



YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

hareketli kamera ile gerçek zaman..

Yüksek Lisans Tezi

tülin şentürk

YILDIZ TEKNİK ÜNİVERSİTESİ
KÜTÜPHANE VE DOKÜMANTASYON
DAİRE BAŞKANLIĞI

Yer No (DDC) : R 368-143

Kayıt No : 4179

Geldiği Yer : Fen Bil. Erit.

Tarih : 27/10/08

Fiyat : 2,85-

Fatura No :

Ayniyat No : 62

Ek :

HAREKETLİ KAMERA İLE GERÇEK ZAMANLI HAREKETLİ NESNE TAKİBİ

Tülin ŞENTÜRK

FBE Bilgisayar Mühendisliği Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd.Doç. Dr. M. Elif Karşılıgil

E. Karşılıgil

Prof. Dr. A. Cenk Samur
Doç. Dr. Cem Ünsalan

İSTANBUL, 2008

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
1.1 Literatürde Nesne Takibi	1
2. NESNE TAKİBİ	5
2.1 Hareketli Görüntülerde Takip Edilecek Nesnenin Yerinin Bulunması	5
2.1.1 Arka Plan Farkı Alınarak Nesne Bulma	5
2.1.2 Sıralı Görüntülerin Farkının Alınmasıyla Nesne Bulma	6
2.1.3 Bölütleme ile Nesne Bulma	6
2.1.4 CamShift Yöntemi ile Nesne Bulma	8
2.2 Nesne Takibinde Önemli Rol Oynayan Kriterler	8
2.2.1 Nesne Gösterimleri	8
2.3 Özellik Seçimi	10
2.4 Nesne Takibi Metotları	13
2.4.1 Kalman Filtresi ile Nesne Takibi	15
2.4.2 CONDENSATION Yöntemi ile Nesne Takibi	16
3. HAREKETLİ BİR NESNENİN HAREKETLİ KAMERA İLE GERÇEK ZAMANLI TAKİBİ	18
3.1 Görüntü Alımı	18
3.2 Renk İndirgeme	18
3.3 Nesne Bulma ve Takibi	19
3.3.1 Mean-Shift ile Nesne Bulma	20
3.3.2 Mean-Shift Yöntemi ile Nesne Takibi	24
3.3.3 CamShift Yöntemi ile Nesne Bulma	29
3.3.4 CamShift Yöntemi ile Nesne Takibi	30
4. UYGULAMA VE DENEYSEL SONUÇLAR	36
4.1 Nesnenin Kameraya Olan Uzaklığının Değişmesi Durumu	36
4.1.1 Nesne Takibi Sırasında Uygun Pencere Boyutunun Seçilmesi	36
4.2 Nesnenin Bir Kısımının veya Tamamının Kaybolması Durumu	38
4.3 Sabit Kamera İle Hareketli Görüntü Takibi	39
4.4 Hareketli Kamera İle Hareketli Görüntü Takibi	45

4.5	Hızlı Bir Hedef Nesnenin Takibi.....	47
4.6	Birden Fazla Nesnenin Takibi	49
5.	SONUÇ.....	51
KAYNAKLAR.....		53
ÖZGEÇMİŞ.....		55

SİMGE LİSTESİ

T	Eşik seviyesi değeri
q	Hedef modelin olasılık yoğunluk fonksiyonu
p	Aday modelin olasılık yoğunluk fonksiyonu
x_i	Piksel koordinatları
$k(x)$	Çekirdek fonksiyonu
δ	Kronecker delta fonksiyonu
ρ	Bhattacharyya uzaklığı
$M00$	Sıfırıncı moment
$M10$	Birinci dereceden moment
$M01$	Birinci dereceden moment
$M11$	İkinci dereceden moment
$M20$	İkinci dereceden moment
$M02$	İkinci dereceden moment
(x_c, y_c)	Hedef nesnenin merkez koordinatları
l	Hedef nesnenin yüksekliği
w	Hedef nesnenin genişliği
s	Arama alanının boyutu

KISALTMA LİSTESİ

FPGA	Field Programmable Gate Array	2
ASIC	Application Specific Integrated Circuit	3
RGB	Red Green Blue	7
HSV	Hue Saturation Value	8
JPDAF	Joint Probabilistic Data Association Filter	9
PMHT	Probabilistic Multiple Hypothesis Tracking	10
KLT	Kanade Lucas Tomasi Feature Tracker	10
SVM	Support Vector Machine	11
DPI	Dot Per Inch	11
IEEE	Institute of Electrical and Electronics Engineers	12
Sekil 2.8	(a) Orijinal resim (b) Doku özellikleri	12
Sekil 2.9	Nokta takibi	14
Sekil 2.10	Dikdörtgeni piksel parametrik dördüncü	14
Sekil 2.11	Çevre görsel oluşturma örnekleri	15
Sekil 3.1	Sistem blok diyagramı	18
Sekil 3.2	Renk uzayının indirgenmesi (a) 256 renk derinliğine sahip RGB uzayındaki görüntü (b) 16 renk derinliğine sahip RGB uzayındaki görüntü	19
Sekil 3.3	Mean-Shift ile nesne yatağı bulunması anı yarı kodlama	21
Sekil 3.4	Bluetecheryye uzaklığı	23
Sekil 3.5	Hedef nesnenin görüntüsü	24
Sekil 3.6	Futbol topunun Mean-Shift yöntemi ile bulunması	24
Sekil 3.7	Hedef nesnenin olasılık dağılımının en yakın olguya merkezi bulunması	25
Sekil 3.8	Mean-Shift yöntemi ile nokta takibine ait akış diyagramı	27
Sekil 3.9	Futbol topunun Mean-Shift yöntemi ile takibi	29
Sekil 3.10	CamShift yöntemi ile nesne takibi anı akış diyagramı	34
Sekil 3.11	CamShift yöntemi ile nesne takibi anı kodlama	35
Sekil 4.1	Uygun pencere boyutunun belirlenmesi için değerlendirilen alanların gösterimi	37
Sekil 4.2	Pencere boyutunun genişletilmesi	38
Sekil 4.3	Hedef nesnenin robotun gözetim alanı içerisinde bulunması anı akış diyagramı	39
Sekil 4.4	Hedef nesne	39
Sekil 4.5	Futbol topunun uzaklığı nesnenin takibi	40
Sekil 4.6	Kameralara yaklaşıp uzaklaşma takibi	40
Sekil 4.7	Hedef nesnenin görüntüsü	41
Sekil 4.8	Howling topunun takibi	41

ŞEKİL LİSTESİ

Şekil 1.1 Masa tenisi topunun takip edilmesi	2
Şekil 2.1 Arka plan farkının alınmasıyla elde edilen sonuç	4
Şekil 2.2 Ön plandaki nesnelerin bölütleme ile bulunması	7
Şekil 2.3 Nesnelerin değişik açılardan çekilmiş pozları	8
Şekil 2.4 Nesne gösterimleri. (a) Merkez nokta, (b) çoklu nokta, (c) dikdörtgen, (d) elips, (e) parça tabanlı çoklu elips, (f) iskelet, (g) çevre çizgisinde kontrol noktaları, (h) çevre çizgisi, (i) nesne silueti	10
Şekil 2.5 Tek boyutlu uzaya indirgenmiş renk olasılık dağılımı	11
Şekil 2.6 (a) Orjinal resim (b) Canny kenar bulma uygulandığında elde edilen sonuç	11
Şekil 2.7 Optik akış (a) 1. görüntü (b) yön doğrultuları (c) 2. Görüntü	12
Şekil 2.8 (a) Orjinal resim (b) Doku özellikleri	12
Şekil 2.9 Nokta takibi	14
Şekil 2.10 Dikdörtgen şeklin parametrik dönüşümü	14
Şekil 2.11 Çevre çizgisi oluşturma örnekleri	15
Şekil 3.1 Sistem blok diyagramı	18
Şekil 3.2 Renk uzayının indirgenmesi (a) 256 renk derinliğine sahip RGB uzayındaki görüntü (b) 16 renk derinliğine sahip RGB uzayındaki görüntü	19
Şekil 3.3 Mean-Shift ile nesne yerinin bulunmasına ait yarı kodlama	21
Şekil 3.4 Bhattacharyya uzaklığı	23
Şekil 3.5 Hedef nesnenin görüntüsü	24
Şekil 3.6 Futbol topunun Mean-Shift yöntemi ile bulunması	24
Şekil 3.7 Hedef nesnenin olasılık dağılımının en yoğun olduğu merkezin bulunması	25
Şekil 3.8 Mean-Shift yöntemi ile nesne takibine ait akış diyagramı	27
Şekil 3.9 Futbol topunun Mean-Shift yöntemi ile takibi	29
Şekil 3.10 CamShift yöntemi ile nesne takibine ait akış diyagramı	34
Şekil 3.11 CamShift yöntemi ile nesne takibine yarı kodlama	35
Şekil 4.1 Uygun pencere boyutunun belirlenmesi için değerlendirilen alanların gösterimi	37
Şekil 4.2 Pencere boyutunun güncellenmesi	38
Şekil 4.3 Hedef nesnenin robotun görüş açısından çıkması durumuna ait akış diyagramı	39
Şekil 4.4 Hedef nesne	39
Şekil 4.5 Kameradan uzaklaşan nesnenin takibi	40
Şekil 4.6 Kameraya yaklaşan nesnenin takibi	40
Şekil 4.7 Hedef nesnenin görüntüsü	41
Şekil 4.8 Bowling topunun takibi	41

Şekil 4.9 Futbol topunun takibi	42
Şekil 4.10 Hedef nesnenin görüntüsü	43
Şekil 4.11 Hareketli robotun takibi	43
Şekil 4.12 Hedef nesnenin görüntüsü	44
Şekil 4.13 Hareketli nesnenin sahneye sonradan girmesi durumunda takibi	45
Şekil 4.14 Hedef nesnenin görüntüsü	46
Şekil 4.15 Hareketli kamera ile hedef nesnenin takibi	46
Şekil 4.16 Hızlı hedef nesnenin takibi.....	47
Şekil 4.17 Hızlı hedef nesnenin takibi.....	48
Şekil 4.18 Hareketli iki nesnenin takibi.....	49

ÇİZELGE LİSTESİ

Çizelge 3.1 Nesne takibi kategorileri	13
Çizelge 4.1 Nesne takibi sonuçları	50

ÖNSÖZ

Bu tez kapsamında hareketli kamera ile hareketli bir objenin gerçek zamanlı takibi gerçekleştirilmiştir. Bu tarz bir uygulamanın ülkemizde yapılmış çalışmalar içerisinde azlığı ve yetersizliği bu tez konusunun seçilmesini teşvik etmiştir.

Tezimin araştırma ve uygulama aşamalarında konu hakkındaki bilgisini bana aktaran ve çalışmalarına destek veren değerli hocam ve tez danışmanım Yrd. Doç. Dr. M. Elif Karslıgil'e, uygulama ve çalışma ortamım konusunda bana yardımcı olan Yrd. Doç. Dr. A. Gökhan Yavuz'a, bölümdeki her çalışmamda desteğini her zaman hissettiğim ve bilgi birikiminin çekirdeğini oluşturan bölümümüzün kurucusu Sayın Prof. M. Yahya Karslıgil'e, Bilgisayar Mühendisliği Bölümündeki bütün hocalarıma ve arkadaşlarıma teşekkürlerimi sunarım.

Hayatım boyunca bana her konuda manevi ve maddi destek veren, bugünlere gelmemi sağlayan ve ileride de yanımda olacak sevgili anneme, babama ve ablama çok teşekkür ederim.

ÖZET

Nesne takibi, askeri robot uygulamalarında; insan yaşamına elverişli olmayan ortamlarda uzaktan kontrollü olarak çalışan uygulamalarda; şüpheli şahısların tespiti, trafik kontrolü ve düzenlenmesi, insansız gözetim, insansız araç navigasyonu gibi uygulamalarda önemli çalışma alanlarından biridir.

Bu tez çalışmasında, dinamik ortamlarda hareketli kamera ile hedef takibi yapan gerçek zamanlı bir sistem tasarımı gerçekleştirilmiştir. Hareketli görüntü alımı, hareketli bir platform üzerinde bulunan doksan derece görme açısına sahip, renkli bir video kamera ile yapılmaktadır. Sistem; görüntü alımı, hedef nesnenin tanınması ve hedef nesnenin takibi ana modüllerinden oluşmaktadır.

Takip edilmesi istenen hareketli nesne, görüntü bilgisi önceden sisteme tanıtılmış herhangi bir nesne olabilmektedir. Takip edilecek nesnenin yeri, renk dağılımı bilgisi özellik olarak kullanılarak CamShift (Continuously Adaptive Mean-Shift - Sürekli Uyarlamalı Ortalama Değer Kayması) algoritması ile bulunmaktadır. Takip edilmesi istenen hareketli nesne bulunana kadar platform hareket etmeden ortam gözlenmektedir. Hedef nesne kameranın görüş açısına girdiğinde nesnenin görüntü üzerindeki yeri bulunmakta ve nesne takibi işlemi başlamaktadır.

Gerçek zamanlı hareketli nesne takibi CamShift yöntemi kullanılarak yapılmıştır. Takip edilmesi hedeflenen nesnenin bir önceki görüntüdeki yerinden ve nesnenin renk olasılık yoğunluk dağılımından yararlanılarak hedef nesnenin o anki görüntüdeki yeri bulunmaktadır. Bu amaçla her bir görüntü üzerinde, nesnenin bir önceki görüntüde bulunduğu yere göre bir arama alanı belirlenmekte ve bu arama alanındaki piksellerin renklerinin olasılığından ve yer bilgisinden yararlanılarak moment bilgileri hesaplanmaktadır. Bu moment bilgileri kullanılarak hedef nesnenin renk dağılımının yoğun olduğu bölge itaretif olarak bulunmaktadır. Bu bölgenin merkezi hedef nesnenin merkezi olarak kabul edilmektedir. Yine moment bilgisinden yararlanılarak hedef nesnenin görüntüde kapladığı alan bulunmakta ve nesnenin bir sonraki görüntüde aranacağı alanın büyüklüğü güncellenmektedir. Bu sayede takip edilen nesnenin kameraya yaklaşmasına veya uzaklaşmasına bağlı olarak boyutunun değişmesine rağmen nesne başarıyla takip edilebilmektedir.

Anahtar kelimeler: Nesne takibi, Nesne bulma, CamShift, Mean-Shift, Bhattacharyya uzaklığı

ABSTRACT

Object tracking applications are among the most important research areas of especially military applications, fire-fighting, police and civil defense, remote controlled operational robotic applications working in environments not suitable for human life and systems like determining suspicious people, traffic control and regulation, surveillance without human, vehicle navigation without human and warehouse management.

In this project, a real-time system is performed that tracks a moving object using an active camera in dynamic environments. Images are taken from a mobile and colored camera with 90 degrees field of view which has been put on a mobile platform. System contains three main modules: image capturing, target object detection and moving target object tracking.

In the system, the target object can be any object which has been represented to the system with its image. The location of the target object is found by CamShift (Continuously Adaptive Mean-Shift) algorithm using color probability density function as feature. The system monitors the environment and the platform is stable until the target object appears in the scene. When the moving target object appears in the scene the location of the object is estimated and then object tracking starts.

In this system CamShift method is performed for real-time moving object tracking. The location of the moving target object in the current frame is estimated by using the location of the moving target object in the previous frame and color probability density function of the target object. In each frame, a search area is defined according to the location of the object in the previous frame. Moments are estimated by using the color probabilities of the pixels in this search area and the location of these pixels. The densest region which includes the color probability density function of the target object is found using estimated moments iteratively. The center of this region is assumed as the center of the moving target object. The dimensions of the target object are estimated and the search area in the next frame is updated by using moments. So, the target object is tracked successfully although its dimensions are changed.

Keywords: Object tracking, Object detection, CamShift, Mean-Shift, Bhattacharyya distance

1. GİRİŞ

Günümüz teknolojisinde görüntünün kısa sürede işlenebilir hale gelmesiyle bilgisayarla görme alanında yapılan çalışmalarda insanların günlük ihtiyaçlarını en aza indirmek amaçlanmaktadır.

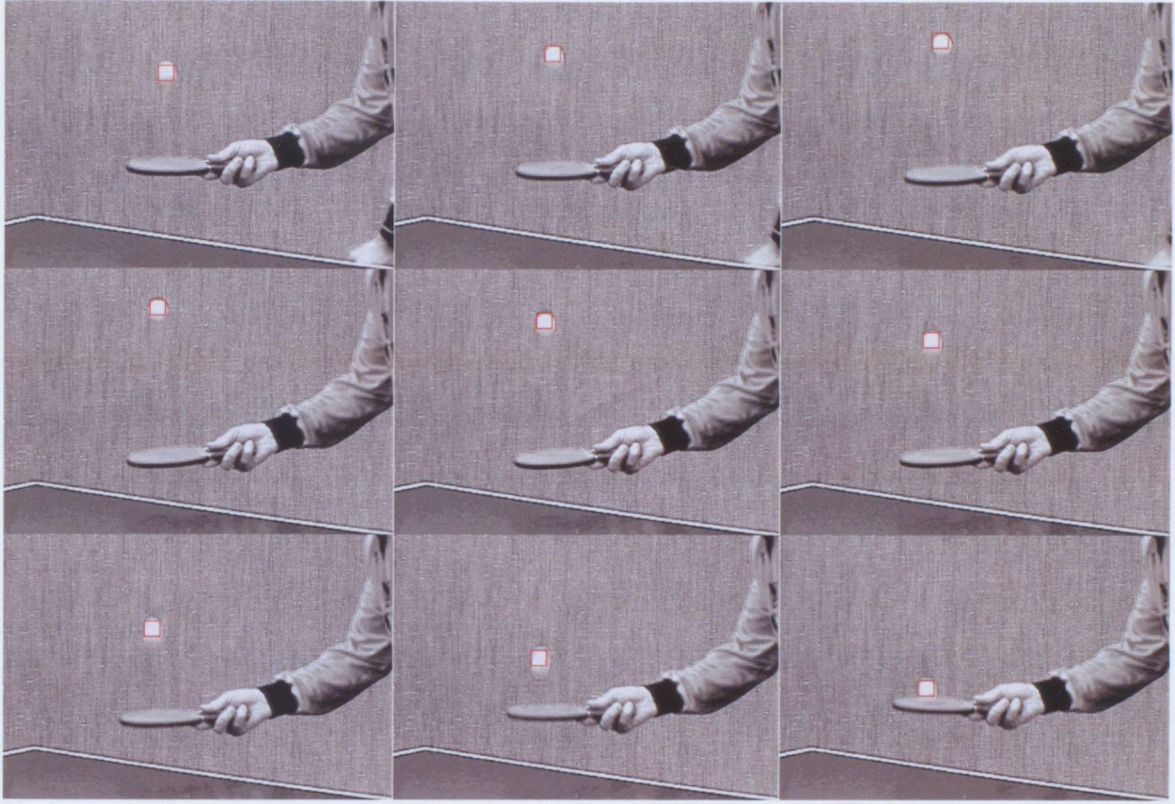
Şüpheli şahısların teşhisi, trafik kontrolü ve düzenlenmesi, insansız gözetim, insansız araç navigasyonu gibi bilgisayarla görme alanının ilgilendiği önemli uygulamalarda nesne takibi kullanılmaktadır. Görüntü bilgilerini değerlendirerek nesne takibi konusunda literatürde çeşitli çalışmalar yapılmış, uygulamanın amacına yönelik olarak kullanılan yöntemlerde iyileştirmeler yapılmıştır. Genel olarak nesne takibi, ortamda hareket eden hedef nesnenin ardışık görüntüler üzerinde yerinin bulunmasıyla hareket doğrultusunun çıkarılması olarak tanımlanabilir (Şekil 1.1).

