

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**UYARLAMALI KALMAN FİLTRESİ DESTEKLİ ORTALAMA  
KAYMA TABANLI NESNE TAKİBİ**

**Mehmet Murat TURHAN**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**NİSAN 2016**

**Tezin Bařlıđı :** Uyarlamalı Kalman Filtresi Destekli Ortalama Kayma Tabanlı Nesne Takibi

**Tezi Hazırlayan :** Mehmet Murat TURHAN

**Sınav Tarihi :** 20.04.2016

Yukarıda adı geen tez jürimizce deđerlendirilerek Bilgisayar Mühendisliđi Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

### **Sınav Jüri Üyeleri**

**Tez Danıřmanı : Do.Dr. Davut HANBAY** .....  
İnönü Üniversitesi

**Do.Dr. Fatih TALU** .....  
İnönü Üniversitesi

**Yrd.Do.Dr. Ahmet INAR** .....  
Fırat Üniversitesi

**Prof.Dr. Alaattin ESEN**  
Enstitü Müdürü

## ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduđum “Uyarlamalı Kalman Filtresi Destekli Ortalama Kayma Tabanlı Nesne Takibi” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakça yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Mehmet Murat TURHAN

## ÖZET

Yüksek Lisans Tezi

### UYARLAMALI KALMAN FİLTRESİ DESTEKLİ ORTALAMA KAYMA TABANLI NESNE TAKİBİ

Mehmet Murat TURHAN

İnönü Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

65+xi sayfa

2016

Danışman: Doç.Dr. Davut HANBAY

Bu tezde, ortalama kayma algoritması ve uyarlamalı Kalman filtresi birlikte kullanılarak görüntü dizilerinde nesne takibi gerçekleştirilmiştir.

Ortalama kayma algoritması takip edilen nesnenin ardışık iki görüntü arasında hızlı yer değiştirmesi veya nesnenin başka nesnelere tarafından engellenmesi gibi durumlarda iyi sonuçlar verememektedir. Yapılan çalışmada, takip edilen nesnenin aranacağı başlangıç konumu Kalman filtresi tarafından tahmin edilir ve ortalama kayma algoritması nesneyi bu konumda aramaya başlar. Ortalama kayma algoritmasından elde edilen Bhattacharyya katsayısı, Kalman filtresinin ölçüm hata kovaryans matrisini anlık güncellemede ve nesnenin engelleme maruz kalmadığına karar vermede kullanılır.

Deneysel sonuçlar önerilen yöntemin nesnenin engellenme ve hızlı hareket etme durumları için standart ortalama kayma algoritmasına kıyasla daha etkili olduğunu göstermektedir.

**ANAHTAR KELİMELER:** Nesne Takibi, Ortalama Kayma Algoritması, Uyarlamalı Kalman Filtresi

# **ABSTRACT**

Master Thesis

## **MEAN SHIFT BASED OBJECT TRACKING SUPPORTED BY ADAPTIVE KALMAN FILTER**

Mehmet Murat TURHAN

İnönü University  
Graduate School of Natural and Applied Sciences  
Department of Computer Engineering

65+xi pages

2016

Supervisor: Assoc.Prof.Dr. Davut HANBAY

In this thesis, object tracking in video sequences is implemented by using both mean shift algorithm and adaptive Kalman filter.

Mean shift algorithm cannot give good results when the position of the tracked object is changed rapidly between subsequent two frames or the tracked object is occluded. In this study, initial position to search the tracked object is predicted by Kalman filter and then the mean shift algorithm begins to search the object in this position. Bhattacharyya coefficient, which is obtained from mean shift algorithm, is used to instantly update Kalman filters error covariance matrix and determine whether object is occluded or not.

Experimental results demonstrate that the proposed method has been more efficient technique as compared to standard mean shift algorithm in case of occlusion and fast object tracking.

**KEYWORDS:** Object Tracking, Mean Shift Algorithm, Adaptive Kalman Filter

## **TEŐEKKÜR**

Eđitim hayatım boyunca üzerimde emeđi bulunan bütun hocalarıma, tez danışmanıma ve tezimin düzenlenmesi aşamasında yardımlarını esirgemeyen Doç.Dr. Celaleddin YEROĐLU, Arş.Gör. Abdullah ATEŐ'e ve Yrd.Doç.Dr. Barıő Baykant ALAGÖZ'e teşekkür ederim.

## İÇİNDEKİLER

ÖZET .....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ.....	vi
TABLOLAR DİZİNİ.....	x
SİMGELER VE KISALTMALAR .....	xi
1. GİRİŞ.....	1
1.1. Hedef Nesnenin Kapladığı Bölgenin Temsili.....	2
1.2. Hedef Nesnenin Görünümünün Temsili.....	3
2. ORTALAMA KAYMA ALGORİTMASI.....	5
2.1. Ortalama Kayma Algoritması.....	5
2.2. Ortalama Kayma Algoritmasının Nesne Takibinde Kullanılması.....	8
2.2.1 Hedef Nesnenin Temsili.....	8
2.2.2 Aday Nesnenin Temsili.....	9
2.2.3 Takip İşlemi.....	10
3. KALMAN FİLTRESİ VE ÖNERİLEN YÖNTEM.....	17
3.1. Kalman Filtresi.....	17
3.2. Önerilen Yöntem.....	19
4. UYGULAMA.....	23
4.1 Ortalama Kayma Algoritmasının Nesne Takibinde Kullanılması.....	25
4.1.1. Panda Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı...	25
4.1.2. Person Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı..	25
4.1.3. Person Partially Occluded Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı.....	26
4.1.4. Rubik's Cube Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı.....	26
4.1.5. Cup Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı.....	26
4.2 Önerilen Yöntem ve Ortala Kayma Algoritmasının Nesne Takibinde Kullanılması.....	27
4.2.1. Ball Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı.....	27

4.2.2.	Juice Box Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı.....	29
4.2.3.	PersonFloorSequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı.....	29
4.2.4.	Person Fully Occluded Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı.....	30
5.	SONUÇ.....	61
6.	KAYNAKLAR.....	62



## ŞEKİLLER DİZİNİ

Şekil 1.1.	Nesne takip algoritmalarının 5 önemli bileşeni.....	2
Şekil 1.2.	Nesne takibinde hedef bölgenin temsili.....	3
Şekil 1.3.	Nesne görünümüne ait özelliklerin temsili a) 2 boyutlu parlaklık dizisi, b) histogram c) özellik vektörü .....	4
Şekil 2.1.	Ortama kayma algoritmasının yinelemeli çalışıp yoğun bölgenin bulunması.....	7
Şekil 2.2.	Ardışık iki çerçeve için ortalama kayma algoritmasının çalışma prensibini gösteren akış diyagramı .....	13
Şekil 2.3.	Ortama kayma algoritmasının ardışık iki çerçeve için soldan sağa yukarıdan aşağıya sırasıyla t anında seçilen hedef nesne, t+1 anındaki; 1. , 2. , 3. , 4. , 5. iterasyon sonucu .....	14
Şekil 2.4.	İterasyonlar boyunca elde edilen Bhattacharyya katsayısı değerleri .....	15
Şekil 3.1.	Kalman Filtresinin Özyinelemeli Çalışma Prensibi.....	19
Şekil 3.2.	Önerilen Yönteme Ait Akış Diyagramı .....	22
Şekil 4.1.	Panda sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 50, 110, 170, 220, 260, 310, 412 numaralı çerçevelere ait takip sonuçları .....	31
Şekil 4.2.	Panda sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri .....	32
Şekil 4.3.	Panda sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri .....	32
Şekil 4.4.	Person sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 200, 350, 450, 480, 650, 750, 947 numaralı çerçevelere ait takip sonuçları.....	33
Şekil 4.5.	Person sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri .....	34
Şekil 4.6.	Person sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri .....	34
Şekil 4.7.	Person partially occluded sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 50, 100, 140, 160, 190, 255, 305 numaralı çerçevelere ait takip sonuçları.....	35
Şekil 4.8.	Person partially occluded sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri .....	36
Şekil 4.9.	Person partially occluded sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri.....	36

Şekil 4.10.	Rubik's cube sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 101, 228, 300, 452, 500, 600, 716 numaralı çerçevelere ait takip sonuçları.....	37
Şekil 4.11.	Rubik's cube sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	38
Şekil 4.12.	Rubik's cube sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri .....	38
Şekil 4.13.	Cup sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 111, 166, 231, 300, 400, 504, 629 numaralı çerçevelere ait takip sonuçları.....	39
Şekil 4.14.	Cup sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	40
Şekil 4.15.	Cup sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri.....	40
Şekil 4.16.	Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları.....	41
Şekil 4.17.	Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	42
Şekil 4.18.	Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	42
Şekil 4.19.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları.....	43
Şekil 4.20.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri .....	44
Şekil 4.21.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri .....	44
Şekil 4.22.	Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapıldığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları.....	45
Şekil 4.23.	Ball sequence videosu için standart ortalama kayma algoritması	46

	kullanarak ve ölçeklemenin yapıldığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri .....	
Şekil 4.24.	Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri .....	46
Şekil 4.25.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları.....	47
Şekil 4.26.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	48
Şekil 4.27.	Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	48
Şekil 4.28.	Juice box sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 60, 120, 190, 246, 285, 314, 367 numaralı çerçevelere ait takip sonuçları.....	49
Şekil 4.29.	Juice box sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	50
Şekil 4.30.	Juice box sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	50
Şekil 4.31.	Juice box sequence videosu için önerilen yöntem kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 60, 120, 190, 246, 285, 314, 367 numaralı çerçevelere ait takip sonuçları.....	51
Şekil 4.32.	Juice box sequence videosu için önerilen yöntem kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	52
Şekil 4.33.	Juice box sequence videosu için önerilen yöntem kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	52
Şekil 4.34.	Person floor sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 84, 150, 210, 280, 295, 315, 388 numaralı çerçevelere ait takip sonuçları.....	53
Şekil 4.35.	Person floor sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	54

