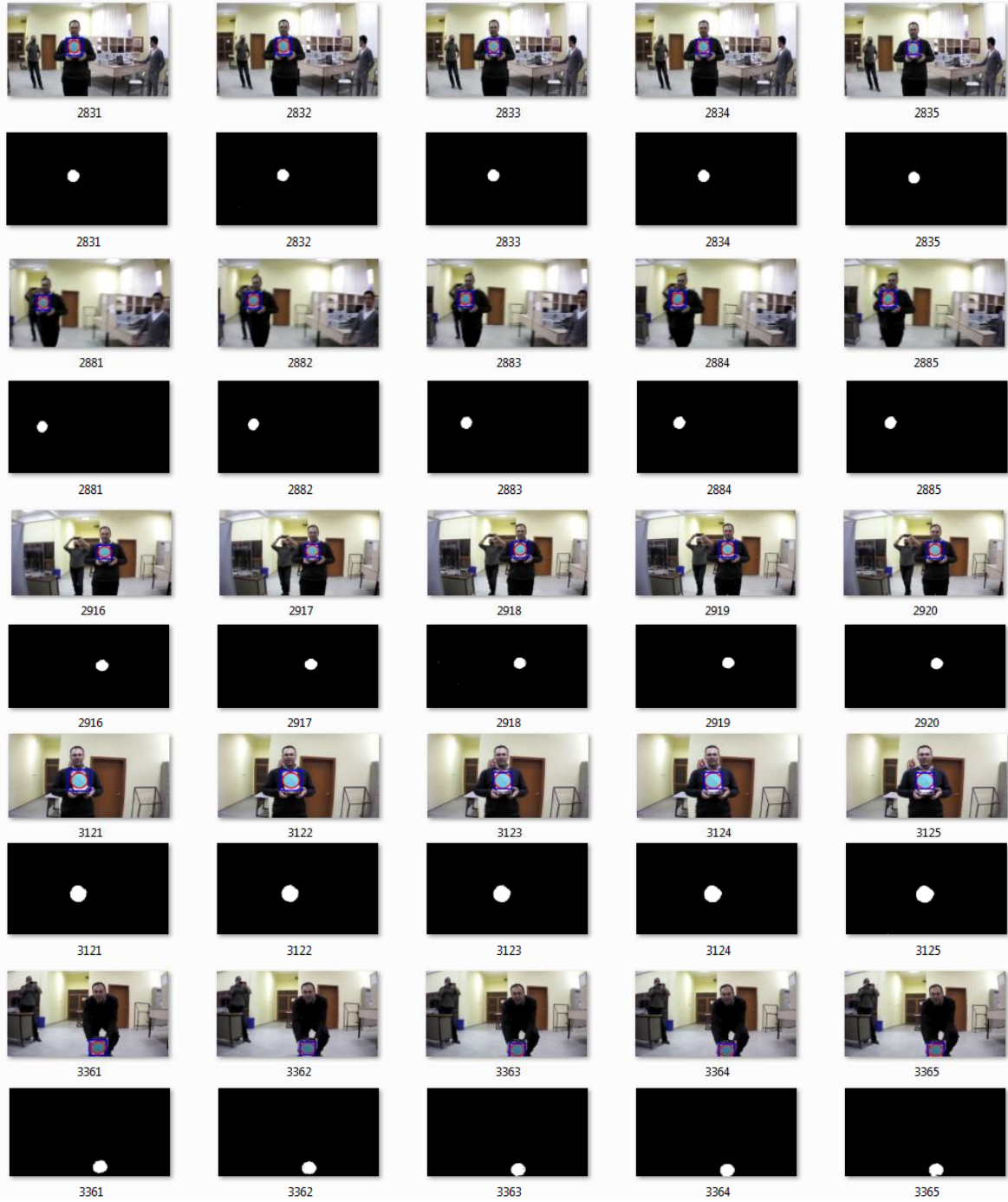
Şekil 5.8. AR.Drone ile hareketli nesne takibi uygulamasına ait akış diyagramı