Nesne takibi yapabilmek için öncelikle hedeflenen nesnenin görüntüdeki yerinin bulunması gerekir. Hedeflenen nesne, herhangi bir hareketli nesne olabileceği gibi uygulamanın amacına göre önceden belirlenmiş bir nesne de olabilir. Eğer hedef nesne önceden belirlenmemişse, öncelikle görüntüde bir hareket olana kadar beklenir, hareket gerçekleştikten sonra hareket eden nesne, hedef olarak belirlenir ve bu nesne takip edilmeye başlanır. Eğer hedef nesne önceden belirlenmişse görüntüde bu nesne aranır ve nesne bulunduktan sonra takip aşamasına geçilir. Takip edilecek nesnenin şekil özelliklerine göre nesneyi temsil edecek bir şekil seçilir. Hedef nesne, kendisini temsil eden şekil içerisindeki özelliklerden yararlanılarak takip edilir. Bu özellikler takip aşamasında önemli rol oynadığı için takip edilecek nesneyi çevreden ayırt edici kılacak şekilde farklı olmalıdır.

Literatürde nesne takibi yapan birçok çalışma yer almaktadır. Bu çalışmalar; nesneyi temsil eden şekle, kullanılan özelliklere, nesnenin hareketinin, görünümünün ve şeklinin nasıl modellendiğine göre farklılık göstermektedir.

1.1 Literatürde Nesne Takibi

Nesne takibinde ve takip aşaması öncesindeki adımlarda kullanılabilecek yöntemler, yapılan çeşitli çalışmalarla incelenmiştir. Bu amaçla yapılmış olan bir çalışmada, uygulamaya bağlı olarak seçilmesi gereken nesne gösterimlerinden, takip için kullanılacak olan özelliklerin seçiminden, nesne bulma yöntemlerinden ve nesne takibi amacıyla kullanılabilecek yöntemlerden bahsedilmiştir (A. Yılmaz, O. Javed, M. Shah, 2006).



Şekil 1.1 Masa tenisi topunun takip edilmesi

Literatürdeki bazı çalışmalar trafik planlama ve düzenleme üzerinedir. Bunlardan bazıları trafik düzenleme uygulamalarının performansını yükseltmeyi amaçlamıştır (Qi Zang, Reinhard Klette, 2003). Bu çalışmada takip edilen nesnenin hareketi Kalman filtresi kullanılarak belirlenmiştir. Geliştirilen sistemde hareketsiz bir kameradan alınan görüntü üzerinde çalışılmaktadır. Çalışılan görüntü üzerindeki arka planın sabit olması sayesinde, hareketli nesneler arka plandan ayrılıp izlenmiştir. Hareketsiz kamera ile çalışan bu tarz uygulamalarda aktif bir nesne takibi söz konusu değildir. Bu sistemler daha ziyade trafik izleme ve düzenleme, kalabalık alanlardaki olağan dışı durumları kontrol etme gibi uygulamalara entegre olarak çalışır.

Bir başka uygulamada, otonom mobil robot üzerine yerleştirilmiş kameradan görüntü alarak, nesne bulma ve takibi gerçekleştiren sistemde hedef nesnenin ve kameranın hareketli olması, hedef nesnenin takibini zorlaştırmıştır (K. K. Kim, S. H. Cho, H. J. Kim ve J. Y. Lee, 2005). Bu problemi aşmak için kamera hareketinin doğrultusu ve hızı kullanılarak takip edilecek nesnenin hareket doğrultusu ve hızı hesaplanmıştır.

Görüntü bölütlemesi, bölütleme sonucu elde edilmiş parçaların özelliklerinin çıkarılması ve şablon eşleme yöntemiyle nesne izleme yapan çalışmalar da bulunmaktadır (T. Morimoto, O. Kiriya, Y. Harada, H. Adachi, T. Koide, H. J. Mattausch, 2005). Geliştirilen modüllerin

karmaşıklık düzeyinin yüksek olması, sistemin gerçek zamanlı uygulamalara adaptasyonunu güçleştirmektedir. Bu sebepten ötürü, aynı çalışma grubu sistemin modüllerini, paralel çalışabilmeleri için FPGA / ASIC yapısını kullanarak yeniden tasarlamışlardır (T. Morimoto, O. Kiriya, Y. Harada, H. Adachi, T. Koide, H. J. Mattausch, 2005).

Literatürde, gerçek zamanlı video görüntüleri üzerinde değil, önceden kaydedilmiş görüntüler üzerinde nesne takibi yapan çalışmalar da mevcuttur. Bu çalışmalardan birinde, takip edilerek bulunan hareketli nesnelerin silinmesi amaçlanmıştır (S. Park, C. Park ve I. Lee, 2005). Bu çalışmada kullanılan kameranın hareketli olmasından dolayı arka arkaya gelen görüntüler birleştirilerek bir mozaik oluşturulmuş, böylece kamera hareketi elimine edilmiştir.

Uçan bir platformdan alınan kayıtlı video verisi üzerinde nesne takibi yapan çalışmalar da vardır. Bu çalışmalardan birinde, hareketli nesneleri takip etmek için iki aşamalı bir yaklaşım sunulmuştur (Isaac Cohen ve Gerard Medioni, 1998). Öncelikle gözlemlene platformunun hareketinin neden olduğu görüntü akışı bulunmuş ve elimine edilmiş, daha sonra iki boyutlu şablon kullanılarak hareketli nesne izlenmiştir.

Nesne takibi amacıyla kullanılan bir yöntem de kenar takibi algoritmasıdır. Hareketli kameradan alınan video görüntülerini kullanan sistemde, doku ve renk özelliklerinin olasılık dağılım fonksiyonlarına dayanan Bayes çıkarımı kullanılmıştır (A. Yılmaz, X. Li ve M. Shah, 2004).

Esnek yapılı nesnelerin takibi üzerine de çalışmalar yapılmıştır. Bu çalışmalardan birinde; gerçek zamanlı olarak hareketli kameradan alınan görüntüler üzerinde esnek yapılı nesnelerin takibi gerçekleştirilmiştir (D. Comaniciu, V. Ramesh ve P. Meer, 2000). Algoritma, takip edilecek nesnenin ardışık her bir görüntüde Ortalama Değer Kayması (Mean-Shift) iterasyonları ile yerinin belirlenmesine dayanır. Diğer bir çalışmada esnek yapılı hedef nesnenin takibi, hedef nesneyi temsil eden bir geometrik şekil ve bu şekil içerisindeki özellik dağılımı kullanılarak gerçekleştirilmiştir (D. Comaniciu, V. Ramesh ve P. Meer, 2003). Bu çalışmada, Bhattacharyya uzaklığı ile benzerlik hesaplanmış, optimizasyon için ise Mean-Shift algoritması kullanılmıştır.

Bazı çalışmalarda nesne takibinde karşılaşılan problemler üzerinde durulmuş ve bu problemlere çözümler üretilmeye çalışılmıştır. Bu çalışmalardan birinde, görüntü özelliklerinin değişimine ve hedef nesnenin tamamının görüntülenememesine rağmen, hedef nesne, enerji fonksiyonu kullanılarak takip edilmiştir (A. Yılmaz, X. Li ve M. Shah, 2004).

Ulusal literatürde nesne takibi üzerinde yapılmış bir çalışmada, görsel güvenlik sistemi için gerekli olan temel parçaların tanımlanması ve geliştirilmesi amacıyla değişken ortam

şartlarını öğrenme kapasitesine sahip, ayrıca gölgeleri de nesne maskelerinden ayırabilen bir arka plan modelleme yöntemi tanımlanmaktadır (Burkay Birant Örten, 2005).

Diğer bir ulusal çalışmada, nesne bulma, sınıflandırma ve takip etme yeteneklerine sahip bir akıllı video gözetim sistemi sunulmuştur (Yiğithan Dedeoğlu, 2004). Sistem sabit bir kameradan elde edilen renkli ve renksiz görüntüler üzerinde çalışabilmektedir.

Tez kapsamında gerçekleştirilen sistem, hareketli bir platform üzerine yerleştirilmiş hareketli kamera ile alınan görüntüleri gerçek zamanlı olarak değerlendirerek, hareketli bir nesneyi değişken ortamlarda takip etmektedir.

Sistemde ilk olarak renkli görüntü çekimi yapan bir kameradan görüntü alımı yapılmaktadır. Daha sonra renk derinliği indirgenen görüntü üzerinde hedef nesne aranmaktadır. Hedef nesne robotun görüş açısına girene kadar robot hedef nesneyi beklemekte ve aynı zamanda ortamı gözlemektedir. Hedef nesne robotun görüş açısına girdiği zaman nesne bulma algoritması hedef nesnenin renk özelliklerini kullanarak hedef nesnenin yerini belirlemekte ve bu yer bilgisini nesne takibi aşamasına aktarmaktadır. Nesne takibi aşaması bu yer bilgisini ve hedef nesnenin renk özelliklerini kullanarak sıradaki görüntüde hedef nesnenin yeni yerini bulmakta ve bu bilgiyi nesne takibi aşamasının bir sonraki adımına aktarmaktadır. Diğer adımlar da ardışık görüntüler üzerinde çalışarak hedef nesnenin bir önceki görüntüde bulunduğu bölgenin civarında hedef nesneyi aramakta ve nesnenin bulunduğu yeri bir sonraki adıma aktarmaktadır.

Tez kitabının ikinci bölümünde nesne takibi gerçekleştirebilmek için uygulanması gereken adımlardan ve bu adımlarda kullanılabilecek yöntemlerden bahsedilmektedir. Üçüncü bölümde tez kapsamında gerçekleştirilen sistemden, bu sistemin adımlarından bahsedilmekte ve kullanılan yöntemler açıklanmaktadır. Dördüncü bölümde sistemin değişik senaryolarla test edilmiş başarısı sunularak sistemin gerçeklenmesinde karşılaşılan problemlerden, uç noktalardan, senaryolardan ve bu durumlara karşı üretilen çözümlerden bahsedilmektedir. Son bölümde ise elde edilen sonuçlar değerlendirilmektedir.

2. NESNE TAKİBİ

Nesne izlemenin amacı, ardışık görüntüler üzerinde hedef nesnenin yerini belirleyerek hedef nesnenin hareket doğrultusunu çıkarmaktır.

2.1 Hareketli Görüntülerde Takip Edilecek Nesnenin Yerinin Bulunması

Nesne takibi aşamasına geçilmeden önce takip edilmesi amaçlanan nesnenin görüntü üzerindeki yeri bulunur. Literatürde nesne bulma amaçlı birçok çalışma yapılmış ve bu doğrultuda değişik yöntemler geliştirilmiştir. Nesne bulmak için kullanılacak yöntem, uygulamanın amacına göre değişir. Aşağıda literatürde yer alan birkaç temel nesne bulma algoritması incelenmiştir.

2.1.1 Arka Plan Farkı Alınarak Nesne Bulma

Arka plan farkı alınarak hareketli nesnenin bulunması, görüntü alımı yapan kameranın sabit, hedef nesnenin hareketli olduğu durumlarda kullanılan bir yöntemdir. İlk olarak, sistem çalışmaya başlamadan önce arka plan görüntüsü belirlenir. Daha sonra gelen her bir görüntü ile arka plan görüntüsünün farkı alınır (A. M. McIvor, 2000) (Şekil 2.1). Başka bir deyişle arka plan görüntüsü ile sıradaki görüntüde aynı koordinatlara denk gelen piksellerin renk değerleri birbirinden çıkarılır. Elde edilen farkın mutlak değeri daha önceden belirlenmiş olan bir eşik seviyesi değeri (T) ile kıyaslanır (Eşitlik 2.1).

$$| \text{görüntü}_i(x,y) - \text{arka plan}(x,y) | > T \quad (2.1)$$

Eğer fark değeri bu eşik seviyesi değerinin üzerindeyse ilgili koordinatlarda bir değişiklik olduğuna karar verilir. Fakat yalnızca bu kriterle o bölgede hareket olduğu kanısına varılamaz. Çünkü ışık farklılığı sebebiyle bazı pikseller bu kriteri sağlayabilir. Amaç hareketli bir nesneyi bulmak ise değişim olan piksellerin kapalı bir alan oluşturması ve bu alanın belirlenen bir eşik seviyesi değerinden büyük olması gerekir. Birbirine bağlı bölgeleri ayrı ayrı belirlemek amacıyla Bağlı Bileşen Analizi (Connected Component Analysis) (Hanan Samet ve Markku Tamminen, 1988) yöntemi kullanılabilir. Bu yöntem uygulandıktan sonra ışık farklılığı sebebiyle değişiklik gösteren pikseller elenmiş, hareket eden nesneler başka bir deyişle arka planda yer almayan nesneler bulunmuş olur.



Şekil 2.1 Arka plan farkının alınmasıyla elde edilen sonuç (Gregor McEwan, Pavitra Wickramasinghe)

2.1.2 Sıralı Görüntülerin Farkının Alınmasıyla Nesne Bulma

Bir önceki yöntem gibi sıralı görüntülerin farkının alınmasıyla hareketli nesne bulan yöntemin de görüntü alımı yapan kameranın sabit, hedef nesnenin hareketli olduğu durumlarda kullanımı uygundur. Sıralı görüntülerin farkının alınmasıyla nesne bulma, video akışında arka arkaya gelen görüntüdeki piksellerin renk değerlerinin farkını alarak bu farkı bir eşik seviyesi ile karşılaştıran bir yöntemdir (A. M. McIvor, 2000) (Eşitlik 2.2).

$$| \text{görüntü}_i(x,y) - \text{görüntü}_{i-1}(x,y) | > T \quad (2.2)$$

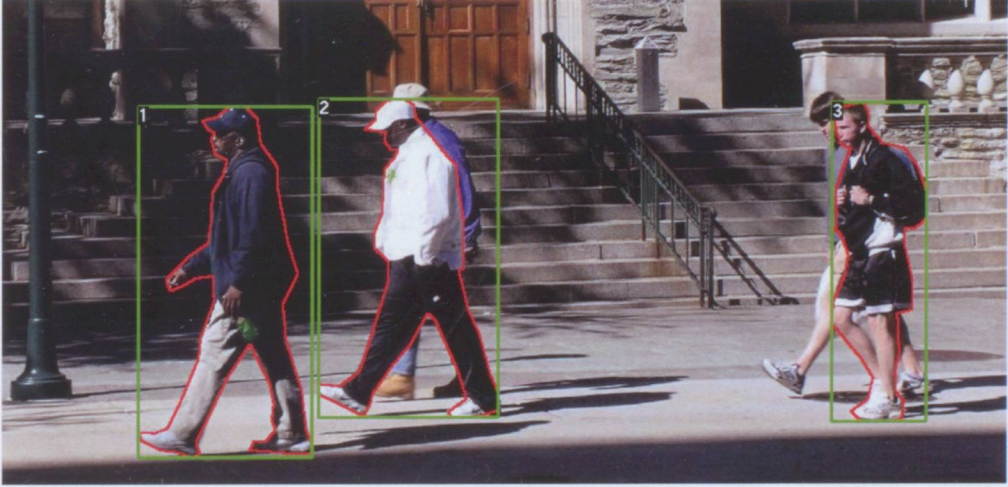
Bu fark sayesinde görüntüdeki yer değişimleri ortaya çıkmaktadır. Bu yöntemin diğer aşamaları arka plan farkı ile nesne bulma yöntemindeki gibidir. Bu yöntemin diğerinden farkı, diğer yöntemde arka planda yer almayan bütün nesneler bulunurken bu yöntemde yalnızca hareketli bölgelerin, yani bir önceki görüntüye göre farklılık gösteren bölgelerin tespit ediliyor olmasıdır.

2.1.3 Bölütleme ile Nesne Bulma

Görüntü bölütleme algoritmaları, görüntüdeki benzer renkli bölgeleri gruplamaktadır. Nesne bulma amacıyla görüntü bölütleme yapılarak elde edilen parçalar üzerinden hedef nesne belirlenebilir (Shi ve Malik, 2000) (Şekil 2.2).

Bölütleme ile nesne bulmada kullanılabilecek bir yöntem öğretmenli öğrenme yöntemi ile hedef nesnenin bulunmasıdır. Bu yöntemde, önceden özellikleri çıkarılıp öğretmenli öğrenme yöntemi ile sisteme öğretilmiş bir hedef nesne bulunmaktadır. Bölütleme işlemi sonucu elde edilen her bir parçanın özellikleri çıkarılır ve aday nesnenin hedef nesne olup olmadığı test edilir. Hedef nesne bulunduğunda takip aşamasına geçilir. Bu yöntemde hedef nesneyi temsil eden özelliklerin seçimi sistemin başarılı çalışmasında önemli bir rol oynar.

Bölütleme ile nesne bulmada kullanılabilecek başka bir yöntem ise Şablon Eşleştirme (Template Matching) ile hedef nesnenin bulunmasıdır. Takip edilmesi amaçlanan hedef nesne ile bölütleme sonucu elde edilen her bir parça şablon eşleştirme yöntemi ile birbiriyle kıyaslanır. Hedef nesneye en çok benzeyen parça eğer benzeme oranı eşik seviyesi değerini de geçmişse hedef nesne olarak tanınır ve takibe başlanır.



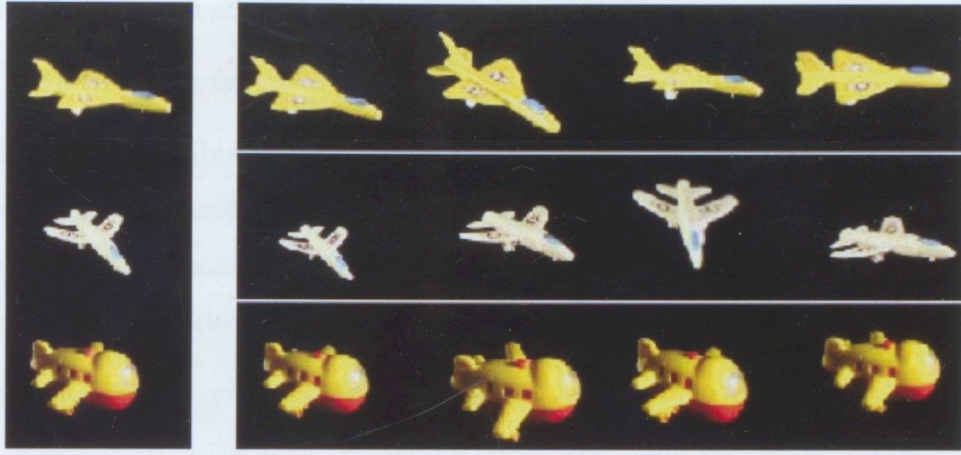
(a)



(b)

Şekil 2.2 Ön plandaki nesnelerin bölütleme ile bulunması (Jianbo Shi)

Bölütleme ile nesne bulmada kullanılabilecek diğer bir yöntemde takip edilecek hedef nesnenin değişik açılardan çekilmiş görüntüleri sisteme öğretilir (Şekil 2.3). Bu sayede nesnenin hareketi sırasında alabileceği değişik pozlar da tanınabilir. Olası nesneler Şablon Eşleştirme yöntemi kullanılarak kıyaslanır ve hedef nesne bulunur.