Şekil 4.36.	Person floor sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	54
Şekil 4.37.	Person floor sequence videosu için önerilen yöntem kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 84, 150, 210, 280, 295, 315, 388 numaralı çerçevelere ait takip sonuçları.....	55
Şekil 4.38.	Person floor sequence videosu için önerilen yöntemi kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	56
Şekil 4.39.	Person floor sequence videosu için önerilen yöntemin kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri...	56
Şekil 4.40.	Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 58, 95, 113, 130, 141, 149, 154, 160 numaralı çerçevelere ait takip sonuçları	57
Şekil 4.41.	Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	58
Şekil 4.42.	Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	58
Şekil 4.43.	Person fully occluded sequence videosu için önerilen yöntem kullanılarak soldan sağa ve yukarıdan aşağıya sırasıyla 58, 95, 113, 130, 141, 149, 154, 160 numaralı çerçevelere ait takip sonuçları.....	59
Şekil 4.44.	Person fully occluded sequence videosu için önerilen yöntem kullanılarak takip işlemi boyunca Bhattacharyya katsayısı değerleri.....	60
Şekil 4.45.	Person fully occluded sequence videosu için önerilen yöntem kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri.....	60

## TABLÖLAR DİZİNİ

Çizelge 4.1.	Kullanılan videolar ve özellikleri.....	24
--------------	---	----

## SİMGELER VE KISALTMALAR

$P_k$	k anındaki hata kovaryans matrisi
$X_k$	k anındaki durum vektörü
$\delta$	Kronocker delta fonksiyonu
$\rho$	Batakarya katsayısı
C	Hedef nesneye ait normalizasyon katsayısı
$C_h$	Aday nesneye ait normalizasyon katsayısı
h	Bant genişliği
H	Ölçüm matrisi
I	Birim matris
R	Ölçüm hata kovaryansı
K	Kalman kazancı
F	Durum geçiş matrisi
Q	Süreç gürültü kovaryans matrisi
q	Hedef nesneye ait olasılık yoğunluğu
p	Aday nesneye ait olasılık yoğunluğu
ALOV++	Amsterdam Library of Ordinary Videos (Amsterdam Doğal Videolar Kütüphanesi)
BoBoT	Bonn Benchmark on Tracking (Takip için Bonn Kıyaslama Kümesi)
FLIR	Forward Looking InfraRed ( İleri Bakış Kızılötesi)
fps	frame per second (Saniyedeki çerçeve sayısı)
HSV	Hue Saturation Value (Renk Doygunluk Değer)
OpenCV	Open Computer Vision (Açık Kaynak Bilgisayarlı Görü)
RGB	Red Green Blue (Kırmızı Yeşil Mavi)

## 1. GİRİŞ

Nesne takibi bilgisayarlı görü alanının önemli bir unsurudur. Yüksek kapasiteli bilgisayarların yaygınlaşması, yüksek kaliteli ve ucuz kameraların elde edilebilmesi ve artan otomatik video analizi ihtiyacı nesne takip algoritmalarına olan ilgiyi arttırmıştır [1]. Nesne takibi, ilgilenilen nesnenin görüntü dizileri boyunca yörüngesinin ve kapladığı alanın belirlenmesi işlemidir. Nesne takibi; algısal kullanıcı arabirimleri (insan bilgisayar etkileşimi) [2], video analizi [3], gözetleme [4], el hareketleri tanıma [5], sürücü yardımcısı [6], spor müsabakaları [7,8], akıllı odalar [9] ve askeri uygulamalar gibi birçok alanda kullanılmaktadır.

Nesne takibi; üç boyutlu uzaydan iki boyutlu uzaya dönüşüm, görüntülerdeki gürültüler, nesnenin görünümünün değişmesi, nesnelerin karmaşık hareketleri, katı olmayan veya parçalı nesne yapısı, nesnenin bir kısmının veya tamamının engele maruz kalması, sahnedeki ışık yoğunluğunun değişmesi, kamera hareketi ve gerçek zamanlı işlem ihtiyacından dolayı karmaşık bir yapıya sahiptir [1]. İnsan bilgisayar etkileşiminin bir parçasını oluşturan bilgisayarlı görü algoritmaları gerçek zamanlı çalışabilmesi için hızlı ve etkili olmalıdırlar. Bilgisayarlı görü algoritmaları takip işlemini, hesapsal kaynakların büyük çoğunluğunu tüketmeden gerçekleştirebilmelidir [2].

Bahsedilen problemlerin çözülmesi için birçok takip algoritması önerilmiştir. Takip algoritmaları genellikle belirli kısıtlamalar çerçevesinde takip problemini ele alıp çözüm üretirler. Şu sorulara yaklaşım yönüyle takip algoritmaları farklılık gösterirler: “Hangi nesne temsili takip için uygundur?”, “Hangi özellikler kullanılmalıdır?”, “Nesnenin hareketi, görünümü ve şekli nasıl modellenmelidir?”. Yukarıdaki soruların cevapları takip algoritmasının yürütüldüğü çevre koşullarıyla doğrudan bağlantılıdır [1].

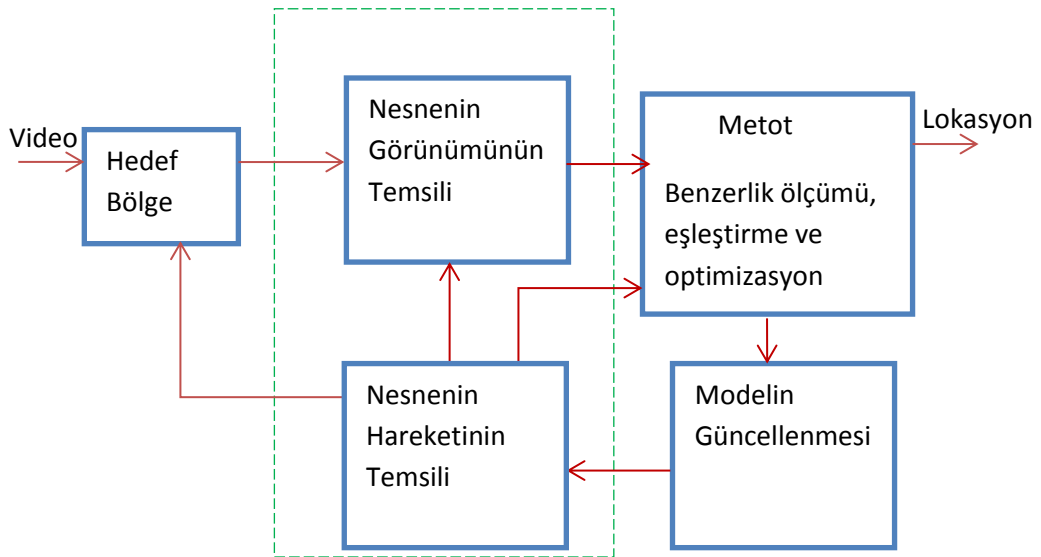
Nesne takip algoritmalarının önemli beş bileşeni: Hedef bölgenin temsili, nesne görünümünün temsili, nesne hareketinin temsili, metot (benzerlik ölçümü) ve nesne modelinin güncellenmesidir. Bu bileşenler arasındaki ilişki şekil 1.1.’de görülmektedir.

## 1.1. Hedef Nesnenin Kapladığı Bölgenin Temsili

Nesne takibinde hedef nesne takip edilmesi istenilen nesnedir. Hedef nesne uygulamaya göre değişebilir. Futbol müsabakası için hedef nesne bir futbolcu olabilirken, trafik güvenliği için hedef nesne bir yaya veya araç olabilir. Nesneler şekil ve görünümüyle temsil edilebilirler. Nesnenin bölgesel temsilde kullanılan yöntemler: Nokta tabanlı, temel geometrik şekil tabanlı, kontur/silüet tabanlı, parça tabanlı temsildir. Seçilen temsil nesnenin görüntü düzleminde yapabileceği hareketi de belirler.

Nokta tabanlı temsilde hedef nesne (şekil 1.2.a) bir nokta ile genellikle ağırlık merkezi olacak şekilde (şekil 1.2.b) veya birkaç nokta ile (şekil 1.2.c) temsil edilir. Nokta tabanlı temsil nesnenin görüntü düzleminde küçük alan kapladığı koşullar için uygundur [1].