Nesne takibi, genel olarak hareketli nesneyi temsil edebilen uygun pencere boyutunun seçilmesi ve nesnenin bu pencere içerisindeki olasılık yoğunluk fonksiyonu ile temsil edilmesiyle gerçekleştirilmektedir. Hareketli nesne AR.Drone'un ön kamerasına yaklaştıkça veya uzaklaştıkça hareketli nesnenin görüntüdeki boyutu değişmektedir. Boyutun değişmesi takip işleminin sabit boyutlu bir pencere ile yapılmasını zorlaştırmaktadır. Bu sebep ile nesnenin boyutunun değişimine karşılık pencere boyutunun da tekrar hesaplanması gerekmektedir. Kameradan alınan görüntülerde nesnenin görüntüde bulunduğu konum şekilde verilen koordinat değerlerine göre SUOK algoritması ile Şekil 5.9'da görüldüğü gibi hareketli nesnenin sağa ve sola gitmesi durumunda AR.Drone yönelme hareketi sayesinde takip işlemini başarıyla gerçekleştirmektedir.



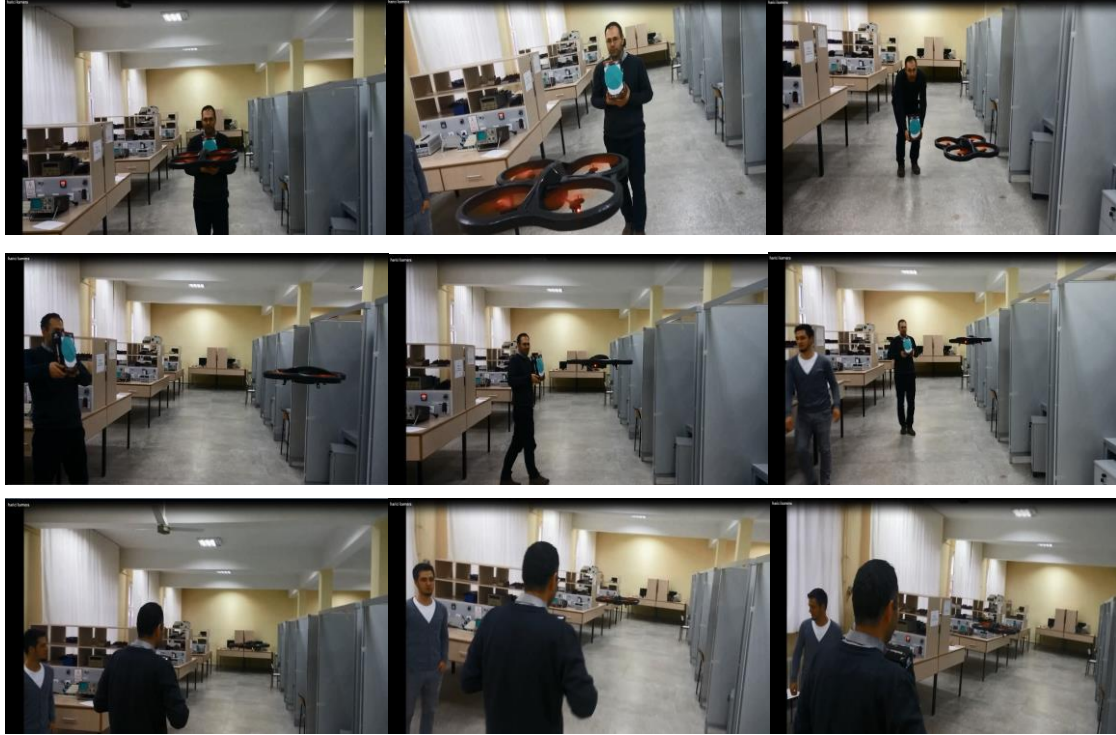
Şekil 5.9. AR.Drone ile geri yayılım imgeleri elde edilmiş hareketli nesne takibi sonuçları

AR.Drone ile yapılan bir başka uygulamada Şekil 5.10’da görüldüğü gibi geri yayılım imgeleri elde edilerek takip edilen nesnenin kameradan uzaklaşmasıyla nesnenin görüntüde kaplamış olduğu alan azaldığı ve buna bağlı olarak pencere boyutunun da küçüldüğü görülmektedir. Benzer şekilde nesnenin kameraya yaklaşması durumunda ise nesnenin görüntüde kaplamış olduğu alan artmaktadır ve dolayısıyla pencere boyutu büyümektedir. Hareketli nesnenin kameradan alınan görüntüde düşey doğrultuda bulunduğu noktaya bağlı olarak aşağı ya da yukarı yönde hareket etmektedir.