Şekil 2.3 Nesnelerin değişik açılardan çekilmiş pozları

2.1.4 CamShift Yöntemi ile Nesne Bulma

Bu tez çalışmasında hedef nesneyi takibe başlamadan önce, nesneyi bulma amacıyla CamShift (Continuously Adaptive Mean Shift – Sürekli Uyarlamalı Ortalama Değer Kayması) (Gary R. Bradski, 1998) yöntemi kullanılmıştır. Bu yöntemde takip edilecek nesnenin renk dağılımından elde edilen histogram arama kriteri olarak kullanılmaktadır. Arama alanı olarak verilen görüntü seçilmekte ve ardışık görüntülerde hedef nesnenin renk dağılımını yoğun olarak taşıyan bölge iteratif olarak gerçekleştirilen işlemlerle bulunmaktadır.

2.2 Nesne Takibinde Önemli Rol Oynayan Kriterler

Nesne takibi işlemine başlamadan önce hedef nesneye, ortama ve uygulamaya göre belirlenmesi gereken bazı kriterler vardır. Bunlar temel olarak nesne gösterimi, takip amacıyla kullanılan özellik ve kullanılacak nesne takibi metodudur.

2.2.1 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 takip edilebilir. Nesneyi sembolize etmek için, takip edilecek nesnenin şekil ve hareket yapısına uygun olan bir şekil seçilir (A. Yılmaz, O. Javed, M. Shah, 2006). Şekil 2.4'te farklı gösterim şekilleri görülmektedir.

Noktalar ile Gösterim:

Nesne, merkezinde yer alan bir nokta ile veya noktalar kümesi ile gösterilir. Nokta gösterimi

görüntüde az yer kaplayan nesneleri takip etmek için uygundur.

Geometrik Şekil ile Gösterim:

Nesne dikdörtgen, elips gibi bir geometrik şekil ile gösterilir. Bu şekillerle gösterilen 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.

Nesne Silueti ve Çevre Çizgisi ile Gösterim:

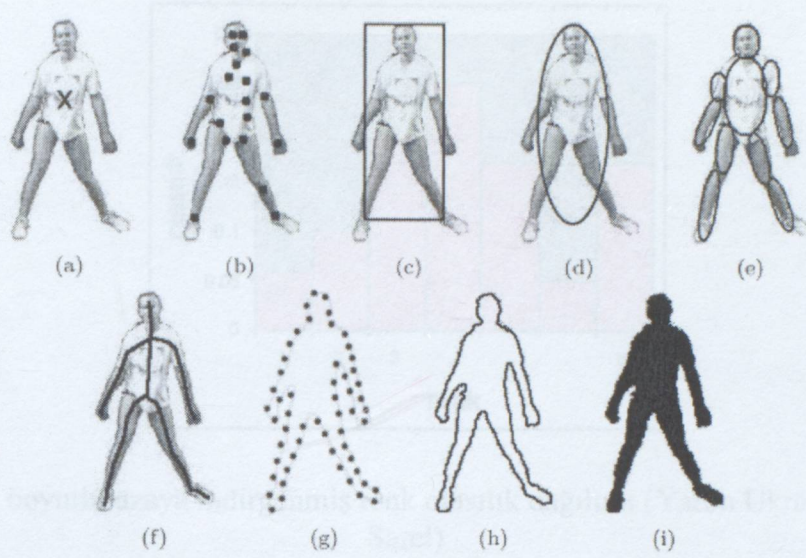
Çevre çizgisi nesnenin sınır çizgilerini temsil eder. Çevre çizgisinin içinde kalan bölge ise nesnenin silueti olarak tanımlanır. Çevre çizgisi ve siluet gösterimlerinin karmaşık ve esnek yapıda nesnelerin takibinde kullanımı uygundur.

Eklemlı Şekil Modelleri ile Gösterim:

Eklemlı nesneler birbirine eklem ile bağlanan parçalardan oluşur. Örneğin 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, nesnenin eklemler ile bağlanan her bir parçası silindir veya elips ile ifade edilebilir.

İskelet Model ile Gösterim:

Nesnenin iskeleti, nesnenin silüetine orta eksen dönüşümü uygulanarak çıkarılabilir. Bu model genellikle nesne tanımada şekil gösterimi olarak kullanılır. İskelet gösterimi, hem eklemlı hem de esnek yapılı nesnelerin gösterimi için kullanılabilir.

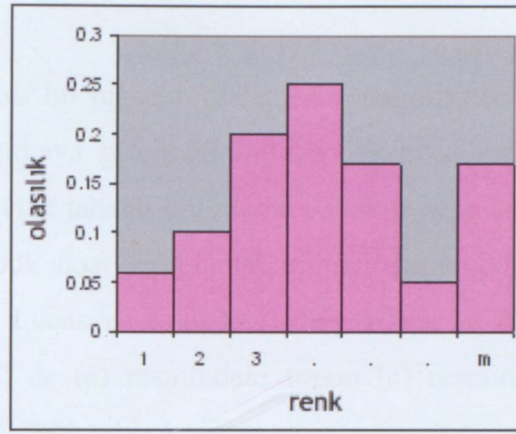


Şekil 2.4 Nesne gösterimleri. (a) Merkez nokta, (b) çoklu nokta, (c) dikdörtgen, (d) elips, (e) parça tabanlı çoklu elips, (f) iskelet, (g) çevre çizgisinde kontrol noktaları, (h) çevre çizgisi, (i) nesne silueti (A. Yılmaz, O. Javed, M. Shah, 2006).

2.3 Özellik Seçimi

Nesne takibinde özellik seçimi kritik bir rol oynar. Seçilecek özelliğin veya özelliklerin takip edilecek nesneyi ayırt edici kılması gerekir. Görsel özelliklerin kullanılmasındaki en önemli sebep de her bir nesnenin kendine özel görsel özellikleri olmasıdır. Özellik seçimi, nesne gösterimi ile yakından alakalıdır. Örneğin histogram tabanlı gösterimlerde renk özelliği kullanılırken, çevre çizgisi tabanlı gösterimlerde kenar özelliği kullanılmaktadır. Çoğu nesne izleme algoritması bu tarz kombinasyonlar kullanmaktadır (A. Yılmaz, O. Javed, M. Shah, 2006).

Renk: Bir nesnenin rengi ışık kaynağının ışık tayfının güç dağılımı ve nesnenin yüzey yansıma özellikleri olmak üzere iki esas fiziksel faktörden etkilenir. Şekil 2.5'te tek boyutlu uzaya indirgenmiş renk olasılık dağılımı görülmektedir. Görüntü işleme çalışmalarında, rengi temsil etme amacıyla genellikle RGB (Red, Green, Blue – Kırmızı, Yeşil, Mavi) renk uzayı kullanılır. Bunun yanında L^*u^*v ve HSV (Hue, Saturation, Value – Renk tonu, Doygunluk, Değer) uzayları da kullanılmaktadır. Nesne takibinde hangi renk uzayının kullanılmasının daha verimli olduğuna dair kesin bir karar yoktur.



Şekil 2.5 Tek boyutlu uzaya indirgenmiş renk olasılık dağılımı (Yaron Ukrainitz, Bernard Sarel)

Kenar:

Nesne dış çizgileri görüntü yoğunluğunda genellikle büyük değişimlere yol açar. Kenarların en önemli özelliği renk özellikleri ile kıyaslandığında ışık değişimlerine daha az duyarlı olmasıdır. Nesnelerin dış çizgisini takip eden algoritmalar özellik olarak genellikle kenar özelliğini kullanırlar. Sonuçlarının basitliği ve kesinliği sebebiyle en çok tercih edilen kenar bulma yaklaşımı Canny Kenar Bulma (Canny Edge Detector) (John F. Canny, 1986) yöntemidir. Şekil 2.6'da (a) resmine Canny Kenar Bulma yöntemi uygulandığında (b) resmi elde edilmiştir.



(a)

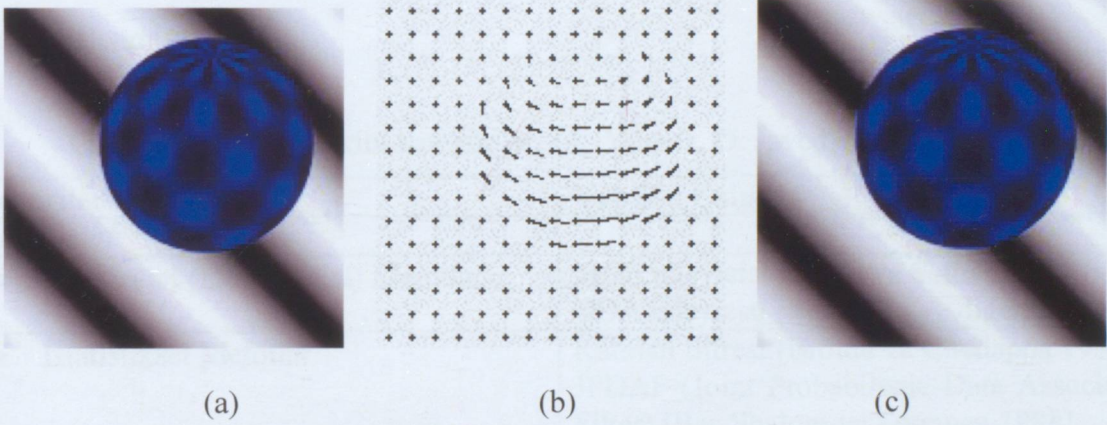


(b)

Şekil 2.6 (a) Orjinal resim (b) Canny kenar bulma uygulandığında elde edilen sonuç

Optik Akış:

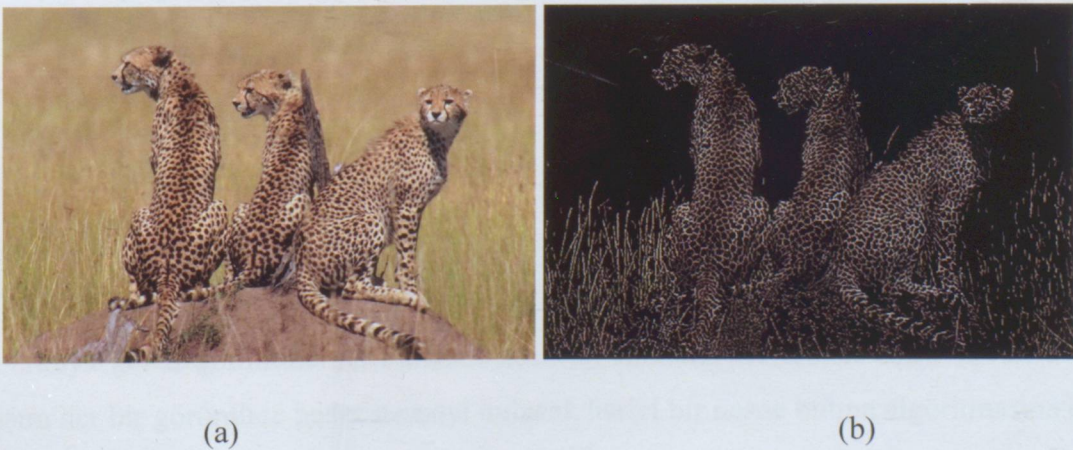
Optik akış, bir bölgedeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Arka arkaya gelen görüntüdeki eş piksellerin parlaklıkları kullanılarak hesaplanır. Optik akış, hareket tabanlı bölütleme ve takip uygulamalarında sıklıkla kullanılan bir özelliktir. Yoğunluk optik akışı hesaplamak için kullanılan popüler yöntemlerden bazıları: Horn ve Schunck [1981], Lucas ve Kanade [1981], Black ve Anandan [1996], Szeliski ve Coughlan [1997]. Şekil 2.7’de (a) resmindeki topun (c) resmindeki yere gelirken hareket değişimi (b) resminde görüldüğü gibi olmuştur.



Şekil 2.7 Optik akış (a) 1. görüntü (b) yön doğrultuları (c) 2. Görüntü
(http://www.cs.otago.ac.nz/gpxpriv/vision_optflow.html)

Doku:

Doku, pürüzsüzlük 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 2.8 (a)’daki resmin doku özellikleri (b) resmindeki gibidir.



Şekil 2.8 (a) Orjinal resim (b) Doku özellikleri

2.4 Nesne Takibi Metotları

Nesneyi temsil etmesi için seçilen model, nesnenin hareketini kısıtlar. Örneğin nesne yalnızca bir nokta ile temsil ediliyorsa, noktanın dolayısıyla nesnenin yapabileceği hareket yalnızca ötelemeli bir harekettir. Elips gibi bir geometrik şekil gösterimi kullanılması durumunda ise ilgin ve izdüşümsel dönüşümler gibi parametrik hareket modelleri uygundur. Bu gösterimler esnek olmayan nesnelerin hareketini çözümlemek için uygundur. Esnek bir nesne içinse, siluet ve çevre çizgisi modelleri gösterim için en uygun modellerdir. Nesnenin hareketini tanımlamak için hem parametrik hem de parametrik olmayan modeller kullanılabilir (A. Yılmaz, O. Javed, M. Shah, 2006). Literatürde sık kullanılan nesne takibi metotları Çizelge 3.1’de listelenmiştir.

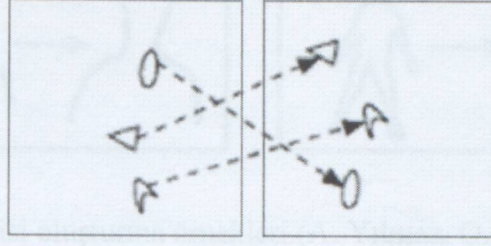
Çizelge 3.1 Nesne takibi kategorileri (A. Yılmaz, O. Javed, M. Shah, 2006)

Kategoriler	Yapılan Çalışma
<i>Nokta Takibi</i>	
• Belirleyici (Determinist) Metotlar	MGE izleyicisi [Salari ve Sethi 1990], GOA izleyicisi [Veenman ve diğerleri 2001].
• İstatistiksel Metotlar	Kalman filtresi [Broida ve Chellappa 1986], JPDAF (Joint Probabilistic Data Association Filter) [Bar-Shalom ve Foreman 1988], PMHT (Probabilistic Multiple Hypothesis Tracking) [Streit ve Luginbuhl 1994].
<i>Çekirdek Takibi</i>	
• Şablon ve yoğunluk tabanlı görünüm modelleri	Mean-shift [Comaniciu ve diğerleri 2003], KLT (Kanade Lucas Tomasi Feature Tracker) [Shi ve Tomasi 1994], Layering [Tao ve diğerleri 2002].
• Çoklu görüşlü görünüm modelleri	Eigentracking [Black ve Jepson 1998], SVM (Support Vector Machine) izleyicisi [Avidan 2001].
<i>Siluet Takibi</i>	
• Çevre çizgisi yayılımı	Durum uzay modelleri [Isard ve Blake 1998], Varyasyon metotları [Bertalmio ve diğ. 2000], Sezgisel metotlar [Ronfard 1994].
• Şekil eşleştirme	Hausdorff [Huttenlocher ve diğerleri 1993], Hough dönüşümü [Sato ve Aggarwal 2004], Histogram [Kang ve diğerleri 2004].

Nokta Takibi:

Arka arkaya gelen görüntülerde, bulunan nesneler noktalar ile temsil edilir (Şekil 2.9). Bu yaklaşım her bir görüntüde hedef nesneyi bulacak harici bir nesne bulma algoritmasına ihtiyaç duyar. Nokta eşleştirme; özellikle kaybolma, yanlış bulma, nesnelerin görüntüye giriş ve

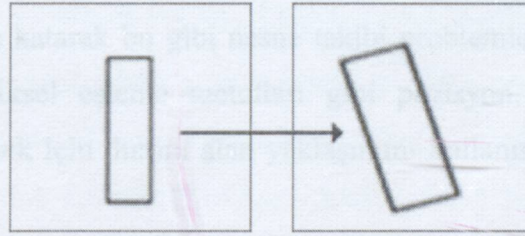
çıkışlarının var olduğu durumlarda karmaşık bir problemdir.



Şekil 2.9 Nokta takibi (A. Yılmaz, O. Javed, M. Shah, 2006)

Çekirdek Takibi:

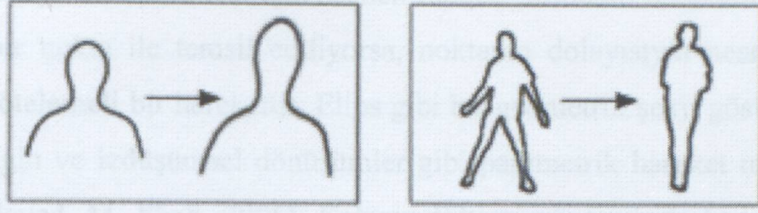
Çekirdek, objenin görünümünü ve şeklini temsil eder. Örneğin, çekirdek o bölgenin histogramı ile ilişkilendirilmiş dikdörtgen bir şablon veya eliptik bir şekil olabilir. Bu yöntemde nesne takibi, arka arkaya gelen görüntülerdeki çekirdeğin hareketinin hesaplanmasıyla gerçekleştirilir. Bu hareket genellikle öteleme, dönme ve ilgin dönüşüm gibi parametrik dönüşümler şeklinde olmaktadır (Şekil 2.10).



Şekil 2.10 Dikdörtgen şeklin parametrik dönüşümü (A. Yılmaz, O. Javed, M. Shah, 2006)

Siluet Takibi:

Nesneler çok karmaşık şekillere sahip olabilirler. Örneğin insan eller, kafa, omuzlar vs. gibi eklemlerle bağlanmış parçalardan oluşmaktadır. Bu yapıda bir nesnenin basit geometrik şekillerle tanımlanması zordur. Siluet tabanlı metotlar bu tarz nesneler için kesin şekil tanımlamaları sunar (Şekil 2.11). Siluet tabanlı obje takibinde amaç, önceki görüntülerde tanımlanmış nesne modelinin kullanılmasıyla arka arkaya gelen her bir görüntüdeki nesne bölgesinin bulunmasıdır. Kullanılan bu nesne modeli renk histogramı, nesne kenarları veya nesne dış çizgisi şeklinde olabilir. Siluet takibi, şekil eşleştirme ve çevre çizgisi takibi olmak üzere iki kategoriye ayrılabilir. Şekil eşleştirme yaklaşımları, o anki görüntüde nesnenin silüetini aramaktadır. Çevre çizgisi yaklaşımları ise durum uzay modellerini veya enerji fonksiyonlarının doğrudan minimizasyonunu kullanarak o anki görüntüdeki nesnenin yeni pozisyonuyla bir çevre çizgisini ilişkilendirir.



Şekil 2.11 Çevre çizgisi oluşturma örnekleri (A. Yılmaz, O. Javed, M. Shah, 2006)

2.4.1 Kalman Filtresi ile Nesne Takibi

Nesne takibi amacıyla kullanılan yöntemlerden en bilineni Kalman filtresidir (Kalman R. E., 1960). Bu yöntemin uygulamasının basit olmasından ve gerçek zamanlı çalışabilmesinden dolayı literatürde birçok çalışmada kullanılmıştır. Kalman filtresi, nesne takibinde hedef nesnenin doğrusal hareket ettiğini ve sistemde Gauss paraziti bulunduğunu kabul eder.