Temel geometrik şekil ile nesne temsilde hedef nesne tek dikdörtgen (şekil 1.2.d), çoklu dikdörtgen (şekil 1.2.h) veya elips (şekil 1.2.e) gibi şekillerle temsil edilir. Basit katı nesnelerin temsili için temel geometrik şekiller uygundur [1]. Düşük model karmaşıklığından dolayı nesne takip algoritmalarında çokça tercih edilir.



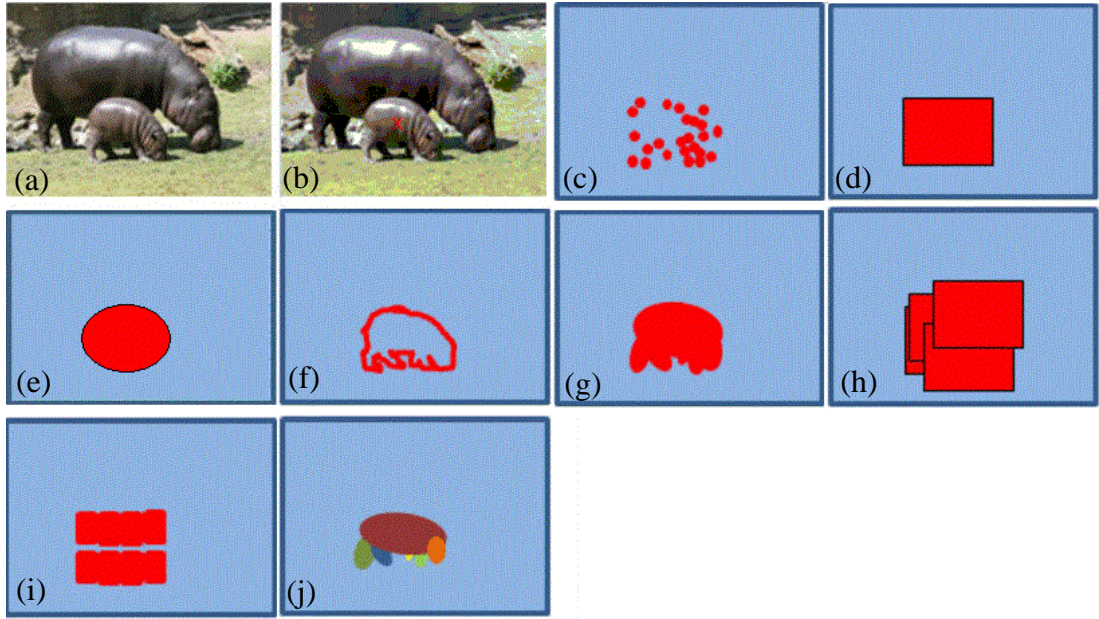
Şekil 1.1. Nesne takip algoritmalarının 5 önemli bileşeni [10]

Kontur nesnenin sınırlarını belirtirken (şekil 1.2.f) silüetse (şekil 1.2.g) kontur içerisinde kalan alanı belirtir. Katı olmayan, karmaşık şekle sahip nesneler için



kontur/silüet tabanlı nesne temsili uygundur [1]. Kontur/silüet tabanlı temsil, nesne görünümünün değişikliği için maksimum serbestlik sağlar. Yaya takibi ve biyomedikal uygulamalar için uygundur [10].

Parça tabanlı nesne temsili (şekil 1.2.i, şekil 1.2.j) nesne bağımsız parçalara ayrılır. İnsan takibi için bu temsil uygundur. İnsan vücudu gövde, kollar, bacaklar, baş ve ayaklardan oluşur. Bu parçalar eklemlerle birbirlerine bağlıdır ve belirli bir kinematik hareket modeline tabidirler.



Şekil 1.2. Nesne takibinde hedef bölgenin temsili [10].

## 1.2. Hedef Nesnenin Görünümünün Temsili

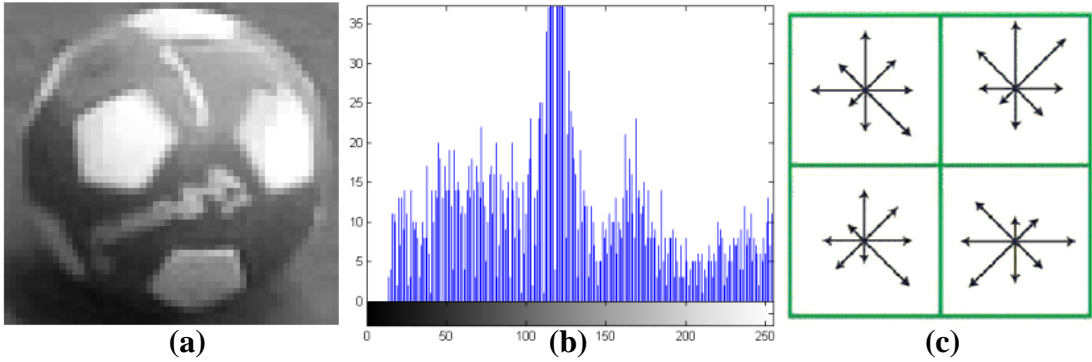
Nesnenin görünümü görsel özellikler tarafından temsil edilir. Seçilen özelliğin nesneyi bulunduğu sahneden ve sahnedeki diğer nesnelere ayırt ettirebilmesi beklenir. Nesnenin ayırt edici özelliklerle temsili takip işleminin başarısını olumlu yönde etkiler. Nesnenin görünümünün temsili için kullanılan başlıca yöntemler: Şablon, olasılık yoğunluğu ve özellik vektörü temsildir.

Şablon tabanlı temsilde (şekil 1.3.a) nesne genellikle 2 boyutlu parlaklık dizisi şeklinde temsil edilir. Bu temsilde şablon sadece nesnenin belirli bir yönden alınan

pozuna ait bilgi tutar. Nesnenin görünümünün bariz bir şekilde değişmesi veya ortamdaki ışık şiddetinin değişmesi nesne takibinin başarısını olumsuz yönde etkiler.

Olasılık yoğunluğu temsilinde (şekil 1.3.b) nesnenin seçilen özellik uzayına ait (renk, doku) olasılık dağılımı hesaplanır. Bu olasılık dağılımı hesabı Gauss gibi parametrik olan bir dağılımdan veya histogram gibi parametrik olmayan bir dağılımdan elde edilebilir.

Özellik vektörü temsilinde (şekil 1.3.c) nesne, belirlenen özellik çıkarıcı yöntemler sonucunda elde edilen bilgilerle temsil edilir.



Şekil 1.3. Nesne görünümüne ait özelliklerin temsili a) 2 boyutlu parlaklık dizisi, b) histogram c) özellik vektörü [10].

Takip problemini belirli koşullarda çözebilmek için birçok algoritma önerilmiştir. Bu algoritmaların kıyaslanabilmesi önemlidir. Smeulders vd. 315 video dizisinden oluşan ALOV++ veri kümesini 19 farklı nesne takip algoritması için çalıştırmıştır. Değerlendirilen nesne takip algoritmalarının yarısı 1999-2006 yıllarına aitken diğer yarısı son zamanlardaki büyük konferanslarda yayınlanan çalışmalardan oluşur [10]. Wu vd. ise 50 video dizisini 29 farklı takip algoritması için çalıştırmışlardır [11].

Bölüm 2’de, bu tezde takip algoritması olarak seçilen ortalama kayma algoritması tanıtılacak, bölüm 3’te Kalman filtresi ve önerilen yöntem anlatılacak, bölüm 4’te uygulama, bölüm 5’te ise sonuçlar sunulacaktır.

## 2. ORTALAMA KAYMA ALGORİTMASI

Bu bölümde ilk olarak ortalama kayma algoritması tanıtılacak sonra bu algoritmanın nesne takibinde kullanımı sunulacaktır.

### 2.1. Ortalama Kayma Algoritması

Sürekli bir noktadaki olasılık fonksiyonunun değeri, o nokta civarında küçük bir bölgedeki örnek gözlemleri kullanarak kestirilebilir. Parametrik olmayan olasılık yoğunluk fonksiyonu kestirimi bu kavrama dayanır [12].

Çekirdek yoğunluk kestirici de, parametrik olmayan bir yöntemdir ve bu yöntemin amacı sonlu sayıda verilen d-boyutlu verileri kullanarak verilerin üretildiği olasılık dağılım fonksiyonunu hesaplamaktır.  $\{x_i\}_{i=1\dots n}$  d-boyutlu uzayda veriler olsun.  $K(\cdot)$  çekirdek fonksiyonu ve  $h$  pencere genişliği kullanılarak,  $x$  noktası için hesaplanan çok değişkenli çekirdek yoğunluk kestirimi denklem 2.1.'deki gibidir.

$$\hat{f}_{K,h}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2.1)$$

$h$ , bant genişliği veya düzleştirme parametresi olarak da adlandırılır [13].  $h$  parametresi çekirdek yoğunluk kestiricinin sonucunu önemli ölçüde etkiler.  $h$  parametresinin çok büyük olması olasılık kestiriminin fazlaca düzleştirilmesine sebep olurken küçük olması ise olasılık kestiriminin keskin olmasına yol açar. Çekirdek yoğunluk kestiriminde yaygın olarak Epanechnikov ve Gauss çekirdek fonksiyonları kullanılır. Bu fonksiyonlar sırasıyla denklem 2.2. ve denklem 2.3.'te görülmektedir. Denklem 2.2.'deki  $c_d$ , d boyutlu birim kürenin hacmidir ve olasılık dağılımının 1 olmasını sağlar.