Şekil 5.10. AR.Drone ile geri yayılım imgeleri elde edilmiş hareketli nesne takibi sonuçları

Takip edilen nesnenin AR.Drone kamerasına olan uzaklığına bağılı olarak görüntüde kaplamış olduğı alan hesaplanıp daha önce belirlenen sınır deęerlerinin dıřına ıkması durumları tespit edildikten sonra AR.Drone takip edilen nesneyi ynelme, ykselme, alalma ve yunuslama hareketleri sayesinde x,y ve z doęrultularında bařarılı bir řekilde takip edilmektedir. Yapılan uygulamalarda farklı aılardan hareketli nesnenin AR.Drone ile nasıl takip ettięi řekil 5.11’de grlmektedir.



řekil 5.11. AR.Drone ile hareketli nesne takibinin dıřarıdan izlenmesi

6. SONUÇLAR

Bu tezde dört rotorlu insansız hava aracı olarak Parrot firması tarafından üretilen Parrot AR.Drone 2.0 Power edition versiyonu kullanılmıştır. AR.Drone üzerinde bulunan ön kamera ile alınan görüntüler ile yönelme, yükselme, alçalma ve yunuslama hareketleri sayesinde x, y ve z eksenlerinde hareketli nesne takibi gerçekleştirilmiştir. Hareketli nesne takibi sırayla sabit kamera ve AR.Drone ile başarılı bir şekilde yapılmıştır. Tez kapsamında nesne takibinde kullanılan özellikler ve yöntemleri araştırılmıştır. Bu yöntemlerden sürekli uyarlamalı ortalama kayma algoritması ile nesne takibinin gerçekleştirilmesine karar verilmiştir. Kameradan alınan görüntülerde takip edilmesi istenen nesne seçilip takip işlemine başlanmıştır. Hedef nesnenin kameraya yaklaşmasına ve uzaklaşmasına karşılık AR.Drone hareketli nesneyi başarılı bir şekilde takip etmiştir.

Bir video dizisindeki renkli nesnelerin sürekli uyarlamalı ortalama kayma algoritması ile takibi işlemi arzu edilen rengin olasılık dağılımı imgesi oluşturularak yapılabilmektedir. Bu çalışmada takip işleminin gerçekleştirilmesi amacıyla ilk önce renk histogramı kullanılarak takip edilecek nesneye ait model çıkartılıp bir sonraki görüntü karesinde yeri bulunarak takip edilmektedir.

Nesne takibi işlemi takip edilecek olan nesnenin görüntüde seçilmesiyle başlamaktadır. İlk olarak seçilen hareketli nesnenin görüntüdeki yer bilgisi ve renk olasılık dağılımı bulunur. Daha sonra bu bilgiler kullanılarak gerçek zamanlı olarak sıralı görüntü kareleri üzerinde hareketli nesne aranmaktadır. Nesnenin aranacağı alanın merkezi bir önceki görüntü karesinde nesnenin bulunmuş olduğu yerin merkezi referans alınarak belirlenir. Arama alanının genişliği takip edilen nesneye ait piksellerin renk olasılığı ve piksellerin yer bilgisinden faydalanılarak hesaplanan momentler ile bulunmaktadır. Takip edilen hareketli nesne sürekli olarak her bir iterasyonda yeri ve genişliği belirlenen arama alanı içinde aranmaktadır. Arama işlemi sürekli uyarlamalı ortalama kayma algoritmasındaki moment değerleri kullanılarak gerçekleştirilmektedir. Bu moment değerleri ile arama alanında takip edilen hareketli nesnenin renk olasılık dağılımının en yoğun olduğu bölge bulunmaktadır ve konumu bu bölgenin merkezi olarak belirlenmektedir. Takip edilen hareketli nesnenin görüntüde kaplamış olduğu alan ise moment değerleri kullanılarak bulunmaktadır. Elde edilen tüm sonuçlar bir sonraki aşamaya aktarılır. Böylece kameradan görüntü alımı sürdüğü müddetçe takip işlemi devam etmektedir.

Yapılan alıřmada AR.Drone kamerası ile takip edilen hareketli nesne arasına bařka bir nesnenin tamamen girmesi durumunda takip iřlemi bařarısızlıęa uęramaktadır. Bundan sonraki alıřmalarda bu olumsuz durumun ortadan kaldırılabilmesi iin hareketli nesnenin hız bilgisi ve hareket doęrultusu ile bir sonraki grnt karesindeki olası konumu tahmin edebilecek bir takip yntemi geliřtirilip gerek zamanlı olarak uygulanacaktır.