Nesne takibi amacıyla kameralardan elde edilen ölçümler çoğu zaman parazit içermektedir. Bunun yanında nesne hareketleri de rastgele düzensizlikler içerebilir. Kalman filtresinin de dahil olduğu istatistiksel eşleme metotları, nesne durum tahmini boyunca ölçüm ve model belirsizliklerini de hesaba katarak bu gibi nesne takibi problemlerine çözüm üretir. Kalman Filtresi de diğer istatistiksel eşleme metotları gibi pozisyon, hız ve ivme gibi nesne özelliklerini belirleyebilmek için durum alan yaklaşımını kullanır (A. Yılmaz, O. Javed, M. Shah, 2006).

Kalman filtresi, Gauss dağılımı olan durumlarda doğrusal bir sistemin durumunu tahmin etmek için kullanılır. Kalman filtrelemesi üç adımdan oluşur: önerme, doğrulama ve asimilasyon.

Bu algorithmada amaç, $t-1$ anındaki sistem model önermesi ile t anındaki ölçümler kullanılarak t anındaki sistemin durumunu bulmaktır. Önerme aşamasında, $t-1$ anındaki sistem modeline ve nesne durumuna dayanılarak t anındaki nesnenin durumu önerilir. Ölçüm aşamasında, t anındaki görüntü üzerindeki özellikler çıkarılır. Bu özellikler, nesne durumunun doğrusal dönüşümü olarak kabul edilir. Asimilasyon aşamasında ise, önerilmiş durumlar ile ölçülmüş durumlar kombine edilerek nesnenin yeni durumu çıkarılır (Neil Alldrin).

Kalman filtresinin, tek şekilli ve Gauss olasılık dağılımının olduğu durumlarda çalışması bu algoritmanın önemli bir sınırlamasıdır. Çünkü genellikle nesneler Gauss dağılımlı değildir. Kalman filtresinin başka bir dezavantajı da karmaşık arka planlara karşı hassas olmasıdır.

Kalman Filtresi, bir sonraki görüntü üzerinde nesnenin yeri hakkında tek bir tahmin yapar. Bu sebepten ötürü Kalman Filtresi ile yalnızca tek bir nesnenin takibi yapılabilir.

Bilindiği gibi nesneyi temsil etmesi için seçilen model, nesnenin hareketini kısıtlar. Örneğin nesne yalnızca bir nokta ile temsil ediliyorsa, noktanın dolayısıyla nesnenin yapabileceği hareket yalnızca ötelemeli bir harekettir. Elips gibi bir geometrik şekil gösterimi kullanılması durumunda ise ilgin ve izdüşümsel dönüşümler gibi parametrik hareket modelleri uygundur (A. Yılmaz, O. Javed, M. Shah, 2006). Kalman Filtresi yönteminde hedef nesne, nokta ile temsil edilmektedir. Eğer hedef nesne geometrik bir şekil ile temsil edilebilseydi takip işlemi sırasında nesnenin boyutlarındaki değişiklik ölçülebilirdi. Bu nedenle Kalman Filtresi yöntemi ile yapılan nesne takibinde hedef nesnenin boyutları, kapladığı alan ölçülemez yalnızca görüntü üzerindeki yeri bulunabilir.

2.4.2 - CONDENSATION Yöntemi ile Nesne Takibi

CONDENSATION (Conditional Density Propagation – Koşullu Yoğunluk Yayılımı) algoritmasının temel kullanım alanı karmaşık bir ortamda hareket eden nesnelerin dış çizgilerinin bulunması ve takip edilmesidir. Görüntü üzerindeki piksellerin hangilerinin nesnenin kenarlarına ait olduğunu bulmak önemli bir problemidir. CONDENSATION bu problemi çözmeye çalışan olasılıksal bir algoritmadır (Isard M., Blake A., 1998).

CONDENSATION algoritması, 1998 yılında Isard ve Blake tarafından tanımlanmıştır. Bu algoritma, parçacık filtresinin bir varyasyonu olup doğrusal olmayan, Gauss olmayan takip problemlerinin de üstesinden gelmektedir. Algoritmanın farklı yönlerinden biri görüntüdeki her bir piksel üzerinde işlem yapmıyor olmasıdır. Bunun yerine, işlenecek pikseller rastgele seçilmekte ve yalnızca bu piksellerin bir alt kümesi işleme sokulmaktadır.

CONDENSATION her bir iterasyonunda üç tane adım içeren iteratif bir algoritmadır. Bu adımlar sırayla önerme, güncelleme ve yeniden örneklemedir. CONDENSATION, sıralı MonteCarlo önem örneklemesini kullanan bir Bayes özyinelemeli tahmincidir (P. Barrera, J. M. Canas, V. Matellan, 2005).

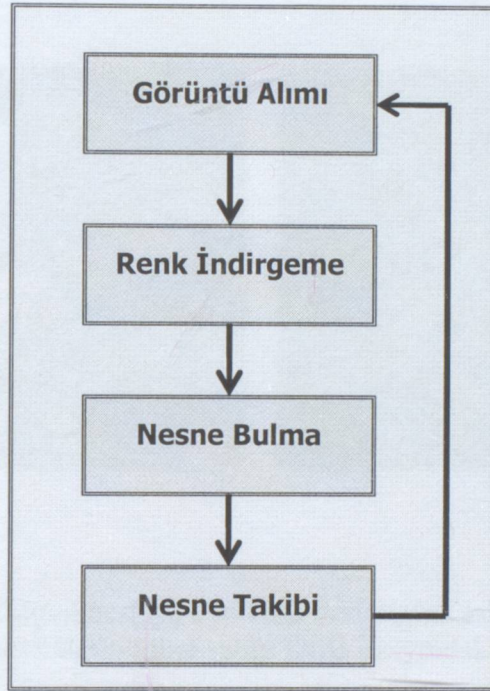
Kısaca anlatılacak olursa, sıralı gözlemler kullanılarak o anki çok boyutlu durum tahmin edilir. Gözlemler, durum ile olasılıksal gözlem modeli üzerinden ilişkilidir. Durum dinamik olabilir, bu dinamizm hareket modeli içerisinde yakalanır. Algoritmanın sıralı doğası iteratif eşitlikler sağlar, örnekleme doğası ise N parçacıktan oluşan kümeyi yönetme yetisi sağlar.

Her bir parçacık bir durum tahminini temsil eder ve buna ilişkilendirilmiş bir ağırlığa sahiptir. Global tahminler parçacık kümesinin tamamına dayanarak yapılabilir.

CONDENSATION'ın her iterasyonundaki önerme adımında, yeni bir parçacık elde edilerek ve parçacık kümesi oluşturularak her bir parçacığın hareket modeli örneklenir. Güncelleme

3. HAREKETLİ BİR NESNENİN HAREKETLİ KAMERA İLE GERÇEK ZAMANLI TAKİBİ

Bu çalışmada gerçekleştirilen, hareketli bir nesnenin hareketli kamera ile gerçek zamanlı takibini sağlayan sistemin ana adımlarını gösteren blok diyagramı Şekil 3.1’de görüldüğü gibidir.



Şekil 3.1 Sistem blok diyagramı

3.1 Görüntü Alımı

Sistem, hareketli nesne takibini hareketli bir düzeneğe yerleştirilmiş kamerayla alınan görüntü akışı üzerinden yapmaktadır. 90^0 görme açısına sahip kameradan alınan renkli görüntüler 640×480 'lik genişlikte olup çözünürlükleri 96 DPI, saniyede alınan görüntü sayısı ise 10'dur.

3.2 Renk İndirgeme

Kameradan alınan renkli görüntülerin RGB bileşenlerinin her biri 256 renge sahiptir. Sistemin karmaşık yapısı, bunun yanında gerçek zamanlı çalışıyor olması açısından programın çalışma zamanını azaltmak için 256 renkli sahip RGB uzayı Bit-Cutting (Rudolf K. Bock, 1998) yöntemi ile 16 renge indirgenmiştir. Bunun için, her bir renk değerinin düşük anlamlı 4 bitine bakılmaksızın yüksek anlamlı 4 bitine bakılarak o rengin yeni değeri belirlenmiştir. Başka bir deyişle üzerinde çalışılan toplam renk sayısı $256 \times 256 \times 256$ 'dan $16 \times 16 \times 16$ 'ya indirilmiştir.

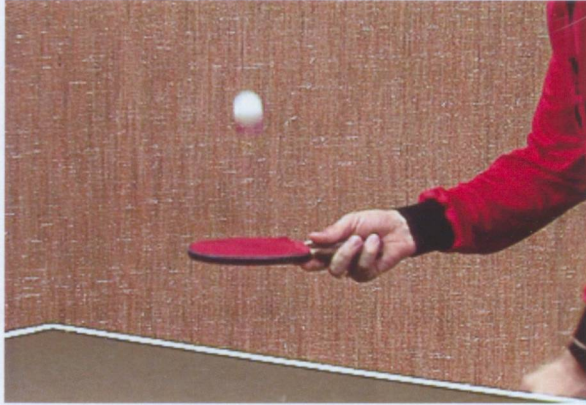
Alınan görüntülerde her bir renk bileşeni değeri 8 bit ile ifade edilmektedir. 8 bitle $256 (2^8)$

farklı sayı gösterilebilir. Her rengi 4 bitle ifade edersek renk derinliği de 16^4 'ya (2^4) düşmüş olur. Örneğin aşağıda ikili düzen karşılığında görüldüğü gibi renk bileşeni değeri 0-15 olan renkler 0 ile 240-255 arası renkler 240 ile gösterilmektedir.

0000 0000 – 0000 1111 → 0000 0000

1111 0000 – 1111 1111 → 1111 0000

Şekil 3.2 (a)'da 256x256x256 renkli orjinal resim (b)'de ise 16x16x16 renge indirgenmiş hali görülmektedir.



(a)



(b)

Şekil 3.2 Renk uzayının indirgenmesi (a) 256 renk derinliğine sahip RGB uzayındaki görüntü (b) 16 renk derinliğine sahip RGB uzayındaki görüntü

16x16x16 renge indirgenmiş görüntü üzerinde çalışılırken, öncelikle aşağıda görüldüğü gibi her bir rengi (kırmızı, yeşil, mavi) temsil eden 8 bitlik sayı dört kere sağa kaydırılır, yani yüksek anlamlı 4 bit düşük anlamlı 4 bite yerleştirilir ve 8 bitlik sayı 4 bitle ifade edilmiş olur.

1000 0000 → 0000 1000

1011 0000 → 0000 1011

1111 0000 → 0000 1111

Bundan sonra kırmızı, yeşil ve mavi renklerin 4 bitleri yan yana birleştirilerek 12 bitlik bir sayı elde edilir ve her bir pikselin renk değeri bu 12 bitlik sayı ile ifade edilir.

Kırmızı (R) → 0000 0100

Yeşil (G) → 0000 1101

Pikselin ifade edildiği renk değeri → 0100 1101 1111

Mavi (B) → 0000 1111

3.3 Nesne Bulma ve Takibi

Bu tez çalışmasında nesne takibi iki farklı yöntemle gerçekleştirilmiştir. İlk olarak Mean-Shift yöntemi ile nesne bulma ve takibi gerçekleştirilmiştir (D. Comaniciu, V. Ramesh ve P. Meer, 2000). Elde edilen çıktılar sonucunda sistemin problemleri göz önünde bulundurularak, bu

problemleri çözmek amacıyla Mean-Shift yönteminin iyileştirilmiş hali olan Cam-Shift yöntemi ile nesne bulma ve takibi gerçekleştirilmiştir.

3.3.1 Mean-Shift ile Nesne Bulma

Oluşturulan sistemin gerçek zamanlı çalışıyor olması sistemi oluşturan modüllerde kullanılacak yöntemlerin belirlenmesinde birinci derecede önemli kriterdir. Bu nedenle kullanılacak yöntemin başarısının yanında çalışma süresi de çok önemlidir.

Hareketli kamera, hedef nesne görüntüde belirene kadar bulunduğu ortamı gözlemlemektedir. Bu süre içerisinde ortamda hedef nesneyi bulmaya çalışmaktadır. Hedef nesneyi bulmak için video akışı üzerinden alınan 16 renk derinliğine indirgenmiş görüntüler üzerinde çalışılır.

Hedef nesneyi temsil etmek için öncelikle bir özellik uzayı seçilir. Bu tez çalışmasında RGB renk özellik uzayı kullanılmıştır. Hedef nesnenin yerinin Mean-Shift yöntemi ile bulunmasına ait işlem adımları Şekil 3.3'te görüldüğü gibidir.


```

Hedef nesnenin renk olasılık yoğunluk fonksiyonu hesaplanır ( $q$ )
Arama alanının renk olasılık yoğunluk fonksiyonu hesaplanır ( $p$ )

detected = FALSE;

while(detected == FALSE){

    Sıradaki görüntü kameradan alınır

    for(i = 0; i < iterationNum; i++){

        Ağırlıklar hesaplanır (weights)

        Dikdörtgen bölgenin yeni merkezi bulunur (inew, jnew)

        Arama alanının renk olasılık yoğunluk fonksiyonun
        Hesaplanması ( $p$ )

        // q ile p arası benzerlik bulunur
        bhat = bhattacharyya(q, p);

        iold = inew;
        jold = jnew;
    }

    if(bhat > threshold)
        detected = TRUE;
}

```

Şekil 3.3 Mean-Shift ile nesne yerinin bulunmasına ait yarı kodlama

Hedef model, görüntü üzerinde dikdörtgen bir bölge ve bu bölge içerisindeki renk olasılık yoğunluk fonksiyonu (q) ile temsil edilir. Bu dikdörtgen bölgenin merkezi (0, 0) noktası olarak kabul edilir. Bölgedeki diğer piksellerin koordinatları da bu kabule göre normalize edilir. Bölgedeki piksel koordinatları, $\{x_{ij}\}_{i=1..n}$ ile ifade edilir. Dikdörtgen bölgedeki her bir piksele merkezden uzaklaştıkça azalan bir ağırlık değeri ($k(x)$) atanır. Bu ağırlık ataması olasılık yoğunluk fonksiyonunun sonuçlarını güçlendirir. Çünkü merkezden uzak bölgelerin

başka bir nesnenin arkasında kalıp kaybolma olasılığı yüksektir (D. Comaniciu, V. Ramesh ve P. Meer, 2003). Hedef modeldeki u özelliğinin bulunma olasılığı Eşitlik 3.1'de görülmektedir.

$$q_u = C \sum_{i=1}^n k\left(\|x_i\|^2\right) \delta[b(x_i) - u] \quad (3.1)$$

δ Kronecker delta fonksiyonudur (Eşitlik 3.2).

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (3.2)$$

C sabiti $\sum_{i=1}^n q_u = 1$ eşitliğinden yararlanılarak çıkarılmıştır (Eşitlik 3.3).

$$C = \frac{1}{\sum_{i=1}^n k\left(\|x_i\|^2\right)} \quad (3.3)$$

Hedef nesnenin renk dağılımına göre görüntüde hedef nesne aranmaktadır. Arama alanı bütün görüntü olarak seçilmektedir. Görüntüde kullanılan renklerin olasılık yoğunluk fonksiyonu (p) bulunur (Eşitlik 3.4).

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (3.4)$$

Hedef nesnenin ve görüntünün renk olasılık dağılımdan yararlanılarak ağırlıklar hesaplanır (Eşitlik 3.5).

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (3.5)$$

Daha sonra Mean-Shift (Ortalama Kayması) vektöründen yararlanılarak arama alanının merkezinin yeni koordinatları belirlenir (Eşitlik 3.6).

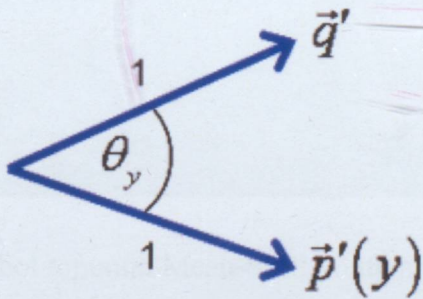
$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \quad (3.6)$$

İteratif olarak bu işlemler tekrar edilerek hedef nesnenin olası merkezi bulunmuş olur. Daha sonra bu olası bölgenin olasılık yoğunluk fonksiyonu ile hedef nesnenin olasılık yoğunluk fonksiyonu Bhattacharyya uzaklığı (Eşitlik 3.5) kullanılarak kıyaslanır.

$$\rho[p(y_0), q] = \sum_{u=1}^m \sqrt{p_u(y_0) q_u} \quad (3.5)$$

Bhattacharyya uzaklığı iki vektör arasındaki benzerliği vermektedir (Şekil 3.4).

$$\vec{q}' = (\sqrt{q_1}, \dots, \sqrt{q_m})$$

$$\vec{p}'(y) = (\sqrt{p_1(y)}, \dots, \sqrt{p_m(y)})$$


Şekil 3.4 Bhattacharyya uzaklığı

Bhattacharyya uzaklığı 0-1 arası değerler üretmektedir. İki olasılık yoğunluk fonksiyonunun kıyaslanması ile elde edilen sonuç 1'e ne kadar yakınsa benzerlik de o kadar fazladır. Pencere olasılık yoğunluk fonksiyonu ile hedef nesnenin olasılık yoğunluk fonksiyonu kıyaslandığında hedef nesneye en fazla benzeyen pencerenin benzerlik oranına bakılır. Eğer bu oran önceden belirlenmiş olan eşik seviyesi değerinin üzerindeyse hedef nesnenin bu bölgede olduğuna karar verilir. Bu bölgenin merkez koordinatları nesne takibi aşamasına aktarılarak nesne takibine başlanır. Eğer benzerlik derecesi eşik seviyesini geçmediyse hedef nesne bulunana kadar ortam gözlenmeye devam edilir. Mean-Shift ile tespit edilmiş örnek bir hedef nesne Şekil 3.6'da görülmektedir. Hedef nesnenin nesne bulma aşamasına başlamadan

önce alınmış olan görüntüsü de Şekil 3.5'te görülmektedir.