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - \|x\|^2), & \|x\| < 1 \\ 0, & \text{diğer durumda} \end{cases} \quad (2.2)$$

$$K_N(x) = 2\pi^{-d/2} \exp\left(-\frac{1}{2}\|x\|^2\right) \quad (2.3)$$

Denklem 2.4.'teki gibi  $k(\cdot)$  fonksiyonu tanımlanıp, denklem 2.1.'de yerine bırakılırsa denklem 2.5. elde edilir.

$$K(x) = k(\|x\|^2) \quad (2.4)$$

$$\hat{f}_{K,h}(x) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (2.5)$$

Denklem 2.5.'in gradyanı alınırsa denklem 2.6. elde edilir.

$$\nabla \hat{f}_{K,h}(x) = \frac{2}{nh^{d+2}} \sum_{i=1}^n (x - x_i) k'\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (2.6)$$

$$g(x) = -k'(x) \quad (2.7)$$

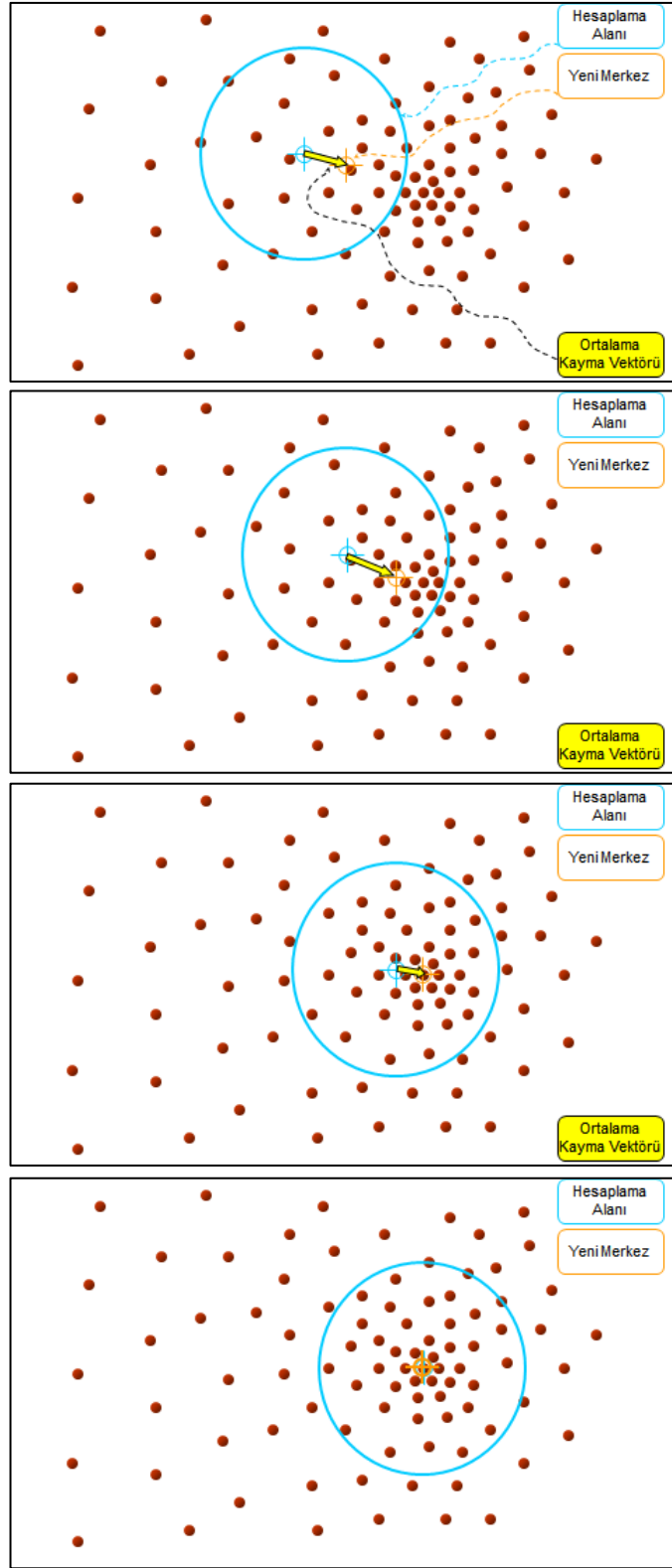
$g(\cdot)$  fonksiyonu denklem 2.7.'deki gibi tanımlanırsa ve denklem 2.6.'da yerine bırakılırsa denklem 2.8. elde edilir.

$$\nabla \hat{f}_{K,h}(x) = \frac{2}{nh^{d+2}} \sum_{i=1}^n (x_i - x) g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (2.8)$$

Denklem 2.8.'de gerekli düzenlemeler yapılrsa, denklem 2.9 elde edilir.

$$\nabla \hat{f}_{K,h}(x) = \frac{2}{nh^{d+2}} \left[ \sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \right] \left[ \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right] \quad (2.9)$$

Denklem 2.9.'daki son köşeli parantezdeki işlemler ortalama kayma vektörünü temsil eder. Sürekli bir  $x$  noktasında hesaplanan parametrik olmayan çekirdek yoğunluk kestiriminin gradyanı, o noktada hesaplanan ortalama kayma vektörüyle oransaldır. Ortalama kayma vektörü, yoğunluğun maksimum artış gösterdiği noktaya yönlendirir. Ortalama kayma algoritması olasılık dağılımının yerelde tepe noktasını (mode) bulmaya yarayan parametrik olmayan bir yöntemdir ve ilk olarak Fukunaga tarafından önerilmiştir [12]. Çalışma prensibi aşağıdaki gibidir:



Şekil 2.1. Ortama kayma algoritmasının yinelemeli çalışıp yoğun bölgenin bulunması [20].

- x noktası merkez olacak şekilde çekirdeği yerleştir ve bu noktada ortalama kayma vektörünü hesapla.

- Çekirdeği ortalama kayma vektörünün hesapladığı noktaya kaydır.

Bu işlemler  $\nabla \hat{f}_{K,h}(x) = 0$  bulunana kadar yinelemeli bir şekilde devam ettirilir. Ortalama kayma algoritmasının örnek veriler için çalışması şekil 2.1.'de gözlenmektedir. Hesaplama alanındaki veriler işleme tabi tutularak her bir iterasyonda ortalama kayma vektörü hesaplanmıştır. 3 iterasyon sonunda en yoğun bölge bulunmuştur. Ortalama kayma algoritması işlem kolaylığından ötürü birçok görüntü işleme ve bilgisayarlı görü alanında kullanılmıştır. Kümeleme [14], görüntü bölütleme [15-17] ve nesne takibi [2, 18, 19] ortalama kayma algoritmasının kullanıldığı alanlara örnek verilebilir.

## 2.2. Ortalama Kayma Algoritmasının Nesne Takibinde Kullanılması

Nesneler şekilleriyle, görünümüleriyle temsil edilebilirler. Nesne takibinde nesnelere temsil etmek için seçilen özelliklerin ayırt edici özellikler olması beklenir. Doğru özelliği seçme nesne takibinin başarısında kritik rol oynar. Yapılan nesne takip çalışmalarında renk özelliği kullanılan en temel özelliktir. Ortalama kayma algoritmasında, nesne elips bir şekilde temsil edilir. Nesnenin merkezine oturtulan çekirdek fonksiyonlarıyla nesnenin seçilen özelliğine ait histogramı hesaplanır. Çekirdek fonksiyonları merkezden uzaklaşan piksellerin olasılık yoğunluğuna etkilerini azaltır. Bu işlem etkilidir çünkü nesnelerin merkezden uzak olan pikselleri genellikle başka nesnelere tarafından engellenirler [19].

### 2.2.1. Hedef Nesnenin Temsili

Takip edilmesi istenen nesnenin modeline hedef model denir. Hedef nesneyi karakterize edebilmek için ilk olarak bir özellik uzayı seçilir. Hedef model, seçilen özellik uzayına göre olasılık yoğunluk fonksiyonu şeklinde temsil edilir [19]. Takip işleminde gerçek zamanlı çalışma ihtiyacından dolayı hedef model,  $\hat{q} = \{\hat{q}_u\}_{u=1\dots m}$  m bin histogram olacak şekilde denklem 2.10 yardımıyla hesaplanır.

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (2.10)$$

Bu denklemde  $\{x_i^*\}_{i=1\dots n}$ , hedef modelin merkez noktasına göre normalizasyon uygulanmış piksel koordinatlarını temsil eder.  $k(\cdot)$ , çekirdek fonksiyonudur ve seçilen hedef nesnenin merkezini (0,0) noktası kabul edecek şekilde merkezden uzaklaşan piksellerin olasılık yoğunluğuna daha az katkı yapmasını sağlar. Bu işlem etkilidir çünkü genellikle merkezden uzak olan pikseller engelle maruz kaldıklarından daha az güvenilirlerdir [19].  $b(\cdot)$  fonksiyonu kendisine parametre olarak gelen piksel koordinatındaki seçilen özellik uzayına ait (renk, doku) değeri verir.  $C$ , hedef modele ait normalizasyon katsayısını ifade eder.  $C$ , değeri denklem 2.11 kullanılarak hesaplanır ve denklem 2.12 şartını sağlaması için vardır. Denklem 2.12'deki  $m$  değeri histogramdaki bin sayısını belirtir.

$$C = \frac{1}{\sum_{i=1}^n k(\|x_i^*\|^2)} \quad (2.11)$$

$$\sum_{u=1}^m \hat{q}_u = 1 \quad (2.12)$$

Denklem 2.10'daki Kronocker delta fonksiyonu denklem 2.13'teki gibi sonuç üretir.