7. KAYNAKLAR

- [1] **A.Yılmaz,O.Javed ve M.Shah**, 2006, “Object Tracking: A Survey” ,ACM Comput. Surv. 38 , 4, Article 13.
- [2] **Soon-Yong Park, Chang-Joon Park ve Inho Lee**, 2005, “Moving Object Removal and Background Completion in a Video Sequence”, Image and Vision Computing New Zealand.
- [3] **Isaac Cohen ve Gerard Medioni**, 1998, “Detecting and Tracking Moving Objects in Video from an Airborne Observer”, Image Understanding Workshop,217-222.
- [4] **Qi Zang ve Reinhard Klette**, 2003, “Object Classification and Tracking in Video Surveillance”, CAIP, 198-205.
- [5] **Kye Kyung Kim, Soo Hyun Cho, Hae Jin Kim ve Jae Yeon Lee**, 2005 “Detecting and Tracking Moving Object Using an Active Camera”, Digital Object Identifier 10.1109/ICACT, 817-820
- [6] **Alper Yılmaz, X. Li ve M. Shah**, 2004, “Contour Based Object Tracking With Occlusion Handling in Video Acquired Using Mobile Cameras,” IEEE Trans. Pattern Anal. Mach. Intell. 26, 1531-1536.
- [7] **Dorin Comaniciu, V. Ramesh ve P. Meer**, 2000, “Real-TimeTracking of Non-Rigid Objects using Mean Shift”, Computer Vision and Pattern Recognition,142-149.
- [8] **Dorin Comaniciu, V. Ramesh ve P. Meer**, 2003, “Kernel-Based Object Tracking”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-575.
- [9] **Alper Yılmaz, X. Li, M. Shah**, 2004, “Object Contour Tracking Using Level Sets”, Asian Conference on Computer Vision, ACCV 2004, Jaju Islands, Korea.
- [10] **Matej Cerny ve Martin Dobrovolny**, 2013, “The Eye Tracking System for Positioning of the High Definition Camera”, 23th Conference Radioelektronika, Pardubice, Czech Republic.
- [11] **Tran Thi Trang ve Cheolkeun Ha**, 2013, “Irregular Moving Object Detecting and Tracking based on Color and Shape in Real-time System”, ComManTel International Conference, Vietnam.
- [12] **Fahad Fazal Elahi Guraya, Pierre-Yves Bayle ve Faouzi Alaya Cheikh**, 2009, “People Tracking via a Modified CAMSHIFT Algorithm”, The 8th DCABES.

- [13] **Woo-han Yun, Dohyung Kim, Jaeyeon Lee ve Jaehong Kim**, 2013, “Visual Object Tracking for Handheld Devices”, Industrial Electronics (ISIE) IEEE International Symposium, İstanbul.
- [14] **E. Altug, J.P. Ostrowski, and R. Mahony**, 2002, “Control of a quadrotor helicopter using visual feedback” In IEEE ICRA, pp. 72-77
- [15] **J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet**, 2010, “Vision-based navigation of unmanned aerial vehicles”, Control Engineering Practice, 18(7):789 – 799.
- [16] **B. Heriss, T. Hamel, R. Mahony, and F.-X. Russotto**, 2010, “A terrainfollowing control approach for a vtol unmanned aerial vehicle using average optical flow”, Autonomous Robots, pp. 1–19.
- [17] **S. Saripalli, J.F. Montgomery, and G.S. Sukhatme**, 2003, “Visually guided landing of an unmanned aerial vehicle. IEEE T. on Robotics and Automation”, 19(3):371 – 380.
- [18] **O. Shakernia, Y. Ma, T. John Koo, and S. Sastry**, 1999, “Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control”, Asian Journal of Control, 1:128–145.
- [19] **C. Teuliere**, 2010, “3d model-based tracking for uav position control”, In IEEE/RSJ IROS’10, Taipei, Taiwan.
- [20] **M. Meingast, C. Geyer, and S. Sastry.**, 2004, “Vision based terrain recovery for landing unmanned aerial vehicles.”, CDC, pp. 1670 – 1675.
- [21] **P. Perez, C. Hue, J. Vermaak, and M. Gangnet.**, 2002, “Color-based probabilistic tracking.”, In ECCV, pp. 661–675.
- [22] **Bettencourt, K., Somers, D. C.**, 2007, “Effects of task difficulty on multiple objecttracking performance”, Journal of Vison, 7(9):898, 898a.
- [23] **Veenman, C., Reinders, M., And Backer, E.**, 2001, “Resolving motion correspondence for densely movingpoints”, IEEE Trans. Patt. Analy. Mach. Intell. 23, 1, 54–72.
- [24] **Shafique, K. And Shah, M.**, 2003, “A Non-Iterative Greedy Algorithm For Multi-Frame Point Correspondence”, In IEEE International Conference On Computer Vision (Iccv). 110–115.