Şekil 3.5 Hedef nesnenin görüntüsü

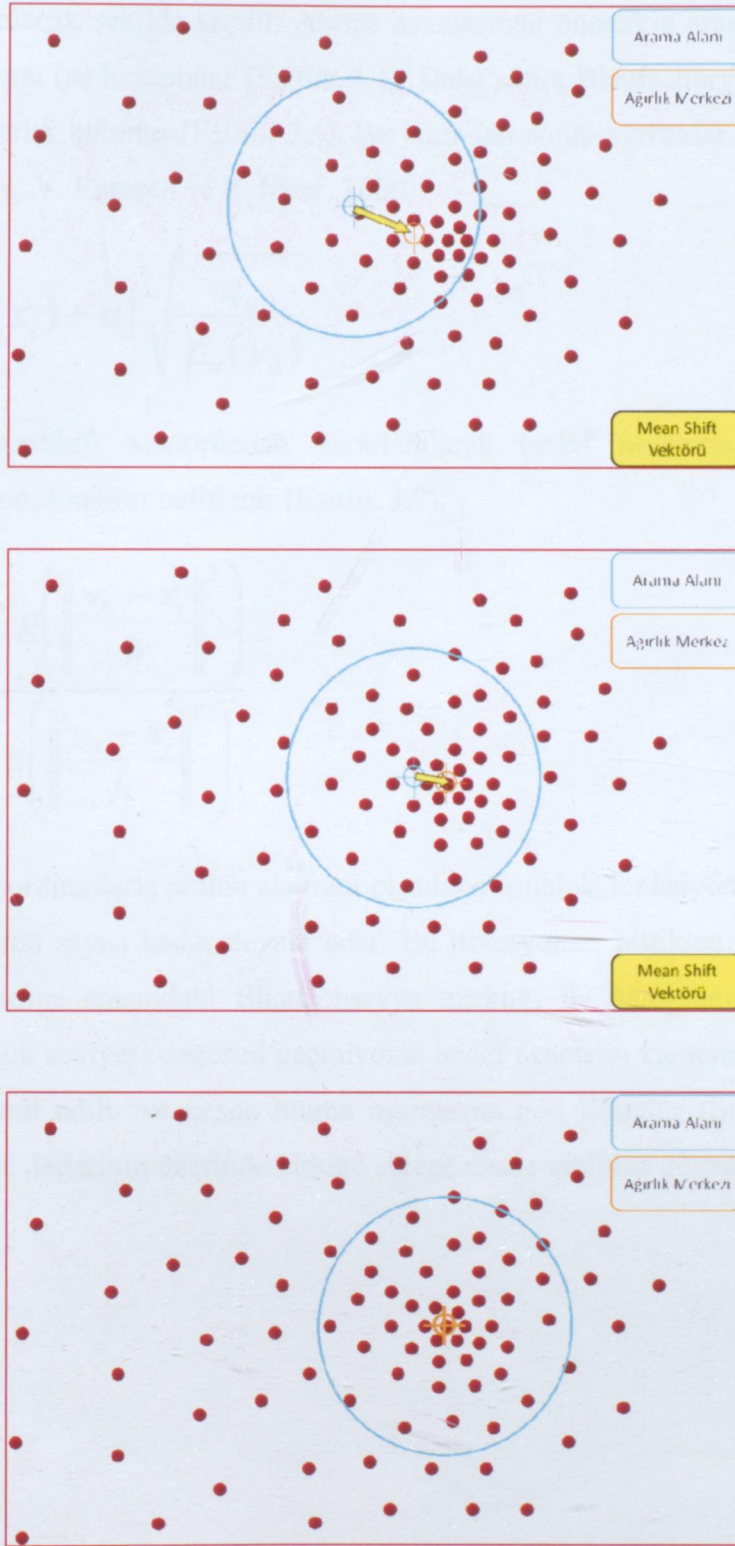


Şekil 3.6 Futbol topunun Mean-Shift yöntemi ile bulunması

3.3.2 Mean-Shift Yöntemi ile Nesne Takibi

Hedef nesne kameranın görüş açısına girdiğinde nesne bulma modülü tarafından hedef nesnenin görüntüdeki yeri saptanmakta, hedef nesnenin görüntüde yer aldığı koordinatlar nesne takibi aşamasına aktarılmakta ve nesne takibi başlamaktadır.

Amaç, arka arkaya gelen görüntüler üzerinde hedef nesnenin yerini belirleyerek hedef nesnenin hareket doğrultusunu çıkarmaktır (Şekil 3.7). Bu amaç doğrultusunda ilk olarak takip edilecek hedef nesnenin olasılık yoğunluk fonksiyonu çıkarılmaktadır. Daha sonra video akışındaki bir önceki görüntüde hedef nesnenin bulunduğu, nesne bulma modülünden gelen koordinat bilgisi kullanılarak bir sonraki görüntüde hedef nesne aranmaktadır.



Şekil 3.7 Hedef nesnenin olasılık dağılımının en yoğun olduğu merkezin bulunması (Yaron Ukrainitz ve Bernard Sarel)

Bu çalışmada, hedef nesne dikdörtgen bir bölge ve o bölgeye ait olasılık yoğunluk fonksiyonu (q) ile temsil edilmektedir (Eşitlik 3.1). t . görüntüde hedef nesnenin yeri bulunmakta, $t+1$. görüntüde hedef nesne aranmaktadır. Arama alanı, merkezinin ilk değeri bir önceki görüntüde

hedef nesnenin bulunduğu koordinatlar olacak şekilde, boyutları ise hedef nesnenin kapladığı alanın 3 (n) katı olacak şekilde seçilir. Arama aşamasında öncelikle arama alanının olasılık yoğunluk fonksiyonu (p) hesaplanır (Eşitlik 3.4). Daha sonra Bhattacharyya uzaklığı ile p ile q arasındaki benzerlik bulunur (Eşitlik 3.5). Bu adımdan sonra ağırlıklar güncellenir (Eşitlik 3.6) (D. Comaniciu, V. Ramesh ve P. Meer, 2000).

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (3.6)$$

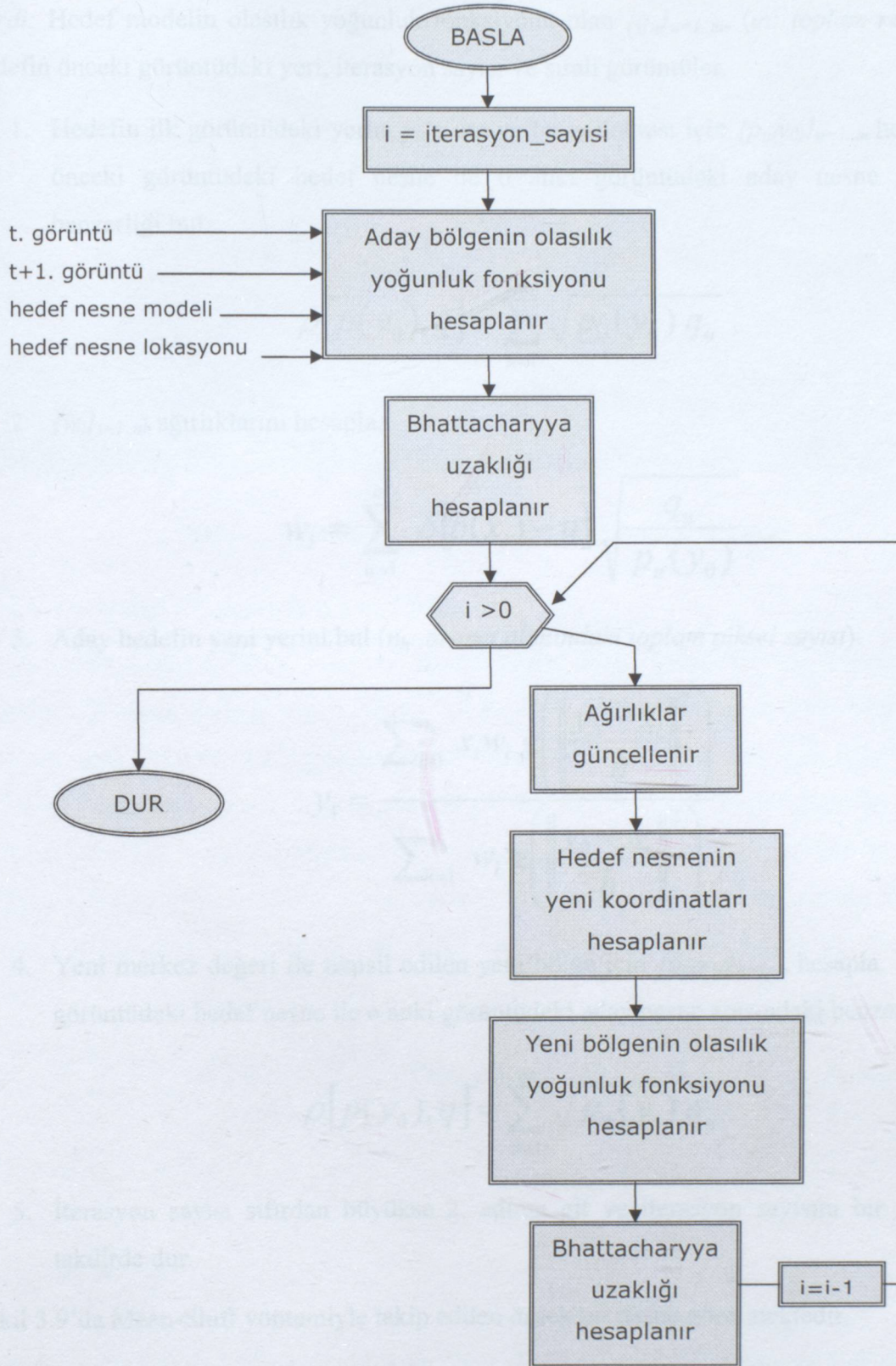
Daha sonra Mean-Shift vektöründen yararlanılarak hedef nesnenin (arama alanının) merkezinin yeni koordinatları belirlenir (Eşitlik 3.7).

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \quad (3.7)$$

Yeni belirlenen koordinatlarla arama alanının olasılık yoğunluk fonksiyonu tekrar hesaplanır. Bu adımlar iterasyon sayısı kadar devam eder. Bu iterasyonlar bittikten sonra eğer bulunan bölge ile hedef nesne arasındaki Bhattacharyya uzaklığı ile hesaplanmış olan benzerlik, belirlenmiş olan eşik seviyesi değerini geçmiyorsa hedef nesnenin kameranın görüş açısından çıkmış olduğu kabul edilir ve nesne bulma aşamasına geri dönülür (Şekil 3.8). Benzerlik değeri eşik seviyesi değerinin üzerinde olduğu sürece nesne takibine devam edilir.

Şekil 3.8 Mean-Shift yöntemi ile nesne takibine ait algoritmaya ait blok diyagramı

Mean-Shift Yöntemi ile Nesne Takibi Akış Diyagramı:



Şekil 3.8 Mean-Shift yöntemi ile nesne takibine ait akış diyagramı

Mean-Shift ile Nesne Takibi Algoritmasının Adımları:

Girdi: Hedef modelin olasılık yoğunluk fonksiyonu olan $\{q_u\}_{u=1..m}$, (m : toplam renk sayısı) hedefin önceki görüntüdeki yeri, iterasyon sayısı ve sıralı görüntüler.

1. Hedefin ilk görüntüdeki yerini y_0 'a ata ve bu y_0 noktası için $\{p_u(y_0)\}_{u=1..m}$ hesapla. Bir önceki görüntüdeki hedef nesne ile o anki görüntüdeki aday nesne arasındaki benzerliği bul.

$$\rho[p(y_0), q] = \sum_{u=1}^m \sqrt{p_u(y_0) q_u}$$

2. $\{w_i\}_{i=1..n_h}$ ağırlıklarını hesapla.

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}}$$

3. Aday hedefin yeni yerini bul (n_h : arama alanındaki toplam piksel sayısı).

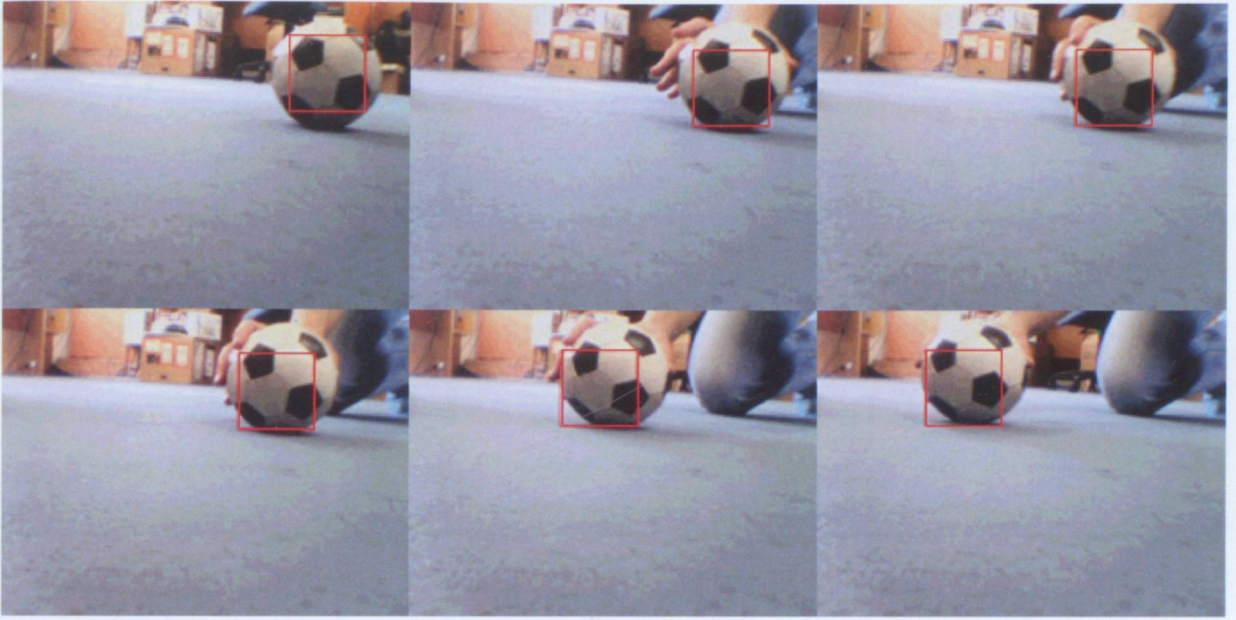
$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}$$

4. Yeni merkez değeri ile temsil edilen yeni bölge için $\{p_u(y_1)\}_{u=1..m}$ hesapla. Bir önceki görüntüdeki hedef nesne ile o anki görüntüdeki aday nesne arasındaki benzerliği bul.

$$\rho[p(y_1), q] = \sum_{u=1}^m \sqrt{p_u(y_1) q_u}$$

5. İterasyon sayısı sıfırdan büyükse 2. adıma git ve iterasyon sayısını bir azalt, aksi takdirde dur.

Şekil 3.9'da Mean-Shift yöntemiyle takip edilen örnek bir nesne görülmektedir.



Şekil 3.9 Futbol topunun Mean-Shift yöntemi ile takibi

3.3.3 CamShift Yöntemi ile Nesne Bulma

Bu tez çalışmasında hedef nesneyi takibe başlamadan önce, nesneyi bulma amacıyla CamShift yöntemi kullanılmıştır. Nesne bulma aşamasına girdi olarak takip edilecek hedef nesnenin renk dağılımı verilmektedir. Nesne bulma adımında ilk olarak renk dağılımı verilen hedef nesnenin histogramı çıkarılmaktadır (Eşitlik 3.13). Daha sonra çıkarılan histogram değerleri [0-255] arasına normalize edilmektedir (Eşitlik 3.14). Elde edilen bu normalize edilmiş histogram moment hesaplamalarında kullanılmak üzere saklanmaktadır. Daha sonra alınan görüntü üzerinde sıfırıncı ve birinci momentler hesaplanmaktadır (Eşitlik 3.8, 3.9, 3.10).

$$M_{00} = \sum_x \sum_y P(x, y) \quad P(x, y) \rightarrow \text{Olasılık Yoğunluk Görüntüsü} \quad (3.8)$$

$$M_{10} = \sum_x \sum_y x P(x, y) \quad (3.9)$$

$$M_{01} = \sum_x \sum_y y P(x, y) \quad (3.10)$$

Moment hesabından sonra bu momentlerden yararlanılarak hedef nesnenin görüntüde yer aldığı olası merkez bulunur (Eşitlik 3.11, 3.12).

$$x_c = \frac{M_{10}}{M_{00}} \quad (3.11)$$

$$y_c = \frac{M_{01}}{M_{00}} \quad (3.12)$$

Belirlenen iterasyon sayısı kadar dönülür; bu iterasyonlar boyunca ilgili görüntüde moment ve yeni merkez hesabı yapılır. Elde edilen merkez koordinatlarından ve hedef nesnenin olası boyutlarından yararlanılarak hedef nesnenin yer aldığı aday bölge belirlenir. Daha sonra bu bölgedeki renk olasılık yoğunluk dağılımı bulunur. Elde edilen bu dağılımla hedef nesnenin renk olasılık yoğunluk dağılımının benzerliği Bhattacharyya uzaklığı kullanılarak hesaplanır. Hesaplanan benzerlik değeri, daha önceden belirlenmiş olan bir eşik seviyesi değerinin üzerindeyse aday bölgenin hedef bölge olduğuna karar verilir. Hedef nesnenin bulunduğu yer bilgisi nesne takibi aşamasına aktarılır ve bu noktada nesne takibi aşaması başlar. Aday bölge ile hedef nesnenin benzerliğinin eşik seviyesinin altında olması durumunda ise nesne bulma aşaması bir sonraki görüntüyü alarak çalışmaya devam eder.

3.3.4 CamShift Yöntemi ile Nesne Takibi

Sistemin uygulama aşamasında değişik problemlerle karşılaşmış, değişik senaryolar üretilmiş ve bu durumlara karşı çeşitli çözümler üretilmeye çalışılmıştır. Bu tez kapsamında öncelikle Mean-Shift yöntemi ile nesne takibi gerçekleştirilmiştir. Hareketli nesnenin kameraya olan uzaklığının değiştiği, nesnenin bir kısmının veya tamamının kaybolduğu durumlarda Mean-Shift yönteminin yetersiz kaldığı görülmüş ve bu durumlara karşı başarılı çözümler üreten CamShift yöntemi seçilmiştir.

CamShift algoritması, yalnızca statik dağılımlara çözüm getirebilen Mean-Shift algoritması temel alınarak üretilmiş efektif bir takip algoritmasıdır. Nesnenin olasılık dağılımının yeri ve büyüklüğü; nesnenin hareketine, ışık değişimine, görüş açısının değişimine, gölgeye göre değişiklik gösterir. CamShift algoritması dinamik olarak değişen dağılımlar için tasarlanmıştır (Gary R. Bradski, 1998).

CamShift yöntemi kullanılarak tasarlanan nesne takibi sisteminde ardışık her bir görüntüde nesnenin yerinin bulunması için kullanılan olasılık yoğunluk fonksiyonu renk bilgisinden yararlanılarak oluşturulur. Algoritma, ortalama yoğunluğun maksimum artış gösterdiği yönde iteratif olarak çalışarak dağılımın ortalamasını bulur. Olasılık yoğunluğu ardışık görüntülerin her birinde tekrardan hesaplanmaktadır. Olasılık görüntüsündeki her bir piksel, o anki görüntüdeki ilgili pikselin renk değerinin hedef nesneye ait bölgedeki olasılığını temsil eder (Eşitlik 3.14). CamShift yönteminde, Mean-Shift algoritmasından farklı olarak iterasyonlar boyunca dağılımın merkezine doğru moment hesabı yapılır (Bogdan Kwolek, 2005).

Hedef bölgedeki renk dağılımını temsil etmek için literatürde parametrik ve parametrik olmayan çeşitli metotlar kullanılmaktadır. Histogram en sık kullanılan parametrik olmayan olasılık tahmin metodudur. Histogram, ilgili bölgedeki piksellerin renk dağılımı bulunarak elde edilir. Yapılan renk kuantalama işlemi sayesinde benzer renkler aynı gruba dahil edilir ve aynı olasılık değerine sahip olur. Bu sayede işlem yükü ve hafıza kullanımı azaltılmış olur (Bogdan Kwolek, 2005). Ağırlıksız histogram, aşağıdaki gibi hesaplanır (Eşitlik 3.13):

$$q_u = \sum_{i=1}^n \delta[c(x_i) - u] \quad (3.13)$$

$c(x_i)$ fonksiyonu x_i koordinatındaki pikselin değerini, n bölgedeki toplam piksel sayısını, δ ise Kronecker delta fonksiyonunu simgelemektedir.