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (2.13)$$

### 2.2.2. Aday Nesnenin Temsili

Aday model, hedef nesne olma olasılığı olan modeldir.  $y$  merkezli aday modele ait olasılık yoğunluğu  $\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m}$   $m$  bin histogram olacak şekilde denklem 2.14 yardımıyla hesaplanır.

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (2.14)$$

$\{x_i\}_{i=1\dots n_h}$ ,  $y$  noktası merkez olacak şekilde normalizasyon uygulanmış piksel koordinatlarını temsil eder.  $h$  ise işleme alınacak piksel sayısını belirler yani bant genişliğidir.  $C_h$ , aday modele ait normalizasyon katsayısını ifade eder denklem

2.15'deki gibidir ve denklem 2.16 şartını sağlaması için vardır. Denklem 2.16'daki m değeri histogramdaki bin sayısını belirtir.

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \quad (2.15)$$

$$\sum_{u=1}^m \hat{p}_u = 1 \quad (2.16)$$

### 2.2.3. Takip İşlemi

Hedef modele ve aday modele ait olasılık yoğunlukları hesaplandıktan sonra bu olasılık yoğunlukları arasındaki benzerliğin bulunması gerekir. Bhattacharyya katsayısı iki olasılık yoğunluğunun benzerliğinin hesaplanmasında kullanılabilir. Hedef model ve y merkezinde hesaplanan aday model arasındaki benzerliği temsil eden Bhattacharyya katsayısı denklem 2.17'deki gibi hesaplanır.

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (2.17)$$

Bhattacharyya katsayısı maksimum 1, minimum 0 olabilir. 1 olması iki olasılık yoğunluğunun tamamen benzeştiğini belirtir.

Nesne takibinin etkili bir şekilde gerçekleşmesi için Bhattacharyya katsayısının maksimize edilmesi gerekir. Bunun için Bhattacharyya katsayısını maksimize edecek nokta aranır.  $y_0$ , seçilen hedef nesnenin ilk merkezi olmak üzere 2.17 numaralı denkleme  $\hat{p}_u(y_0)$  civarında Taylor açılımı uygulanırsa denklem 2.18 elde edilir.

$$\rho[\hat{p}(y), \hat{q}] \approx \rho[\hat{p}(y_0), \hat{q}] + \sum_{u=1}^m (\hat{p}_u(y) - \hat{p}_u(y_0)) \rho'(\hat{p}_u(y_0), \hat{q}_u) \quad (2.18)$$

Denklem 2.18'de gerekli düzenlemeler yapılırsa denklem 2.19 elde edilir. Denklem 2.19 aşağıda verildiği gibi düzenlenirse denklem 2.23 elde edilir.

$$\rho[\hat{p}(y), \hat{q}] \approx \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m (\hat{p}_u(y) - \hat{p}_u(y_0)) \frac{\hat{q}_u}{\sqrt{\hat{p}_u(y_0) \hat{q}_u}} \quad (2.19)$$



$$\approx \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m (\hat{p}_u(y) - \hat{p}_u(y_0)) \frac{\sqrt{\hat{q}_u}}{\sqrt{\hat{p}_u(y_0)}} \quad (2.20)$$

$$\approx \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \frac{\sqrt{\hat{q}_u}}{\sqrt{\hat{p}_u(y_0)}} - \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} \quad (2.21)$$

$$\approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \frac{\sqrt{\hat{q}_u}}{\sqrt{\hat{p}_u(y_0)}} \quad (2.22)$$

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \frac{\sqrt{\hat{q}_u}}{\sqrt{\hat{p}_u(y_0)}} \quad (2.23)$$

Denklem 2.14, denklem 2.23'de kullanılırsa denklem 2.24 elde edilir.

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{c_h}{2} \sum_{i=1}^{n_h} w_i k \left( \left\| \frac{y-x_i}{h} \right\|^2 \right) \quad (2.24)$$

Denklem 2.24'deki  $w_i$ ,  $u$  özelliğine sahip piksel ağırlığını belirtir ve denklem 2.25'deki gibidir.

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \quad (2.25)$$

Bhattacharyya katsayısını maksimize etmek için denklem 2.24'deki 2. terimin maksimize edilmesi gerekir. Bu terim,  $k(\cdot)$  çekirdeği kullanılarak ve  $w_i$  katsayısı ile ağırlıklandırılmış  $y$  merkezli çekirdek yoğunluk kestiricidir ve ortalama kayma vektörü Bhattacharyya katsayısını yerelde maksimize eden koordinata götürür. Bhattacharyya katsayısını yerelde maksimize eden koordinat denklem 2.26 yardımıyla bulunur.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g \left( \left\| \frac{\hat{y}_0 - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left( \left\| \frac{\hat{y}_0 - x_i}{h} \right\|^2 \right)} \quad (2.26)$$

Hedef nesneye ve aday nesneye ait histogram hesaplanırken  $k(.)$  çekirdek fonksiyonu olarak Epanechnikov kullanılırsa  $g(.)$  fonksiyonu sabit değer üretecektir ve denklem 2.26 denklem 2.27 halini alacaktır.

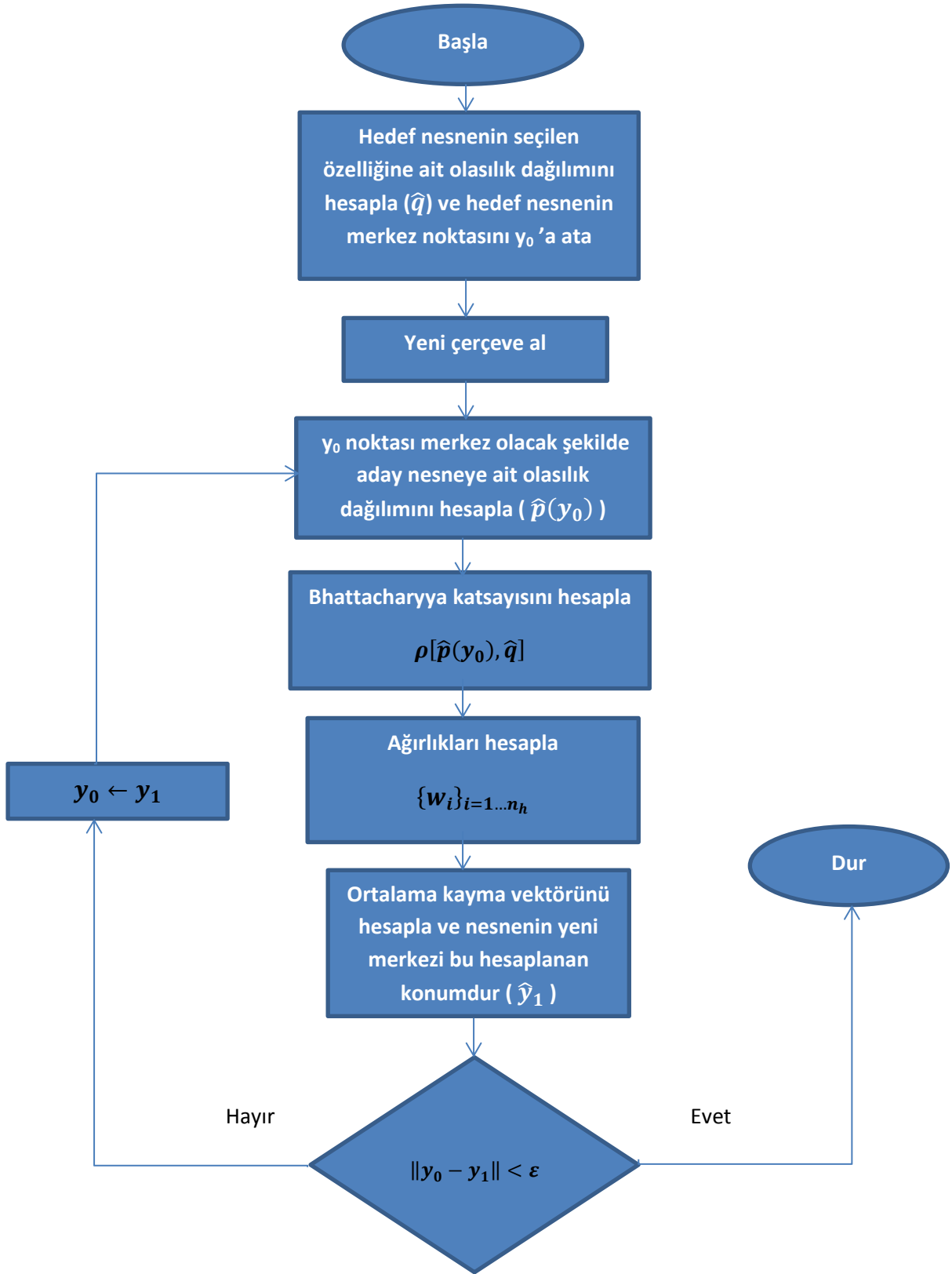
$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (2.27)$$

Ortalama kayma algoritması yinelemeli olarak çalışan bir yöntemdir. Yer değiştirme  $\|\hat{y}_1 - \hat{y}_0\|$  büyüklüğü belirlenen bir epsilon değerinden küçük olana kadar işlemler yinelenir. Hedef nesneyi temsil eden elipsin merkezi  $\hat{y}_0$  noktasından  $\hat{y}_1$  noktasına kaydırılır ve gerekli işlemler tekrardan yapılır. Ortalama kayma algoritmasının çalışma prensibi şekil 2.2.'de sunulmuştur.