- [25] **Black, M. And Jepson, A.**, 1998, "Eigen tracking: Robust Matching And Tracking Of Articulated Objects Using A View-Based Representation", Int. J. Comput. Vision 26, 1, 63–84.
- [26] **Haritaoglu, I., Harwood, D., And Davis, L.**, 2000, "W⁴: real-time surveillance of people and their activities", IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 809–830.
- [27] **Canny, J.**, 1986, "A computational approach to edge detection", IEEE Trans. Patt. Analy. Mach. Intell. 8, 6, 679–698.
- [28] **Horn, B. And Schunk, B.**, 1981, "Determining optical flow", Artific. Intell. 17, 185–203.
- [29] **Lucas, B. D. And Kanade, T.**, 1981, "An iterative image registration technique with an application to stereovision", In International Joint Conference on Artificial Intelligence.
- [30] **Black, M. And Anandan, P.**, 1996, "The robust estimation of multiple motions: Parametric and piecewise smooth flow fields", Comput. Vision Image Understand. 63, 1, 75–104.
- [31] **Şentürk T.**, 2008, "Hareketli kamera ile gerçek zamanlı hareketli nesne takibi", Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi
- [32] **Cremers, D. And Schnorr, C.**, 2003, "Statistical shape knowledge in variational motion segmentation", I.Srael Nent. Cap. J. 21, 77–86.
- [33] **Lowe, D.**, 2004, "Distinctive image features from scale-invariant keypoints.", Int. J. Comput. Vision 60, 2, 91–110.
- [34] **Mikolajczyk, K. And Schmid, C.**, 2003, "A performance evaluation of local descriptors", In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1615–1630.
- [35] **Wren, C., Azarbayejani, A., And Pentland, A.**, 1997, "Pfinder: Real-time tracking of the human body", IEEE Trans. Patt. Analy. Mach. Intell. 19, 7, 780–785.
- [36] **Stauffer, C. And Grimson, W.**, 2000, "Learning patterns of activity using real time tracking", IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 747–767.
- [37] **Elgammal, A., Harwood, D., And Davis, L.**, 2000, "Non-parametric model for background subtraction", In European Conference on Computer Vision (ECCV), 751–767.

- [38] **Rittscher, J., Kato, J., Joga, S., And Blake, A.,** 2000, “A probabilistic background model for tracking”, In European Conference on Computer Vision (ECCV). Vol. 2. 336–350.
- [39] **Stenger, B., Ramesh, V., Paragios, N., Coetzee, F., And Buhmann, J.,** 2001, “Topology free hidden markovmodels: Application to background modeling”, In IEEE International Conference on Computer Vision (ICCV). 294–301.
- [40] **Monnet, A., Mittal, A., Paragios, N., And Ramesh, V.,** 2003, “Background modeling and subtraction of dynamic scenes” In IEEE International Conference on Computer Vision (ICCV). 1305–1312.
- [41] **Zhong, J. And Sclaroff, S.,** 2003, “Segmenting foreground objects from a dynamic textured background via a robust kalman filter”, In IEEE International Conference on Computer Vision (ICCV). 44–50.
- [42] **Rajvir K., Sonit S.,** 2014, “Background Modelling, Detection and Tracking of Human in Video Surveillance System”, International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14).
- [43] **Collins, R., Lipton, A., Fujiyoshi, H., And Kanade, T.,** 2001, “Algorithms for cooperative multisensor surveillance.”, Proceedings of IEEE 89, 10, 1456–1477.
- [44] **Shi, J. And Malik, J.,** 2000, “Normalized cuts and image segmentation”, IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 888–905.
- [45] **Comaniciu, D. And Meer, P.,** 2002, “Mean shift: A robust approach toward feature space analysis”, IEEE Trans. Patt. Analy. Mach. Intell. 24, 5, 603–619.
- [46] **Viola, P., Jones, M., And Snow, D.,** 2003, “Detecting pedestrians using patterns of motion and appearance”, In IEEE International Conference on Computer Vision (ICCV). 734–741.
- [47] **Papageorgiou, C., Oren, M., And Poggio, T.,** 1998, “A general framework for object detection”, In IEEE International Conference on Computer Vision (ICCV). 555–562.
- [48] **1. Forsyth, D. A., Arikan, O., Ikemoto, L., O’Brien, J., and Ramanan, D.,** 2005, “Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis”, Computer Graphics and Vision, 1(2/3):77–254.
- [49] **Sheskin, D.J.,** 2003, “Handbook of Parametric and Nonparametric Statistical Procedures”, Chapman & Hall/CRC.