Özellik olarak kullanılan nesnenin renk dağılımı, nesnenin kameraya olan uzaklığının artıp azalmasıyla değişiklik göstermez. Kenar tabanlı uygulamaların efektif olmadığı, nesnenin bir kısmının kaybolduğu durumlarda da renk dağılımı özelliği başarılı sonuçlar vermektedir.

Olasılık yoğunluk görüntüsü $P(x,y)$, histogram çıkarımından yararlanılarak 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 hedef modelde yer alma olasılığı ile temsil edilmiş olur. Olasılık değerlerinin 0 ile 255 arası değerler alabilmesi için histogram değerleri lineer olarak aşağıdaki formüldeki gibi normalize edilir (Eşitlik 3.14):

$$p_u = \min\left(\frac{255}{q_{\max}} q_u, 255\right), \quad u = 1, \dots, m \quad q_{\max} = \left\{ \max(q_u) \right\}_{u=1}^m \quad (3.14)$$

CamShift Yöntemi ile Nesne Takibi Adımları:

1. Arama alanının merkezi seçilir (x_0, y_0).
2. Hedef nesnenin konumunu (x_1, y_1) bulmak için Mean-Shift iterasyonları gerçekleştirilir, sıfıncı moment (M_{00}) saklanır.
3. Arama alanı büyüklüğü sıfıncı momenti kullanan bir fonksiyon ile hesaplanır.
4. 2. ve 3. adımlar merkez koordinatların yer değişimi belirli bir eşik seviyesinin altında olduğu sürece tekrarlanır.

Takip edilecek hedef nesnenin renk olasılık dağılımı girdi olarak alınıp nesne bulma işlemi ile nesnenin merkez koordinatları bulunduğundan sonra CamShift ile nesne takibi aşamasına geçilir.

Takip aşamasında gerçek zamanlı olarak alınan görüntülerde hedef nesneyi bulma amacı ile özellik olarak seçilen renk olasılık dağılımı aranmaktadır. Takip aşamasının ilk adımında, bir önceki adım olan nesne bulmadan gelen hedef nesnenin merkezinin görüntüde yer aldığı koordinatlar arama alanının merkezi olarak belirlenir. Bu alanda hedef nesnenin renk olasılık dağılımı CamShift yöntemi ile aranmaktadır. İlk olarak arama alanının sıfıncı momenti (Eşitlik 3.15), x ve y için birinci momentleri (Eşitlik 3.16, 3.17) bulunmaktadır. Aşağıdaki eşitliklerde görülen $P(x,y)$, arama alanındaki her bir pikselin olasılık dağılımını ifade etmektedir.

$$M_{00} = \sum_x \sum_y P(x, y) \quad (3.15)$$

$$M_{10} = \sum_x \sum_y x P(x, y) \quad (3.16)$$

$$M_{01} = \sum_x \sum_y y P(x, y) \quad (3.17)$$

Daha sonra yukarıdaki eşitliklerden bulunan momentlerden yararlanılarak hedef nesnenin merkezi bulunur (Eşitlik 3.18, 3.19).

$$x_c = \frac{M_{10}}{M_{00}} \quad (3.18)$$

$$y_c = \frac{M_{01}}{M_{00}} \quad (3.19)$$

Belirlenen iterasyon sayısı kadar ilerlenip her iterasyonda sıfıncı ve birinci momentler hesaplanır ve bu momentler kullanılarak arama alanının merkezi güncellenir. İterasyonlar bittikten sonra nesneyi temsil eden pencerenin boyutları (genişliği ve yüksekliği) Eşitlik 3.26, 3.27'den, arama alanı ise Eşitlik 3.28'dan yararlanılarak yeniden hesaplanır. Bu hesaplamalar için öncelikle ikinci momentler bulunur (Eşitlik 3.20, 3.21, 3.22).

$$M_{11} = \sum_x \sum_y x y P(x, y) \quad (3.20)$$

$$M_{20} = \sum_x \sum_y x^2 P(x, y) \quad (3.21)$$

$$M_{02} = \sum_x \sum_y y^2 P(x, y) \quad (3.22)$$

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad (3.23)$$

$$b = 2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right) \quad (3.24)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \quad (3.25)$$

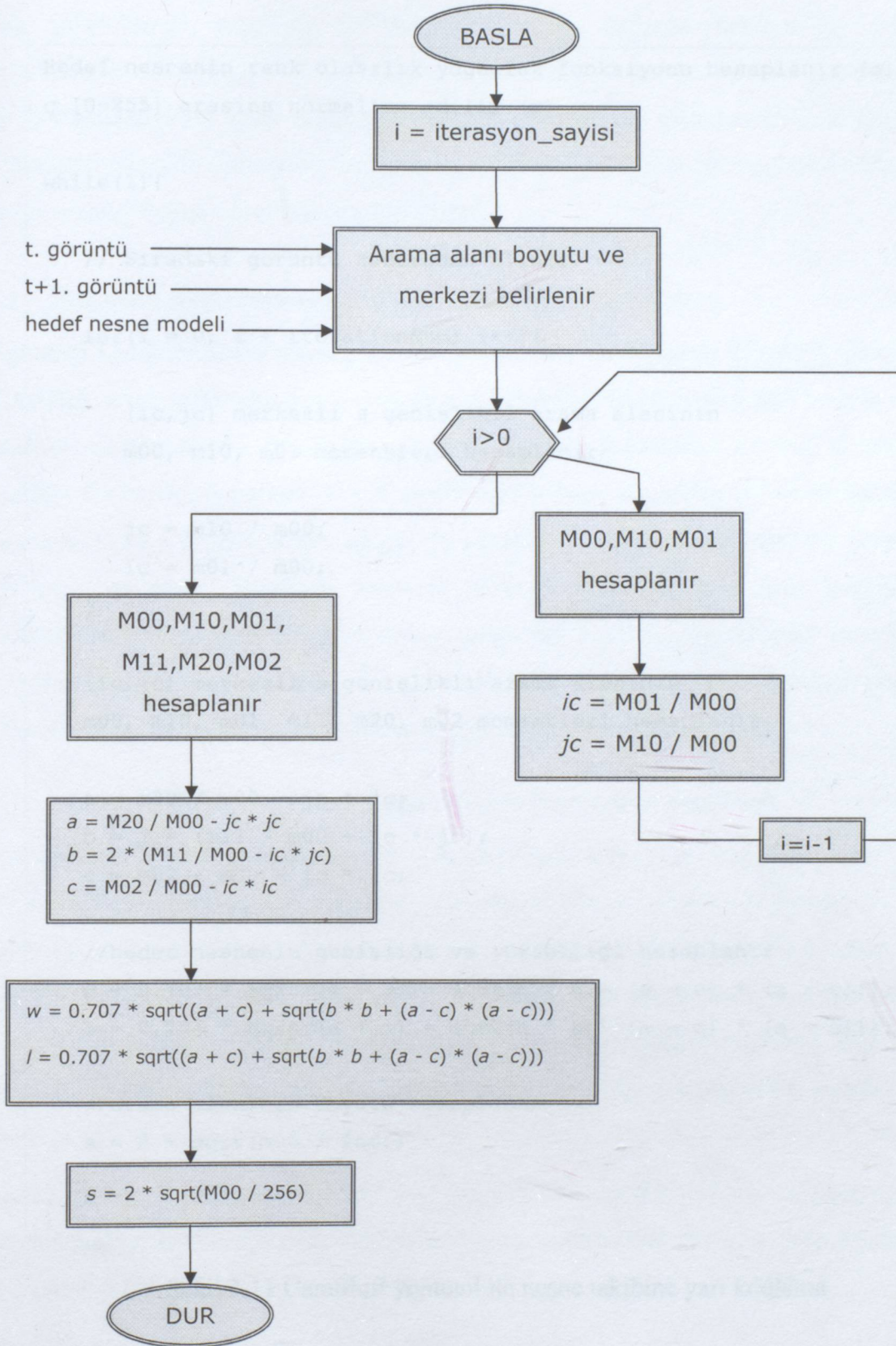
$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (3.26)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (3.27)$$

$$s = 2 \sqrt{\frac{M_{00}}{256}} \quad (3.28)$$

Hedef nesnenin geniş olan kenarını temsil etmek için l uzunluğu, kısa olan kenarını temsil etmek içinse w uzunluğu kullanılır. Eşit kenarlı bir nesnenin bir kenarını temsil etmek içinse l uzunluğu kullanılır. Arama alanının boyutu ise s ile temsil edilir. Bu hesaplamalar da yapıldıktan sonra nesnenin o anki görüntüdeki yeri, nesneyi temsil eden pencerenin boyutları ve arama alanının boyutları bulunmuş olur. Bu aşamada sıradaki görüntü işleme alınır. Yeni görüntüde arama alanının merkezi bir önceki adımda iterasyonlar sonucu bulunmuş olan merkez olarak güncellenir ve aynı adımlar tekrar edilir (Şekil 3.10, 3.11).

CamShift Yöntemi ile Nesne Takibi Akış Diyagramı:



Şekil 3.10 CamShift yöntemi ile nesne takibine ait akış diyagramı

CamShift ile Nesne Takibi Yarı Kodu:

```

Hedef nesnenin renk olasılık yoğunluk fonksiyonu hesaplanır (q)
q [0-255] arasına normalize edilir (p)

while(1){

    // Sıradaki görüntü kameradan alınır

    for(i = 0; i < iterationNum; i++){

        (ic,jc) merkezli s genişlikli arama alanının
        m00, m10, m01 momentleri hesaplanır

        jc = m10 / m00;
        ic = m01 / m00;
    }

    (ic,jc) merkezli s genişlikli arama alanının
    m00, m10, m01, m11, m20, m02 momentleri hesaplanır

    a = m20 / m00 - jc * jc;
    b = 2 * (m11 / m00 - ic * jc);
    c = m02 / m00 - ic * ic;

    //hedef nesnenin genişliği ve yüksekliği hesaplanır
    w = 0.707 * sqrt((a + c) - sqrt(b * b + (a - c) * (a - c)));
    l = 0.707 * sqrt((a + c) + sqrt(b * b + (a - c) * (a - c)));

    //arama alanının boyutu hesaplanır
    s = 2 * sqrt(m00 / 256);
}

```

Şekil 3.11 CamShift yöntemi ile nesne takibine yarı kodlama

4. UYGULAMA VE DENEYSEL SONUÇLAR

Tez çalışmasında, hareketli bir platform üzerinde bulunan kamera ile alınan renkli görüntülerin renk bilgisi değerlendirilerek hareketli nesnenin takibi gerçekleştirilmiştir. Geliştirilen hareketli nesne takibi sisteminin performansının değerlendirilmesi için uygulama aşamasında değişik senaryolar üretilmiş ve karşılaşılan problemler incelenerek bu durumlara karşı uygun çözümler üretilmeye çalışılmıştır.

4.1 Nesnenin Kameraya Olan Uzaklığının Değişmesi Durumu

Hareketli hedef nesnenin hareketi sırasında ortamdaki yeri sürekli değişiklik göstermektedir. Hareketli nesne kendi ekseninde ileri ve geri hareket edebildiği gibi sağa ve sola dönerek de hareket edebilmektedir. Hareketli nesnenin hareket doğrultusu kamera eksenine paralel olabilir. Bu hareketin paralel olması durumunda kamera tarafından alınan görüntülerde hedef nesnenin boyutu değişiklik göstermez. Bu durumda hareketli nesne sabit bir pencere boyutu ile takip edilebilir. Hareketli nesnenin hareketi esnasında kameraya yaklaşması veya uzaklaşması durumunda ise kameradan alınan iki boyutlu görüntülerde nesnenin boyutu büyümekte veya küçülmektedir. Bu durumda nesne takibi esnasında hareketli nesneyi takip etmekte kullanılan pencerenin boyutları dinamik olarak değiştirilmelidir.

4.1.1 Nesne Takibi Sırasında Uygun Pencere Boyutunun Seçilmesi

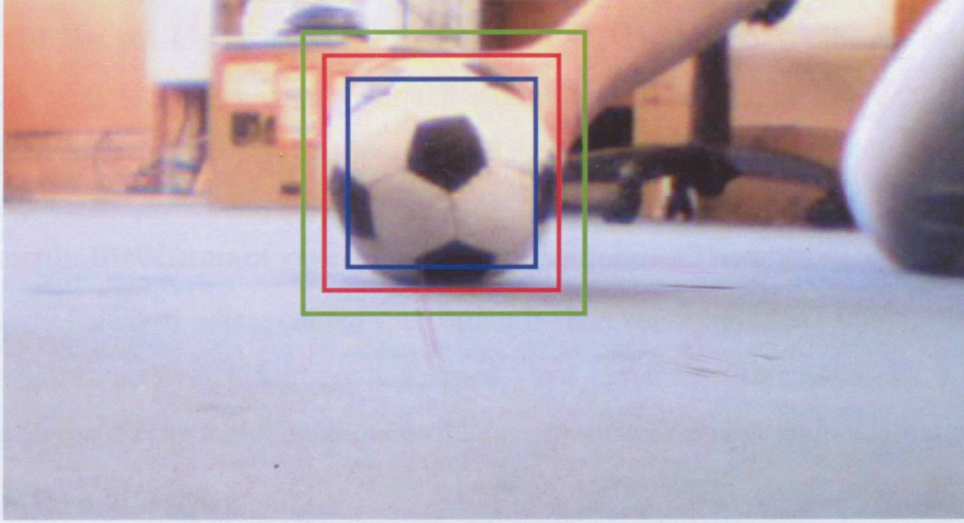
Nesne takibi, genel olarak hareketli nesneyi temsil edecek uygun pencere boyutunun seçilmesiyle ve nesnenin bu pencere içerisindeki olasılık yoğunluk fonksiyonu ile temsil edilmesiyle gerçekleştirilir. Hareketli nesnenin z-eksenindeki koordinatları değiştikçe, başka bir deyişle hareketli nesne kameraya yaklaştıkça veya kameradan uzaklaştıkça hareketli nesnenin görüntüdeki boyutu değişmektedir. Bu boyut değiştikçe nesnenin sabit boyutlu bir pencere ile takip edilmesi zorlaşmaktadır. Bu nedenle nesnenin boyutunun değişimine göre pencere boyutu da yeniden hesaplanmalıdır.

Nesne kameraya yaklaştıkça görüntüdeki boyutları büyümekte dolayısıyla pencere boyutu da büyütülmektedir. Nesne kameradan uzaklaştıkça görüntüdeki boyutları küçülmekte ve buna dayanarak da pencere boyutu küçültülmektedir.

Pencere Boyutunun Tespiti:

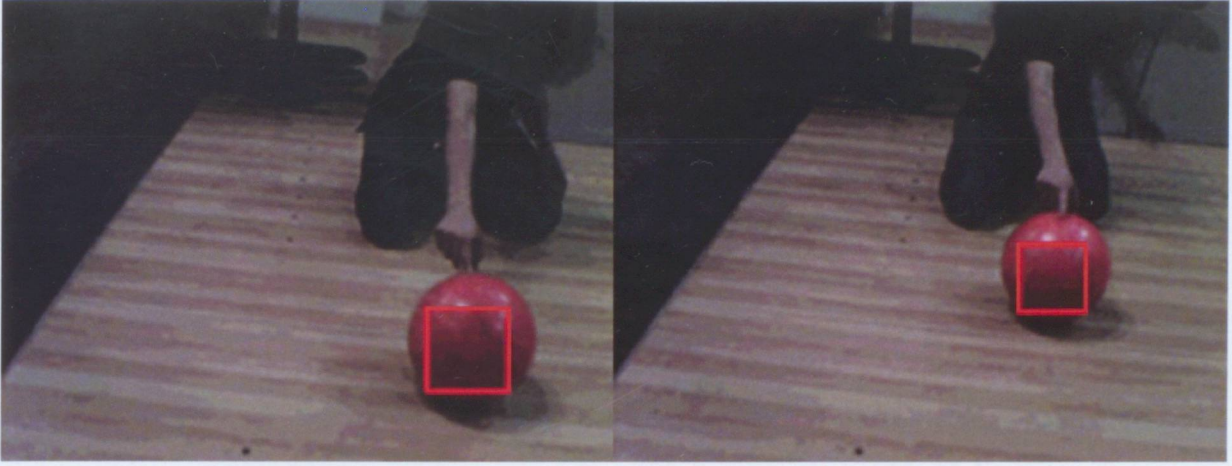
1. *Mean-Shift yöntemi ile nesne takibi yapan sistemde:* Hedef nesnenin bir önceki görüntüden gelen renk olasılık yoğunluk bilgisi mevcuttur. Bu bilgiye dayanarak yeni görüntüdeki arama alanının merkez koordinatları, belirlenen iterasyon sayısı kadar

işlem yapılarak hesaplanmaktadır. Hedef nesnenin yeni görüntüdeki merkezi iterasyonlar sonucu hesaplanan arama alanının merkez koordinatlarıyla aynıdır. Bu merkez koordinatları pencerenin merkezini oluşturacak şekilde, eski pencere boyutu ve bu pencerenin %10 küçüğü ve büyüğü alınarak üç adet aday renk olasılık yoğunluk dağılımı hesaplanır. Bu dağılımlar Bhattacharyya uzaklığı kullanılarak hedef nesnenin olasılık dağılımları ile karşılaştırılır. En fazla benzerliği veren dağılıma ait pencerenin boyutu yeni pencere boyutu olarak belirlenir. Nesne takibi bir sonraki adımda yeni pencere boyutu ile gerçekleştirilir. Şekil 4.1'de o anki pencere kırmızı renkle gösterilen penceredir. Bu pencere %10 büyütülerek yeşil pencere, %10 küçültülerek ise mavi pencere elde edilir. Daha sonra bu alanlardaki olasılık yoğunluk dağılımları bulunarak hedef nesnenin olasılık dağılımı ile maksimum benzerliği veren alan seçilir ve bu alana ait pencere boyutu da yeni pencere boyutu olarak belirlenir.



Şekil 4.1 Uygun pencere boyutunun belirlenmesi için değerlendirilen alanların gösterimi

2. *CamShift ile nesne takibi yapan sistemde:* Pencere boyutunun tespitinde kullanılabilecek daha efektif diğer bir yöntem ise CamShift yöntemi ile moment hesabı yapılarak ve bu moment hesabından yararlanılarak adaptif olarak pencere boyutunu hesaplamaktır. CamShift yönteminde, alınan görüntü üzerindeki arama alanının renk olasılık dağılımına momentler yardımıyla bakılmaktadır. Alınan her görüntüyle beraber arama alanının merkezi yeni hesaplanan momentler yardımı ile güncellenir (Eşitlik 3.18, 3.19). Arama alanının merkezi belirlendikten sonra sıfıncı, birinci ve ikinci momentler tekrardan hesaplanır (Eşitlik 3.15, 3.16, 3.17, 3.20, 3.21, 3.22). Bu momentlerden yararlanılarak da hedef nesnenin ve arama alanının boyutları hesaplanır (Eşitlik 3.26, 3.27, 3.28).