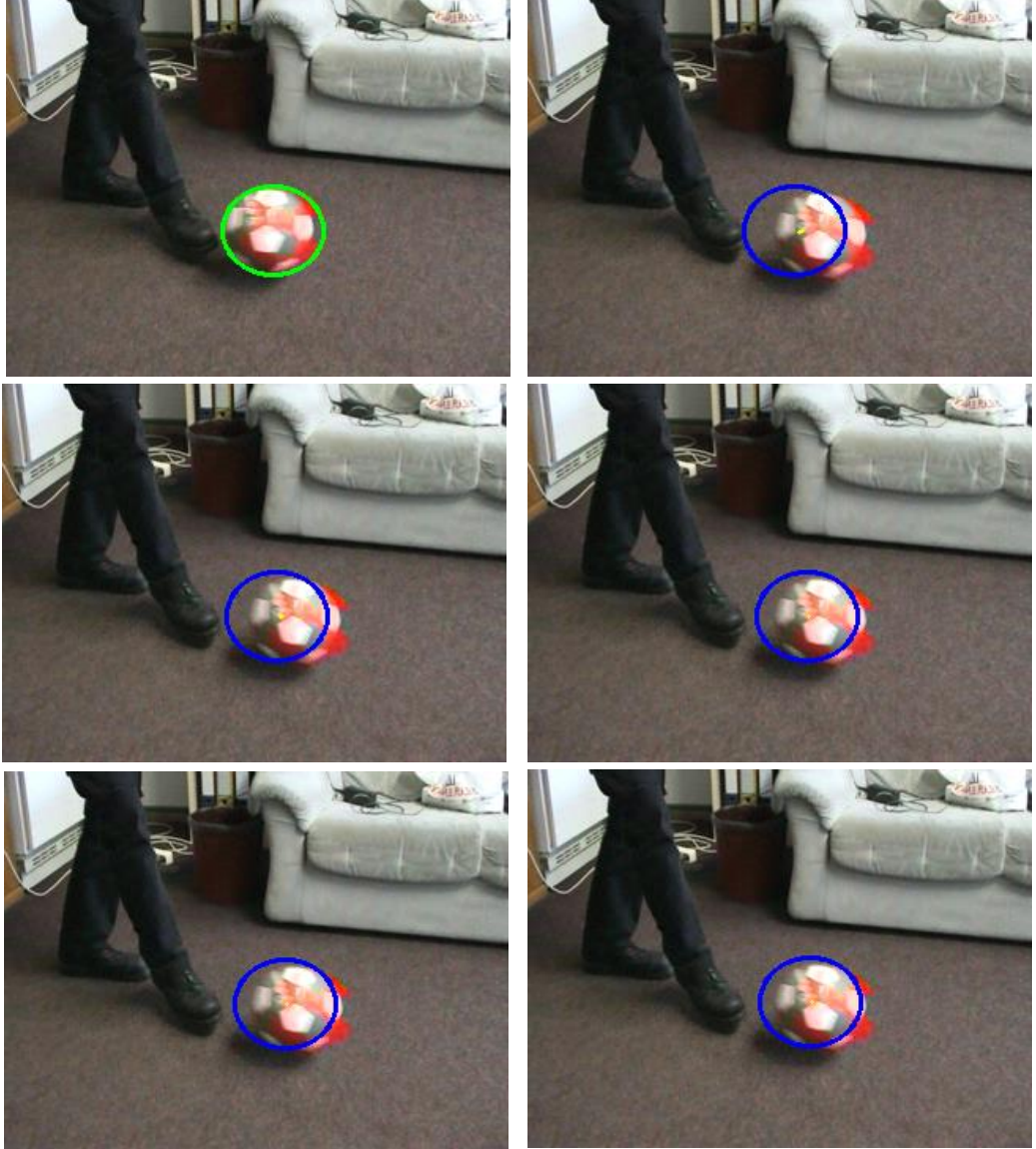
Yine ortalama kayma algoritmasının örnek ardışık iki çerçeve için ürettiği takip sonuçları şekil 2.3.'de ve bu örneğe ait yinelemeler boyunca elde edilen Bhattacharyya değerleri şekil 2.4.'de sunulmuştur. t anında elips şeklinde seçilen hedef nesne olan futbol topu t+1 anında seçildiği koordinat merkez olacak şekilde aranmaya başlar. Ortalama kayma algoritması hedef nesneyi 5 iterasyon sonucunda bulmuştur. Şablon eşleştirme tabanlı algoritmalarla kıyaslandığında ortalama kayma algoritması takip problemini işlem yükü bakımından daha etkili çözer.

Hedef model ve aday model arasındaki benzerliği belirten Bhattacharyya katsayısı değerlerinin her ortalama kayma iterasyonları sonucunda arttığı şekil 2.4.'de gözlenmektedir.

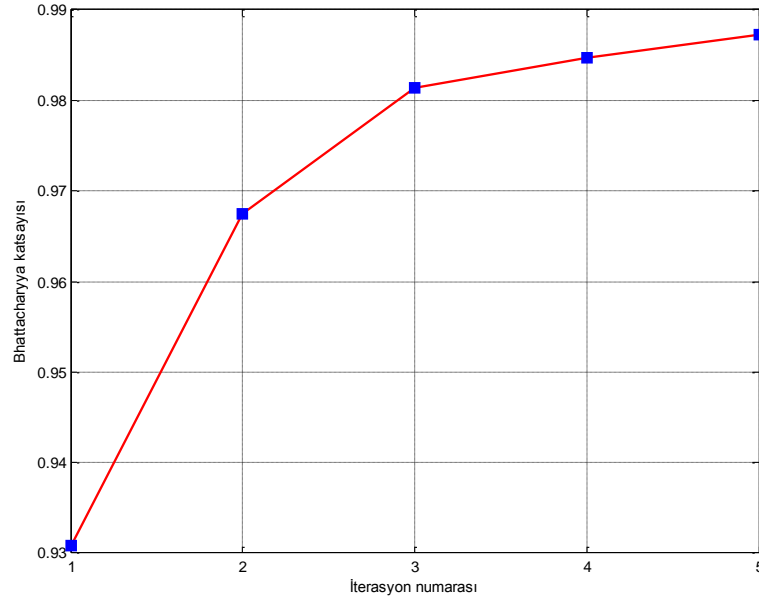
Etkili bir yöntem olan ortalama kayma algoritması nesne takibinde birçok çalışmada kullanılmış ve gerekli iyileştirmeler yapılmıştır. Yılmaz vd., FLIR görüntülerinde nesne takibi için ortalama kayma algoritmasını kullanmışlardır. Özellik uzayı olarak parlaklık ve Gabor filtre cevabı kullanılmıştır. Önerdikleri takip işlemi iki modülden oluşmaktadır: İlk modül, ortalama kayma iterasyonlarıyla hedef nesne ve aday nesne arasındaki benzerliğin maksimize edilmesini gerçekleştirir. İkinci modül ise ilk modülün, hareketli platformun öz hareketinden (ego motion) ötürü hedef nesneyi kaybetmesiyle devreye girer ve global hareketi dengeler [21].



Şekil 2.2. Ardışık iki çerçeve için ortalama kayma algoritmasının çalışma prensibini gösteren akış diyagramı



Şekil 2.3. Ortama kayma algoritmasının ardışık iki çerçeve için soldan sağa yukarıdan aşağıya sırasıyla  $t$  anında seçilen hedef nesne,  $t+1$  anındaki; 1. , 2. , 3. , 4. , 5. iterasyon sonucu



Şekil 2.4. İterasyonlar boyunca elde edilen Bhattacharyya katsayısı değerleri

Jeyakar vd., hedef nesneyi parçalı yapıda olacak şekilde çoklu elipslerle temsil etmişlerdir. Buradaki amaç nesnenin kısmi engele maruz kalması koşullarında da nesneyi doğru bir şekilde takip edebilmektir. Hedef nesneye ait histogram hesaplanırken de nesnenin yakın çevresi göz önüne alınarak hesaplama işlemi gerçekleştirilmiştir. Hedef nesnede bulunan belirli bir özellik (renk, parlaklık, doku) hedef nesnenin çevresinde de bulunuyorsa histogram hesaplanırken o özelliğe ait olasılık dağılımının etkisi azaltılmıştır. İstenen, hedef nesneyi çeldiricilerden (clutter) uzak tutarak takip işleminin doğruluğunu arttırmaktır [22].