- [50] **Berbeco, R. I., Mostafavi, H., Sharp, G. C. and Jiang, S. B.**, 2005, “Towards fluoroscopic respiratory gating for lung tumours without radiopaque markers”, *Phys. Med. Biol.* 50 4481–90.
- [51] **Dong, Y.; DeSouza, G.N.**, 2009, “A new hierarchical particle filtering for markerless human motion capture”, *Computational Intelligence for Visual Intelligence, IEEE Workshop on*, Page(s):14 – 21.
- [52] **Salari, V. and Sethi, I. K.**, 1990, “Feature point correspondence in the presence of occlusion”, *IEEE Trans.Patt. Analy. Mach. Intell.* 12(1):87–91.
- [53] **Xue J., Zheng N., Geng J., Zhong X.**, 2008, “Tracking multiple visual targets via particle-based belief propagation”, *IEEE Trans Syst Man Cybern B Cybern.* 38(1):196-209.
- [54] **Smal, K., Draegestein, N. Galjart, W. Niessen, E. Meijering**, 2008, “Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis”, *IEEE Transactions on Medical Imaging*, 27(6):789-804.
- [55] **Zhao, G., Zhang, S., Chen, X., Wang, C.**, 2007, “Efficient Illumination Insensitive Object Tracking by Normalized Gradient Matching”, *IEEE Signal Processing Letters*, 14(12): 944 – 947.
- [56] **Chang, W. Y., Chen, C.S., Jian, Y.D.**, 2008, “Visual Tracking in High-Dimensional State Space by Appearance-Guided Particle Filtering”, 17(7):1154-67.
- [57] **Leven, W. F., Lanterman, A.D.**, 2009, “Unscented Kalman Filters for Multiple Target Tracking With Symmetric Measurement Equations”, *Proceedings of SPIE, the International Society for Optical Engineering*, 5810:56-67.
- [58] **Oh, S., Russell, S., and Sastry, S. S.**, 2009, “Markov chain Monte Carlo data association for multi-target tracking”, *IEEE Transactions on Automatic Control*, 54(3), 481–497.
- [59] **Yilmaz, A.**, 2007, “Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection”, *Computer Vision and Pattern Recognition*, Page(s):1 – 6.
- [60] **Huang A., Hou Z., Yu W., Liu X.**, 2014, “A Visual Object Tracking Method Based on Improved Bhattacharyya Coefficient and Model Update Strategy”, *Fifth International Conference on Intelligent Systems Design and Engineering Applications*.