Şekil 4.2 Pencere boyutunun güncellenmesi

Nesne takibi yapılırken her görüntüde hedef nesneyi temsil eden pencerenin boyutu yeniden hesaplanmaktadır. Bu sayede hedef nesnenin kameraya olan uzaklığından bağımsız olarak nesne başarılı bir şekilde takip edilmektedir. Şekil 4.2’de görüldüğü gibi sol resimdeki top daha büyük yani kameraya daha yakinken sağ resimdeki top kameraya daha uzaktır. Bu durumda soldaki resimdeki pencere boyutu sağdaki resimdekine göre daha büyüktür.

4.2 Nesnenin Bir Kısımının veya Tamamının Kaybolması Durumu

Takip edilen hareketli nesne doğal bir ortamda hareket ettiğinden dolayı, hedef nesne kameranın görüş açısıyla bakıldığında kısmen veya tamamen ortamda başka bir nesnenin arkasına girebilir. Nesne takipçisinin bu problemi aşip hedef nesneyi takip edebilmesi gerekir.

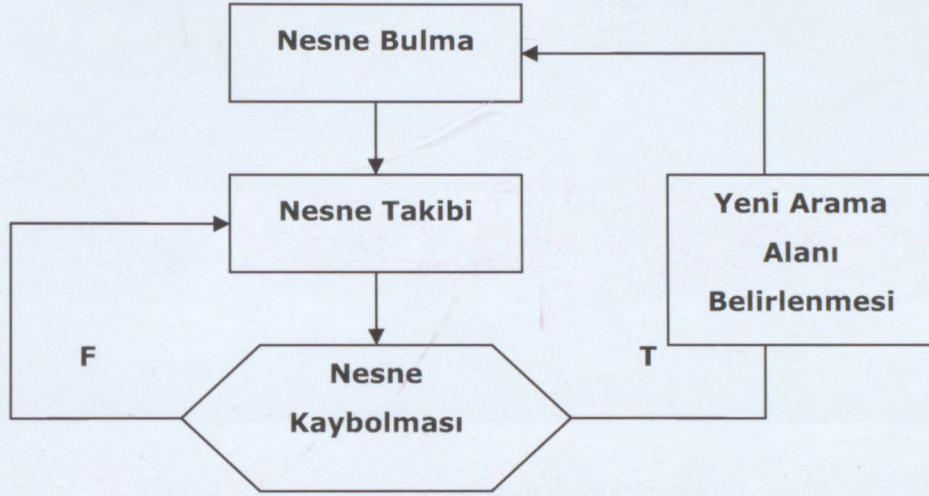
Bu problem iki şekilde çözülebilir:

1. Hedef nesne başka bir nesnenin arkasına geçtiğinde hedef nesnenin bir kısmının kaybolduğu kabul edilirse bu durumda hedef nesneyi temsil eden dikdörtgen bölgenin (çekirdek) boyutları hedef nesnenin görünen kısmının büyüklüğü CamShift yöntemi ile adaptif bir şekilde değiştirilir ve yine CamShift yöntemi kullanılarak nesne takibi yapılır.

Yine nesnenin bir kısmının kaybolması durumunda, nesnenin arandığı arama alanı büyütülerek nesne bulma algoritmasına geri dönülebilir. Hedef nesnenin başta bilinen özellikleri, yani nesnenin dikdörtgen bölge ile temsil edilen alanına ait renk olasılık yoğunluk fonksiyonu kullanılarak hedef nesne yeni belirlenen arama alanında aranmaya başlanır. Hedef nesne bulunduğunda, nesnenin yeni yer bilgisi nesne takibi aşamasına gönderilir ve nesne takibine yeniden başlanır.

2. Hedef nesnenin tamamının kaybolması durumunda ise hedef nesnenin renk olasılık

yoğunluk fonksiyonu kullanılarak hedef nesnenin kaybolduğu alanda arama alanı genişletilerek hedef nesne aranmaya başlanır. Hedef nesne bulunduğunda, nesnenin bulunduğu yeni koordinatlar nesne takibi aşamasına aktarılarak nesne takibine yeniden başlanır (Şekil 4.3).



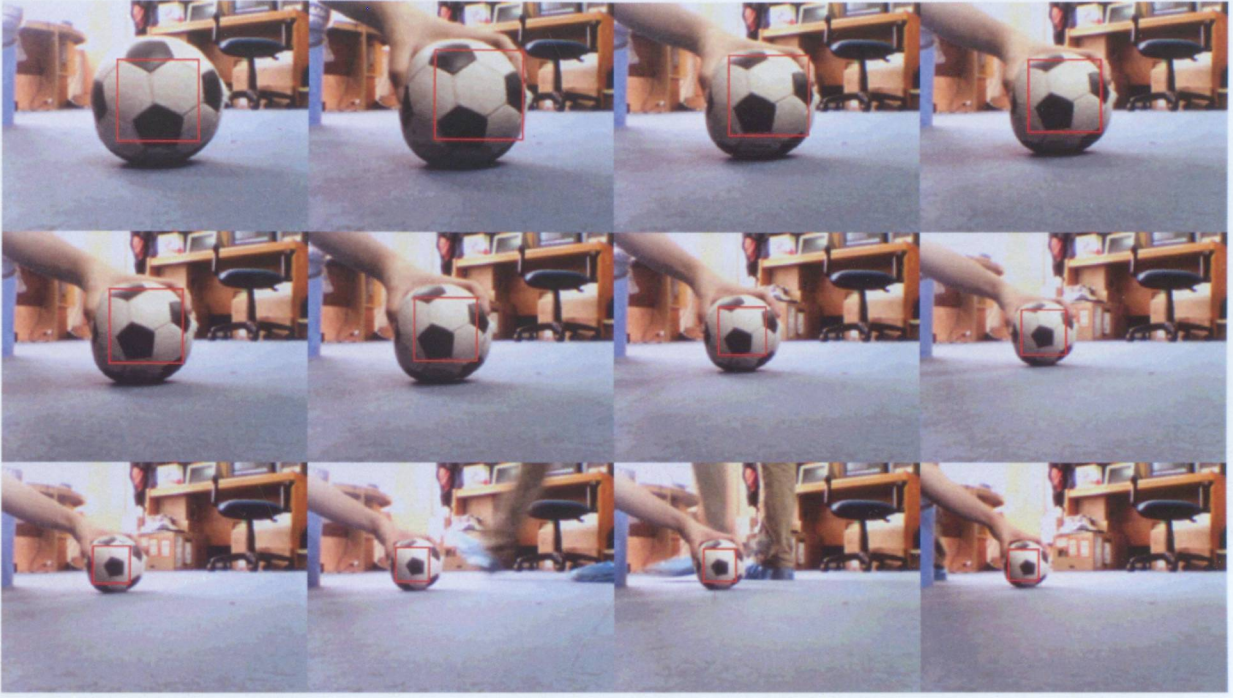
Şekil 4.3 Hedef nesnenin robotun görüş açısından çıkması durumuna ait akış diyagramı

4.3 Sabit Kamera İle Hareketli Görüntü Takibi

Şekil 4.5'te hedef nesnenin kameradan uzaklaştığı durumda takibi görülmektedir. Şekil 4.4'te ileriki görüntülerde de bahsedilen takip edilmesi hedeflenen futbol topu görülmektedir.

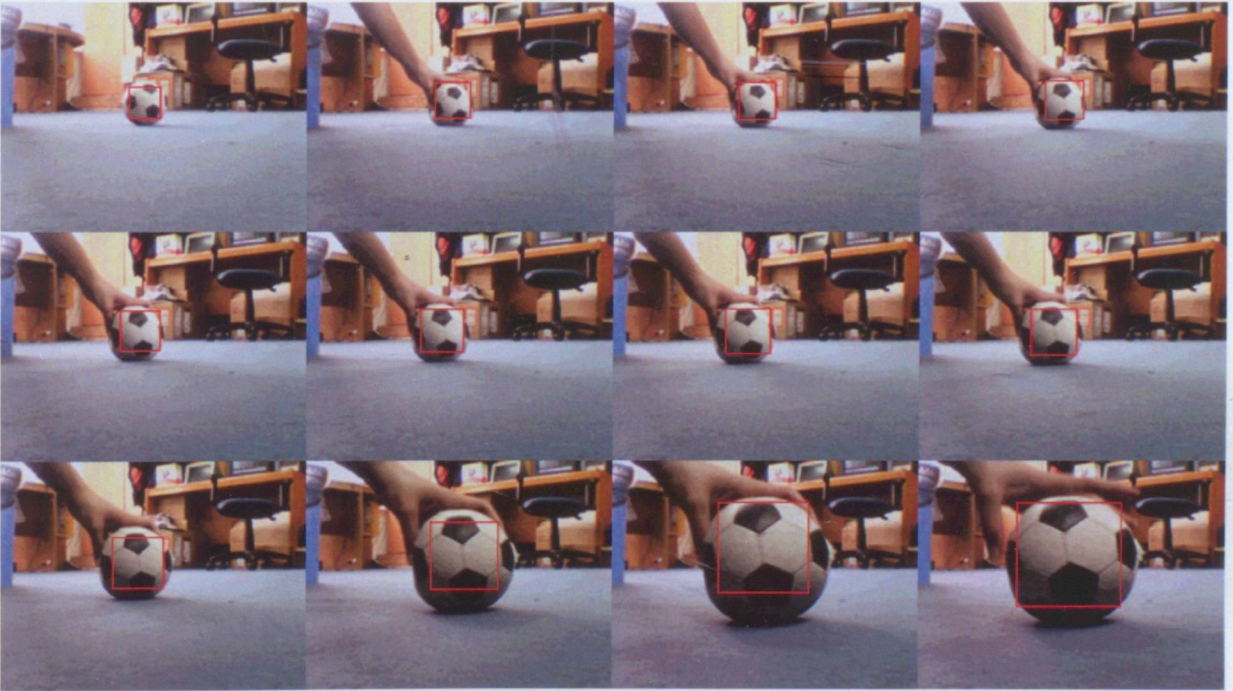


Şekil 4.4 Hedef nesne



Şekil 4.5 Kameradan uzaklaşan nesnenin takibi

Şekil 4.6’da hedef nesnenin kameraya yaklaştığı durumda takibi görülmektedir.

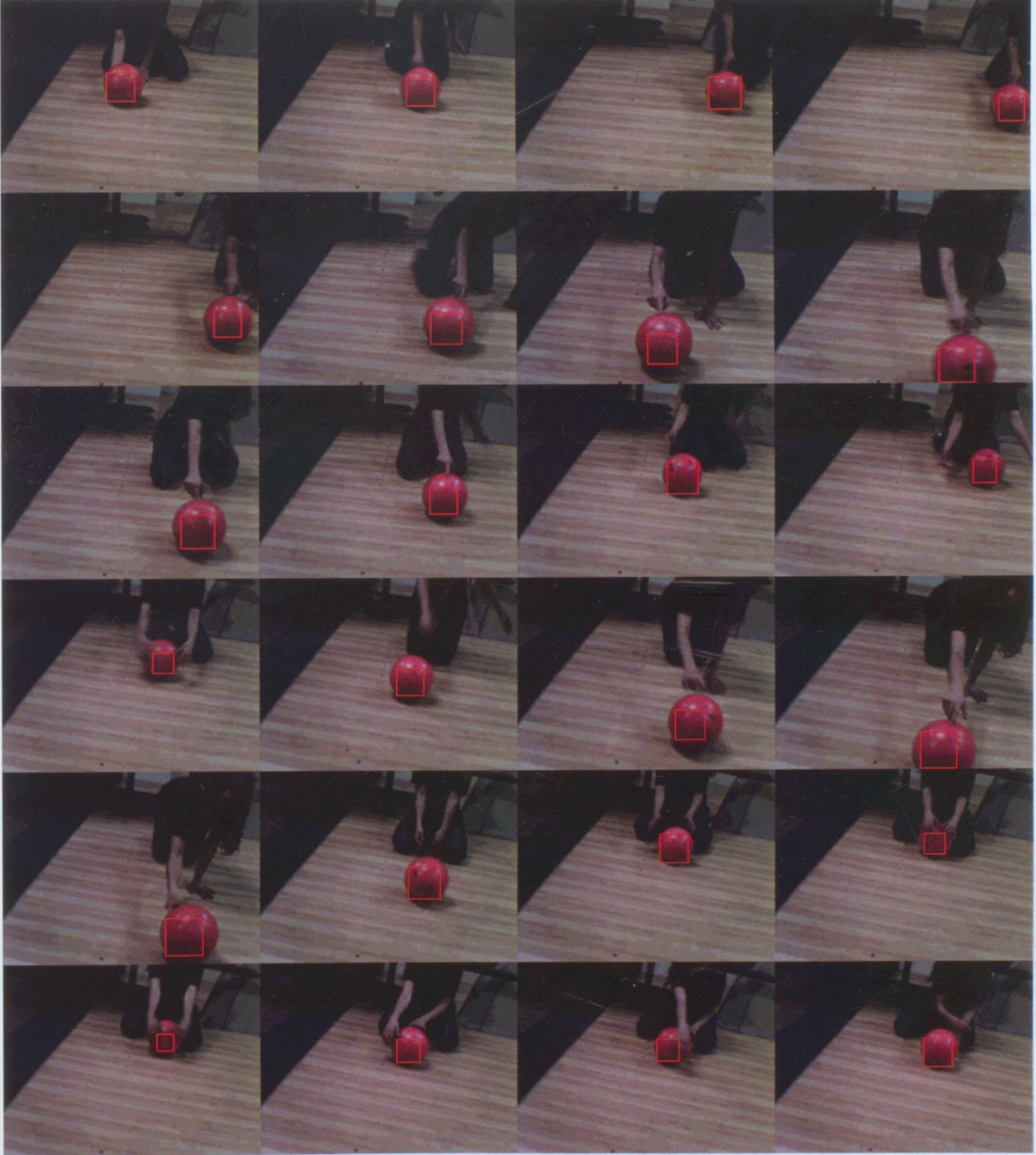


Şekil 4.6 Kameraya yaklaşan nesnenin takibi

Şekil 4.8 ve 4.9’da hareketli bir topun ileri, geri, sağa ve sola hareket ederken önüne herhangi bir cisim geçtiği durumlarda takibi görülmektedir.



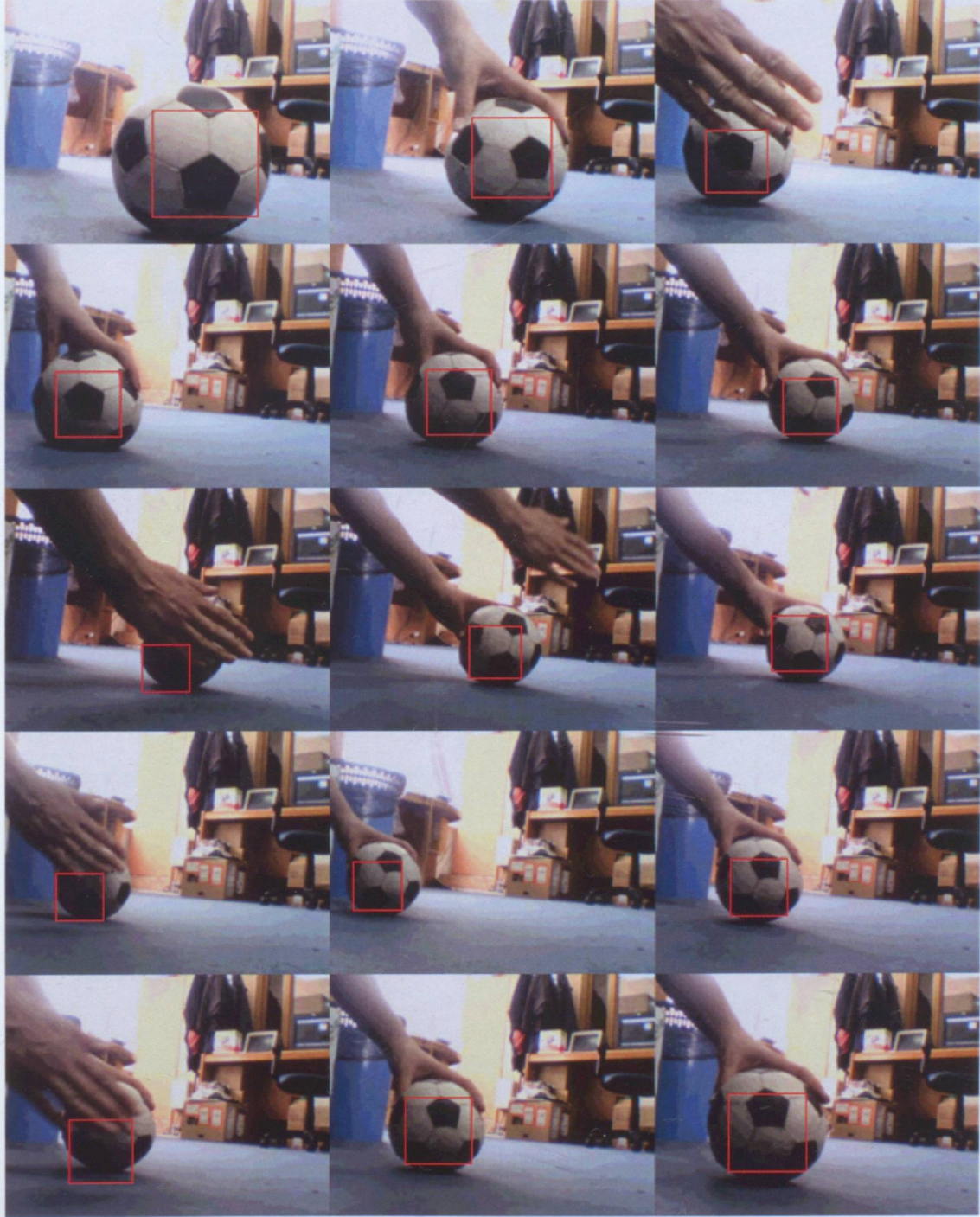
Şekil 4.7 Hedef nesnenin görüntüsü



Şekil 4.8 Bowling topunun takibi

Yukarıdaki görüntülerden de görüldüğü üzere takip edilen cismin dış cephesi parlak bir maddeyle çevrilidir. Bu sebepten ötürü nesne gelen ışığı yansıtmakta ve nesnenin tepesinde ufak da olsa aydınlanmalar, renk farklılaşmaları görülmektedir. Hedef nesne başarıyla takip

edilmesine rağmen nesneyi temsil eden dikdörtgen bazı görüntülerde nesnenin tamamını kaplayamamakta, nesnenin parlak yüzeyi nesnenin bir parçası olarak belirlenememektedir.



Şekil 4.9 Futbol topunun takibi

Şekil 4.11’de bütün bir şekle sahip olmayan, hareket yapısı düzenli olamayan robotun (Robosapien V2) takibi görülmektedir. Sistem çalışmaya başlamadan önce, robotun şekil yapısı sebebiyle sisteme girdi olarak verilen görüntüsünün tamamı robottan oluşmamaktadır (Şekil 4.10). Verilen görüntünün %65’i robot iken %35’i arka plandan oluşmaktadır. Bu da

robotun takip başarısını düşürmektedir.