Yang vd., hedef nesne ve aday nesne arasındaki benzerliği belirten Bhattacharyya katsayısına alternatif mekansal-özellik uzayında yeni bir benzerlik ölçüsü önermişlerdir [23].

Collins, ortalama kayma algoritmasına ölçek uzayını da dahil ederek hedef nesnenin görüntü düzleminde hem mekanın hem de boyutunun belirlenmesi işlemini gerçekleştirmişlerdir [24].

Leichter vd., hedef nesnenin görünümünün bariz değiştiği koşullarda da takip işlemini gerçekleştirebilmek için nesnenin farklı görünümüne ait çoklu referans histogramlarını kullanmışlardır [25].

Bir sonraki bölüm olan bölüm 3'te, Kalman filtresine ve önerilen yönteme değinilmiştir.

### 3. KALMAN FİLTRESİ VE ÖNERİLEN YÖNTEM

Bu bölümde ilk olarak Kalman filtresi sunulacak sonra önerilen yöntem olan, uyarlamalı Kalman filtresinin ortalama kayma algoritmasıyla bütünleştirildiği nesne takip algoritması anlatılacaktır.

#### 3.1. Kalman Filtresi

Kalman filtresi sistem modelinin lineer, süreç ve ölçüm gürültülerinin Gauss dağılım özelliğine sahip olduğu durumda özyinelemeli (recursive) olarak çalışıp sistem durum vektörünü kestiren etkili bir filtredir.

Kalman filtresi, k anındaki sistem durumunun k-1 anındaki sistem durumundan geliştiğini varsayar [26].

$$X_k = \Phi X_{k-1} + w_{k-1} \quad (3.1)$$

Denklem 3.1'de  $X_k$ , durum vektörüdür ve sistemin ilgilenilen parametrelerini (pozisyon, hız) bulundurur.  $\Phi$ , durum geçiş matrisidir ve k-1 anındaki sistemin durumunu k anındaki sistem durumuna yansıtır.  $w_{k-1}$ , süreç gürültüsüdür.

$$Z_k = HX_k + v_k \quad (3.2)$$

Denklem 3.2'de  $Z_k$ , ölçüm vektörüdür.  $H$ , dönüşüm matrisi  $v_k$ , ölçüm gürültüsüdür. Rastsal değişkenler olan süreç gürültüsü ve ölçüm gürültüsü birbirlerinden bağımsızdır, beyaz gürültüdürler ve normal olasılık dağılımına sahiptirler [27].

$$p(w) \sim N(0, Q) \quad (3.3)$$

$$p(v) \sim N(0, R) \quad (3.4)$$

Sistemin gerçek durum vektörü  $X_k$  doğrudan ölçülemediği için Kalman filtresi gürültülü ölçüm verisi ve süreç modelini göz önüne alarak sistemin durumunu

kestirimde bulunur.  $\hat{X}_k$ , k anındaki kestirilen sistem durumunu temsil eder. Kalman filtresinin çalışması iki aşamadan oluşur: Tahmin etme ve düzeltme [27].

Kalman filtresinin tahmin aşamasına ait denklemleri aşağıdaki gibidir.

$$\hat{X}_k^- = \Phi \hat{X}_{k-1} \quad (3.5)$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q \quad (3.6)$$

Denklem 3.5'te  $\hat{X}_k^-$ , k anındaki önsel durum kestirimidir.  $\hat{X}_{k-1}$  ise k-1 anındaki sonsal durum kestirimidir. Denklem 3.6'da  $P_k^-$ , önsel hata kovaryansıdır.  $Q$  ise süreç gürültü kovaryansıdır.

Kalman filtresinin düzeltme aşamasına ait denklemleri ise aşağıda verilmiştir.

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1} \quad (3.7)$$

$$\hat{X}_k = \hat{X}_k^- + K_k (Z_k - H \hat{X}_k^-) \quad (3.8)$$

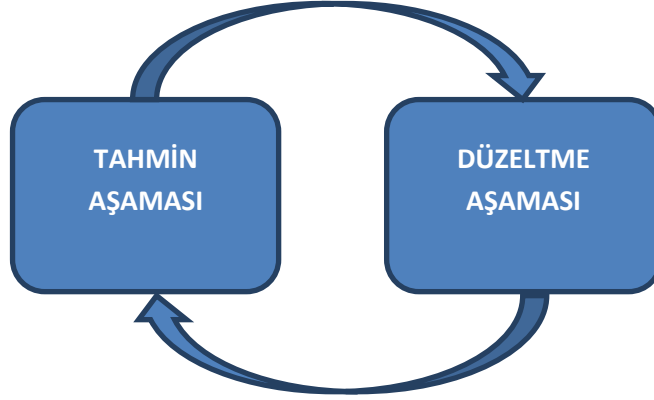
$$P_k = (I - K_k H) P_k^- \quad (3.9)$$

Denklem 3.7'de  $K_k$ , k anındaki Kalman kazancıdır ve denklem 3.8.'deki sonsal hata kovaryansı olan  $P_k$ 'yi minimize eder. Denklem 3.9'da  $R_k$  ölçüm hata kovaryansıdır.  $R_k$  sifıra yakınsadığı zaman ölçüm bilgisi olan  $Z_k$ 'ya daha fazla güvenilip, tahmin edilen  $H \hat{X}_k^-$  bilgisine daha az güvenilecektir. Buna karşılık önsel hata kovaryansı olan  $P_k^-$  sifıra yakınsadığı zaman  $Z_k$ 'ya daha az güvenilip, tahmin edilen  $H \hat{X}_k^-$  bilgisine daha fazla güvenilecektir [27].

Kalman filtresinin çalışma prensibi şekil 3.1.'de görülmektedir. Tahmin aşamasında lineer sistem modeli kullanılarak sistemin önsel kestirimi bulunur, düzeltme aşamasında ise ölçüm bilgisi ve önsel kestirim bilgisi kullanılarak sistemin gerçek durumu kestirilir. Bu işlemler özyinelemeli bir şekilde devam ettirilir. Yani k anındaki sistem durum kestiriminin bulunabilmesi için k-1 anındaki sonsal durum kestiriminin bulunması yeterlidir. Bu da Kalman filtresinin problem çözerken az sayıda parametreye ihtiyaç duyduğunu gösterir. Yine Kalman filtresi, sisteme ait ölçüm alınmadığı durumlarda bile kullanılan modeli göz önüne alarak sistem durumunu kestirebilir. Bu, Kalman filtresinin önemli bir özelliğidir.



İyi bilinmektedir ki Kalman filtresinin performansı iki kısma bağlıdır: Belirlenen uygun bir lineer model ve istatistiksel gürültü özellikleri [28].



Şekil 3.1. Kalman Filtresinin Özyinelemeli Çalışma Prensipleri

### 3.2. Önerilen Yöntem

Kalman filtresi düşük karmaşıklığından ötürü nesne takip problemlerinin çözümünde yaygın olarak kullanılır. Genellikle, ilgilenilen nesnenin engele maruz kaldığı koşullarda takip işlemini gerçekleştirebilmek için kullanılır [29-31]. [32] numaralı çalışmada Kalman filtresinin başarısını etkileyen ölçüm hata kovaryansı ve süreç hata kovaryansı çalışmada belirtilen engellenme oranına göre anlık güncellenmektedir. [33] numaralı çalışmada ise Kalman filtresinin durum geçiş matrisi belirlenen bir ölçüm nitelik fonksiyonuna göre anlık güncellenmektedir.

Hedef nesnenin görünümüne ait modelin uygun bir şekilde güncellenmesi nesne takip işleminin başarısını artırır. Model güncellemesi, ortamdaki ışık şiddetinin değiştiği koşullarda da takip işlemini başarabilmek için önemlidir. Yapılan çalışmalarda Kalman filtresi, hedef nesnenin görünümünün güncellenmesinde de kullanılmıştır. [34] numaralı çalışmada şablon eşleştirme yöntemi kullanılarak nesne takibi gerçekleştirilmiştir. Bulunan hedef nesneye ait şablon, Kalman filtresiyle düzeltilerek bir sonraki çerçeve için güncellenmiştir. [28] numaralı çalışmada ise Kalman filtresi kullanılarak hedef nesneye ait histogram düzeltilmiştir. Takip işlemi ise ortalama kayma algoritmasıyla gerçekleştirilmiştir.

[35] numaralı çalışmada destek vektör makinesi yardımıyla bulunan göz bebekleri Kalman filtresiyle takip edilmiştir. Kalman filtresinin göz bebeklerini takip edemediği durumda ise ortalama kayma algoritması devreye alınmıştır. Ortalama kayma algoritmasının da takip işleminde başarısız olduğu koşullarda tekrardan destek vektör makinesi yardımıyla göz bebeklerinin konumu tespit edilmiştir.

[36] numaralı çalışmada ise Kalman filtresi ve ortalama kayma algoritması birlikte kullanılarak hedef nesnenin görünümünün çevresiyle benzer özelliklerde bulunması ve kısmi engellenmesi koşullarında takip işlemini gerçekleştirmişlerdir.