- [61] **Fan, Z. , Wu, Y., Yang, M.**, 2005, “Multiple Collaborative Kernel Tracking”, Computer Vision and Pattern Recognition, Vol 2, Pages: 502- 509.
- [62] **Lee, S., Kang, J., Shin, J., Paik, J.**, 2007, “Hierarchical active shape model with motion prediction for real-time tracking of non-rigid objects”.
- [63] **Cordea, M. D., Petriu, E. M., Petriu, D. C.**, 2008, “Three-Dimensional Head Tracking and Facial Expression Recovery Using an Anthropometric Muscle-Based Active Appearance Model”, IEEE Transactions on Instrumentation and Measurement, 57(8): 1578 – 1588.
- [64] **Munder, S.; Schnorr, C.; Gavrilu, D.M.**, 2008, “Pedestrian Detection and Tracking Using a Mixture of View-Based Shape–Texture Models”, IEEE Transactions on Intelligent Transportation Systems, 9(2): 333 – 343.
- [65] **Nascimento, J.C.; Marques, J.S.**, 2007, “Robust Shape Tracking With Multiple Models in Ultrasound Images”, IEEE Transactions on Image Processing, 17(3): 392 – 406.
- [66] **Zhu, G., Zhang, S., Chen, X., Wang, C.**, 2007, “Efficient Illumination Insensitive Object Tracking by Normalized Gradient Matching”, IEEE Signal Processing Letters, 14(12): 944 – 947.
- [67] **Schweitzer, H., Bell, J. W., And Wu, F.**, 2002, “Very fast template matching”, In European Conference on Computer Vision (ECCV). 358–372.
- [68] **Fieguth, P. And Terzopoulos, D.**, 1997, “Color-based tracking of heads and other mobile objects at video frame rates”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21–27.
- [69] **Jepson, A., Fleet, D., And Elmaraghi, T.**, 2003, “Robust online appearance models for visual tracking”, IEEE Trans. Patt. Analy. Mach. Intell. 25, 10, 1296–1311.
- [70] **Raykar, V. C., Duraiswami, R.**, 2006, “Fast optimal bandwidth selection for kernel density estimation”, Proceedings of the sixth SIAM International Conference on Data Mining, pp. 524-528.
- [71] **Bertalmio, M., Sapiro, G., And Randall, G.**, 2000, “Morphing active contours”, IEEE Trans. Patt. Analy. Mach. Intell. 22, 7, 733–737.
- [72] **URL-1**, <https://www.tai.com.tr/tr/bolum/iha-sistemleri>, (03.02.2015)
- [73] **URL-2**, http://www.vestelsavunma.com/?page_id=20 (03.02.2015)

- [74] **Selim, E., Uyar, E., Alci, M.**, 2013 “Quadrocopterin Matematiksel Modeli ve Kontrolü”, Otomatik Kontrol Ulusal Toplantısı, TOK.
- [75] **URL-3**, http://www.hse-uav.com/fixed_wing_vs_rotary_wing_uav.htm, (03.02.2015).
- [76] **Lee G., Jeong D. Y. and Khoi N. D.**, 2011, “Attitude Control System Design For A Quadrotor Flying Robot”, 8th International Conference on Taesam Kang Ubiquitous Robots and Ambient Intelligence (URAI), Seoul, South Korea, pp. 74-78
- [77] **A. Nemra and N. Aouf**, 2010, “Robust cooperative UAV Visual SLAM,” Cybernetic Intelligent Systems (CIS), 2010 IEEE 9th International Conference on, pp. 1-6.
- [78] **F. Caballero, L. Merino, J. Ferruz, and A. Ollero**, 2009, “Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs”, Journal of Intelligent and Robotic Systems, vol. 54, no. 1-3, pp. 137-161.
- [79] **Arjomandi, M., Agostino, S., Mammone, M., Nelson, M., & Zhou, T.**, 2003, “Classification of Unmanned Aerial Vehicles”, The University of Adelaide , Australia.
- [80] **Aydoğdu, Ö. ve Bayer, M.**, 2008, “PIC Tabanlı Fırçasız DC Motor Sürücüsü Tasarımı”, Elektrik – Elektronik - Bilgisayar Mühendisliği Sempozyumu, Bursa.
- [81] **Real, J. C. G., Sanchez, E. V., Gil, J. G.**, 2010, “Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application”, Trends, Sensors, 10, 6901-6947.
- [82] “AR.Drone Developer Guide SDK 2.0.”
- [83] **Dijkshoorn, N.**, 2012, “Simultaneous localization and mapping with the AR.Drone”, Doktora Tezi, Amsterdam Üniversitesi.
- [84] **Bristeau, P. J., Callou, François, Vissière, David, Petit, Nicolas**, 2011, “The Navigation and Control technology inside the AR.Drone micro UAV”, 18th IFAC World Congress.
- [85] **URL-4**, <http://ardrone2.parrot.com/repair-your-ardrone/> (03.02.2015)
- [86] **Gary R. Bradski**, 1998, “Computer Vision Face Tracking For Use in a Perceptual User Interface”, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation.

ÖZGEÇMİŞ

1987 yılında Çaycuma’da doğdu. Lise öğrenimini Kovancılar Y.D.A. Lisesi’nde tamamladıktan sonra lisansa 2006 yılında Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümünü kazandı ve 2010 yılında mezun oldu. 1 Eylül 2010 tarihinde Bingöl Üniversitesi’nde ÖYP Araştırma Görevlisi olarak çalışmaya başladı. 2012 yılında Fırat Üniversitesi’nde yüksek lisansa başladı ve aynı kurumda ÖYP Araştırma Görevlisi olarak çalışmaya devam etmektedir.