Şekil 4.10 Hedef nesnenin görüntüsü



Şekil 4.11 Hareketli robotun takibi

Sistemde, hedef nesne sahneye girene kadar nesne görüntü üzerinde aranmaktadır. Nesne bulma aşamasıyla nesnenin yerinin tespit edilmesi ile nesnenin yer bilgisini girdi olarak alan nesne takibi aşaması başlamaktadır. Şekil 4.13’de sahneye sonradan dahil olan hedef nesnenin takibi görülmektedir.



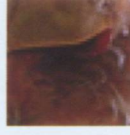
Şekil 4.12 Hedef nesnenin görüntüsü



Şekil 4.13 Hareketli nesnenin sahneye sonradan girmesi durumunda takibi

4.4 Hareketli Kamera İle Hareketli Görüntü Takibi

Şekil 4.15’de hareketli kamera ile hareketli hedef nesnenin takibi görülmektedir. Hedef nesnenin tanıtımı için sisteme verilen ilk görüntü ise 4.14’te görülmektedir.



Şekil 4.14 Hedef nesnenin görüntüsü



Şekil 4.15 Hareketli kamera ile hedef nesnenin takibi

4.5 Hızlı Bir Hedef Nesnenin Takibi

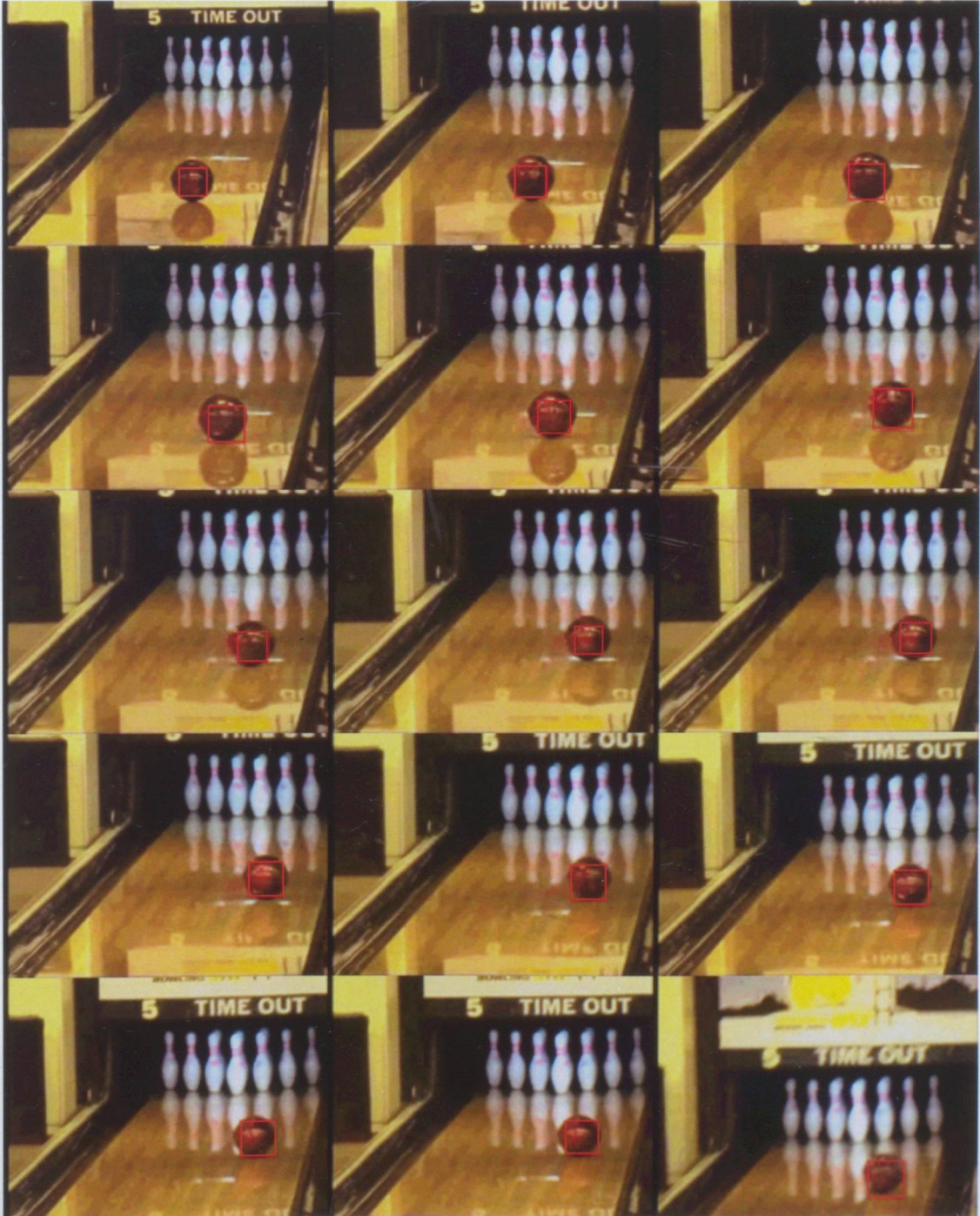
Şekil 4.16'da ve 4.17'de hızlı hedef nesnelerin takibi görülmektedir.

Şekil 4.16'da görülen nesne takibinde, hedef nesnenin yeri başarılı olarak belirlenebilmesine rağmen nesnenin tahmin edilen boyutlarında hatalar ortaya çıkmıştır. Bu hatalar; hedef nesnenin çok küçük olmasından, videonun görüntü kalitesinin düşüklüğünden ve en önemlisi hedef nesnenin hızlı hareketinden doğan şekil bozukluğundan kaynaklanmaktadır.



Şekil 4.16 Hızlı hedef nesnenin takibi

Şekil 4.17’de, saatte ortalama 48 km hızla hareket eden bir bowling topunun takibinde elde edilen sonuçlar görülmektedir. İlgili hedef nesnenin hızının çok yüksek olması sebebiyle video çekimi yapılırken saniyede alınan görüntü sayısı 10’dan 25’e çıkarılmıştır. Bu sayede nesnenin, kameranın görüş açısından çıkma olasılığı azaltılmış olur. Başka bir deyişle hedef nesnenin ardışık görüntülerdeki yer değişim farkı en aza indirgenmiş olur. Bu sayede hedef nesnenin hızının yüksek olmasına rağmen, nesnenin görüntü üzerindeki yeri ve kapladığı alan başarıyla belirlenmiştir.



Şekil 4.17 Hızlı hedef nesnenin takibi

4.6 Birden Fazla Nesnenin Takibi

Şekil 4.18’de farklı renk dağılımına sahip iki ayrı nesnenin takibi görülmektedir. Her bir nesne için ayrı arama alanı belirlenmektedir. Her iki nesnenin sıradaki görüntüdeki merkezleri ve nesneleri temsil eden pencere boyutları ayrı ayrı hesaplanmaktadır.



Şekil 4.18 Hareketli iki nesnenin takibi

Bu çalışmada, AMD Turion 1.59 GHz işlemcili bilgisayar kullanılarak çeşitli videolar üzerinde hareketli nesne takibi yapılmış ve sistemin çalışma süreleri gözlenmiştir. Aşağıda farklı video örnekleri üzerinden elde edilmiş nesne takibi çalışma süresi sonuçları görülmektedir (Çizelge 4.1).

Çizelge 4.1 Nesne takibi sonuçları

Örnek	Nesne Bulma Süresi (s)	Nesne Takibi Süresi (s)	Saniyede Alınan Görüntü Sayısı	Video Süresi	Arama Alanı	görüntü / s
1	0.4	39	10	46	88	11.795
2	0.5	28.5	10	20	141	7.018
3	0.5	57.2	10	34	173	5.944
4	0.4	67	10	31	296	4.627

Çizelge 4.1’de de görüldüğü gibi arama alanı genişledikçe saniyede takip edilen görüntü sayısı (görüntü/s) azalmaktadır. Bunun sebebi nesnenin görüntüde arandığı alan genişledikçe yapılan işlemlerin artmasıdır.

5. SONUÇ

Bu yüksek lisans tezi kapsamında hareketli kameradan alınan görüntüler üzerinden gerçek zamanlı hareketli nesne takibi gerçekleştirilmiştir. Kamera hareketli bir düzencek üzerine yerleştirilmiştir. Sistem aynı anda hareketli kamera ve hareketli hedef nesne üzerinde çalışabilmekte ve hedef nesneyi bu koşullarda da başarıyla takip edebilmektedir.

Sistem iki temel adımdan oluşmaktadır: hareketli hedef nesnenin bulunması ve takibi. Sistemin çalışması nesne bulma aşamasıyla başlamaktadır. Hedef nesne sahneye giriş yapana kadar kameradan alınan görüntüler gerçek zamanlı işlenmekte, hedef nesne bu görüntüler üzerinde aranmaktadır.

Hedef nesneyi görüntü üzerinde tespit etmek amacıyla CamShift yöntemi kullanılmıştır. Hedef nesne, görüntü bilgisi sisteme çalışma zamanından önceden aktarılmış olan herhangi bir hareketli nesne olabilir. Hedef nesneyi bulma amacıyla kullanılan özellik nesnenin renk olasılık dağılımıdır. Hareketli kameradan alınan ardışık görüntülerin her birinin renk olasılık dağılımı hedef nesnenin renk olasılık dağılımından yararlanılarak çıkartılmakta ve hedef nesnenin yeri görüntü üzerinde bulunmaktadır. Bu amaç doğrultusunda çalışılan görüntü üzerindeki bütün piksellerin momentleri renk olasılık ve yer bilgisi kullanılarak hesaplanmaktadır. Hesaplanan bu momentler ışığında, hedef nesnenin görüntü üzerinde bulunduğu aday bölge belirlenir. Bu aday bölge ile hedef nesnenin renk olasılık dağılım fonksiyonları, Bhattacharyya uzaklığı ile karşılaştırılır. Elde edilen benzerlik sonucu, önceden belirlenmiş sabit eşik seviyesinin üzerindeyse hedef nesnenin bulunmuş olduğuna karar verilir. Nesnenin elde edilen yer bilgisi nesne takibi aşamasına aktarılır ve nesne takibine başlanır. Eğer elde edilen benzerlik sonucu, eşik seviyesi değerinin altındaysa bu sonuç eşik seviyesi değerinin üzerine çıkana kadar nesne bulma aşamasına devam edilir.

Nesne takibi aşaması, nesne bulma aşamasından gelen hedef nesnenin bir önceki görüntüdeki yer bilgisini ve hedef nesnenin renk olasılık dağılım bilgisini kullanarak nesne takibine başlamaktadır. Nesne takibi aşamasında gerçek zamanlı olarak alınan ardışık görüntüler üzerinde anlık sonuçlar üretilmektedir. Hedef nesne, gerçek zamanlı olarak alınan her bir ardışık görüntü üzerinde CamShift yöntemi ile aranmaktadır. O anki görüntü üzerinde hedef nesnenin aranacağı alanın merkezi bir önceki görüntü üzerinde nesnenin bulunmuş olduğu yerin merkezi olacak şekilde belirlenir. Arama alanının büyüklüğü ise piksellerin renk olasılığı ve piksellerin yer bilgisi kullanılarak hesaplanan momentler üzerinden bulunmaktadır. Hedef nesne, büyüklüğü ve yeri belirlenen arama alanı içinde aranmaktadır. Arama işlemi yine CamShift yönteminde yer alan moment hesabı kullanılarak yapılmaktadır.

Bu momentlerden yararlanılarak arama alanında hedef nesnenin renk olasılık dağılımının en yoğun olduğu bölge bulunmaktadır ve hedef nesnenin yeri bu bölgenin merkezi olarak belirlenmektedir. Yine momentlerden yararlanılarak hedef nesnenin görüntü üzerinde kapladığı alan bulunmaktadır. Hedef nesnenin görüntü üzerindeki yer bilgisi, nesnenin görüntüde kapladığı alan ve arama alanının boyutu; bir sonraki nesne takibi aşamasına aktarılır. Görüntü alımı devam ettiği sürece nesne takibi adımları da bu şekilde devam etmektedir.

Bu tez kapsamında hareketli bir platform üzerine yerleştirilen 90^0 görme açısına sahip, renkli video kameradan alınan 640×480 çözünürlüğündeki görüntüler üzerinde gerçek zamanlı hareketli nesne takibi yapılmıştır. Video çekimi, doğal ışık ve yapay ışıkla aydınlatılmış iç mekanlarda yapılmıştır.

Hareketli nesnenin hareketli kamera ile gerçek zamanlı takibinde sistemin başarısını etkileyen bazı dış faktörler vardır:

1. Hedef nesnenin parlak bir cisim olması ve ortamdaki ışığın fazla olması durumunda hedef nesne üzerine gelen ışığı yansıtmakta ve bilinen renk olasılık dağılımı ile tamamen benzeşmemektedir. Bu durumda hedef nesnenin tamamı yerine parlama yapmayan diğer kısımları hedef olarak işaretlenmelidir. Bu duruma çözüm olarak hedef nesnenin değişik ışık seviyeli ortamlarda birden fazla görüntüsü nesne takibi aşamasına girdi olarak verilebilir.
2. Kamera kalitesinden, ortamdaki çok fazla ışık değişiminden veya ortamdaki ışığın yetersizliğinden dolayı bazı çekimlerde hedef nesnenin takibi sırasında takipçi tarafından gözden kaçırıldığı gözlemlenmiştir.
3. Hedefin top, kitap gibi bütünsel bir nesne olmadığı durumlarda, yani insan, robot gibi karmaşık bir yapıya sahip nesne olduğunda nesne takibine girdi olarak verilen hedef nesnenin görüntüsü gereksiz bilgi içermekte ve bu gereksiz bilgi de nesne takibi başarısını düşürebilmektedir. Bu problemi çözmek için, girdi olarak verilen hedef nesnenin görüntüsünden arka plan kısmı elenebilir.
4. Takip edilen hareketli nesnenin hareketinin çok hızlı olması durumunda nesne gözden kaçabilmektedir. Bu durumda arama alanını geniş tutmak takip başarısını arttırmakla beraber sistemin çalışma hızını düşürecektir.

KAYNAKLAR

- Alan M. McIvor, (2000), "Background Subtraction Techniques", In Proc. of Image and Vision Computing, Auckland, New Zealand.
- Alper Yilmaz, O. Javed, M. Shah, (2006), "Object tracking: a survey", ACM Computing Surveys 38(4): 13-es.
- Alper Yilmaz, X. Li, M. Shah, (2004), "Object Contour Tracking Using Level Sets", Asian Conference on Computer Vision, ACCV 2004, Jaju Islands, Korea.
- Alper Yilmaz, 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.
- Bogdan Kwolek, (2005), "CamShift-Based Tracking in Joint Color-Spatial Spaces", CAIP 2005, LNCS 3691, pp 693-700.
- Burkay Birant Örtten, (2005), "Güvenlik Amaçlı Video Sistemlerinde Hareketli Nesnelerin Tanınması ve Olay Analizi", Yüksek Lisans Tezi, Tez Yöneticisi: Doç. Dr. A. Aydın Alatan.
- Dorin Comaniciu, P. Meer, (2002) "Mean-Shift: A Robust Approach Toward Feature Space Analysis", IEEE, Vol 24, No 5.
- Dorin Comaniciu, V. Ramesh ve P. Meer, (2000), "Real-Time Tracking of Non-Rigid Objects using Mean Shift", Computer Vision and Pattern Recognition, 142-149.
- Dorin Comaniciu, V. Ramesh ve P. Meer, (2003), "Kernel-Based Object Tracking", IEEE Trans. Pattern Analysis and Machine Intelligence, May 2003, vol. 25, no. 5, pp. 564-575.
- Gary R. Bradski, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation, "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal Q2 '98.
- Hanan Samet ve Markku Tamminen, (1988), "Efficient Component Labeling of Images of Arbitrary Dimension Represented by Linear Bintree", IEEE Trans. Pattern Anal. Mach. Intell.
- Ilhan Kubilay Yalçın ve Muhittin Gökmen, (2005), "License Plate Tracking from Monocular Camera View by Condensation Algorithm", ICIC 2005, Part II, LNCS 3645, pp. 860-869.
- Isaac Cohen ve Gerard Medioni, (1998), "Detecting and Tracking Moving Objects in Video from an Airborne Observer", Image Understanding Workshop, 217-222.
- Isard M., Blake A., (1998), CONDENSATION - Conditional Density Propagation for Visual Tracking. Int. J. Computer Vision, vol. 29 no. 1 5-28.
- J. F. Canny, (1986), "A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence", pages 679-698.
- Kalman R. E., (1960), "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME - Journal of Basic Engineering, pp. 35-45.
- Kye Kyung Kim, Soo Hyun Cho, Hae Jin Kim, Jae Yeon Lee, (2005), "Detecting and Tracking Moving Object Using an Active Camera", Digital Object Identifier 10.1109/ICACT, 817-820.
- Neil Alldrin, "Conditional Density Propagation for Visual Tracking".
- Oya Aran, Lale Akarun, (2008) "Etkileşimli Parçacık Süzgeci Yöntemi ile Kapatmaya Dayanıklı Yüz ve El Takibi", SIU2008.

Pablo Barrera, Jose M. Canas, Vicente Matellan, (2005), "Visual Object Tracking in 3D with Color Based Particle Filter", International Journal of Information Technology Volume 2 Number 1.

Rudolf K. Bock, (1998), Bit-Cutting.

Qi Zang, Reinhard Klette, (2003), "Object Classification and Tracking in Video Surveillance", CAIP, 198-205.

Shi, J. ve Malik, J. 2000. Normalized cuts and image segmentation. IEEE Trans. Patt. Analy. Mach.Intell. 22, 8, 888-905.

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.

Takashi Morimoto, Osamu Kiriya, Youmei Harada, Hidekazu Adachi, Tetsushi Koide, Hans Jürgen Mattausch, (2005), "Object Tracking in Video Pictures based on Image Segmentation and Pattern Matching", IEEE, 3215-3218.

Tetsushi Koide, Hans Jürgen Mattausch, Takashi Morimoto, Hidekazu Adachi , Kosuke Yamaoka, (2005) "Real-Time Multi-Object Tracking Based on Highly Parallel Image Segmentation and Pattern Matching".

Yaron Ukrainitz ve Bernard Sarel, "Mean Shift Theory and Applications".

Yigithan Dedeoglu, (2004), "Akıllı Video Gözetimi İçin Hareketli Nesne Bulma, Takip Etme Ve Sınıflandırma", Tez Yöneticisi: Yrd. Doc. Dr. Uğur GÜDÜKBAY.

INTERNET KAYNAKLARI

http://en.wikipedia.org/wiki/Condensation_algorithm

<http://pages.cpsc.ucalgary.ca/~annie/PeopleVisWebsite/Gregor/gregor.htm>

http://web.missouri.edu/~hantx/ECE8001/notes/lecture8_mean_shift.pdf

http://www.cis.upenn.edu/~jshi/ped_html/

http://www.cs.otago.ac.nz/gpxpriv/vision_optflow.html

ÖZGEÇMİŞ

Doğum tarihi 06.07.1983

Doğum yeri İstanbul

Lise 1998-2001 Fahrettin Kerim Gökay Anadolu Lisesi

Lisans 2001-2005 Yıldız Üniversitesi Elektrik Elektronik Fak.
Bilgisayar Mühendisliği Bölümü

Yüksek Lisans 2005- Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Müh. Anabilim Dalı

Çalıştığı kurum(lar)

2005- Devam ediyor YTÜ Bilgisayar Müh. Bölümü Araştırma Görevlisi