Bu çalışmada ise, ortalama kayma algoritması ve uyarlamalı Kalman filtresi birlikte kullanılarak hedef nesnenin tam engellenmesi ve hızlı hareket etmesi koşullarında da başarılı bir şekilde takip işlemini gerçekleştirebilen bir takip algoritması önerilmiştir. Standart ortalama kayma algoritması hedef nesneyi bir önceki bulunduğu konumda aramaya başlar ve ortalama kayma iterasyonlarıyla hedef nesne bulunmaya çalışılır. Hedef nesne ardışık iki çerçeve arasında hızlı hareket ederse veya engele maruz kalırsa standart ortalama kayma algoritması hedef nesneyi takip edememektedir.

Çalışmada hedef nesnenin belirlenen durum vektörü,  $X_k = [x_k \ y_k \ V_{xk} \ V_{yk}]^T$  gibidir. Durum vektöründeki  $(x_k, y_k)$ , hedef nesnenin k anındaki pozisyon bilgisiyken  $(V_{xk}, V_{yk})$ , k anındaki hız bilgisini temsil eder. Hareket modeli olarak sabit hızlı hareket kullanılmıştır. Kalman filtresinin belirlenen durum geçiş matrisi ve dönüşüm matrisi aşağıdaki gibidir.

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.11)$$

Önerilen yöntemde hedef nesne herhangi bir çerçevede seçildikten sonra yeni bir çerçeve alınır ve takip sisteminin çalışması Kalman filtresinin belirlenen sabit hızlı hareket modeline göre nesnenin yeni pozisyonunu tahmin etmesiyle başlar. Ortalama kayma algoritması Kalman filtresinin tahmin ettiği pozisyondan başlayarak

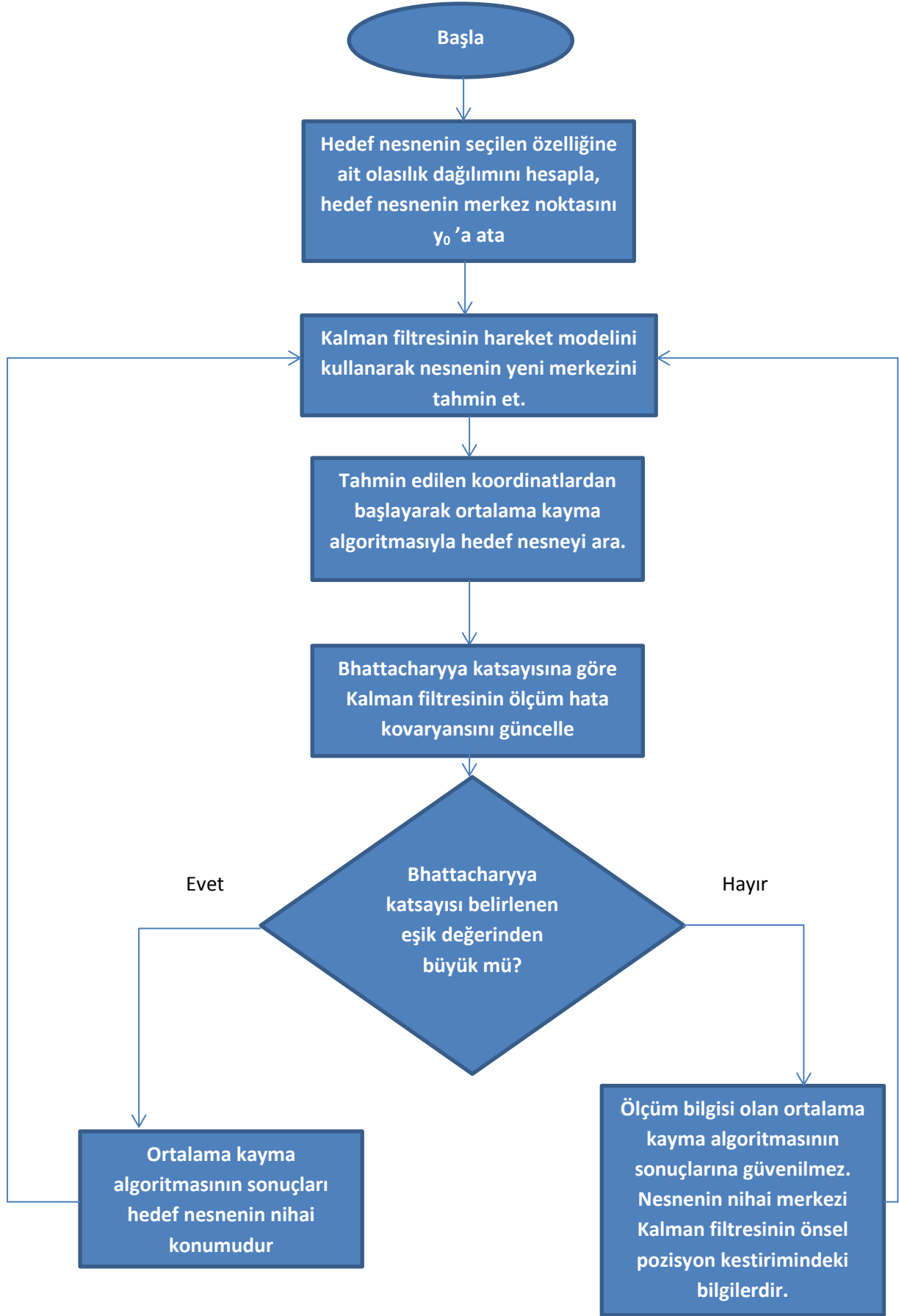
hedef nesneyi arar. Bulunan aday nesne ile hedef nesne arasındaki benzerlik Bhattacharyya katsayısıyla belirlenir. Ortalama kayma algoritması sonucunda ölçülen Bhattacharyya katsayısına bağlı olarak Kalman filtresinin ölçüm gürültü kovaryansı denklem 3.12'deki gibi hesaplanır.

$$R_k = \begin{cases} I - \rho[\hat{p}(y), \hat{q}]I, & \rho[\hat{p}(y), \hat{q}] > \text{eşik değeri} \\ \infty I, & \text{diğer durumda} \end{cases} \quad (3.12)$$

Bhattacharyya katsayısı belirlenen eşik değerinden büyükse ortalama kayma algoritmasının sonucu nesnenin nihai koordinatları olarak kabul edilir ve bu koordinatlar Kalman filtresinin de ölçüm değerleri olur.

Bhattacharyya katsayısı belirlenen eşik değerinden küçükse hedef nesnenin engele maruz kaldığı kabul edilir ve ölçüm bilgisi olan ortalama kayma algoritmasının sonuçlarına güvenilmez. Bu durumda Kalman filtresinin önsel durum kestirimi hedef nesnenin nihai koordinatları olarak kabul edilir.

Önerilen yöntemin ardışık iki çerçeve arasında çalışma prensibini gösteren akış diyagramı şekil 3.2.'de sunulmuştur.



Şekil 3.2. Önerilen Yönteme Ait Akış Diyagramı

#### 4. UYGULAMA

Bu tez çalışmasında, standart ortalama kayma algoritmasının ve önerilen yöntemin kıyaslanabilmesi için BoBoT video dizileri kullanılmıştır [37]. Bu video dizilerinin kullanılmasının amacı, her bir çerçeve için hedef nesneye ait pozisyon bilgisi ve ölçek bilgisinin bulunmasıdır. Takip edilen nesnelere ait bu bilgilerin bulunması kıyaslama işlemini kolaylaştırmıştır. Bu video dizileri farklı zorluklar içermektedir. Ortamdaki ışık şiddetinin değişmesi, hareketli kamera, nesnenin; ani hareket etmesi, eklemlili yapıda olması, kameraya yaklaşp uzaklaşması, arka planla benzer görünüm özelliklerine sahip olması, kısmi veya tam engellenmesi ve görünümünün değişmesi karşılaşılan zorluklardır. Kullanılan video dizileri ve bunlara ait özellikler çizelge 4.1.'de gösterilmektedir.

Uygulama kodları C++ programlama dili ve OpenCV kütüphaneleri kullanılarak yazılmıştır. Intel i7 2.4GHz işlemciye sahip bir bilgisayar üzerinde algoritmalar yürütülmüştür. Bütün video dizilerinde RGB renk uzayı özellik olarak kullanılmıştır. Her bir renk kanalı 32 bine bölünmüştür (32x32x32). Bu ön işlemin amacı işlem maliyetini düşürerek gerçek zamanlı uygulama için performansın artırılmasını sağlamaktır.

Takip işleminde ölçeklendirmenin de gerçekleştirildiği video dizilerinde ortalama kayma algoritması 3 kez çalıştırılmıştır. İlk çalıştırmada hedef nesne bir önceki çerçevedeki boyut bilgisi baz alınarak aranır, ikinci çalıştırmada bu boyut bilgisi %10 arttırılıp, üçüncü çalıştırmada ise %10 azaltılarak aranır. Bu işlemler sonunda 3 farklı Bhattacharyya katsayısı elde edilir. Bhattacharyya katsayısının maksimum olduğu boyut bilgisi hedef nesnenin o anki çerçevedeki boyut bilgisi olarak kabul edilir.

Çizelge 4.1.'deki bazı video dizileri sadece ortalama kayma algoritmasıyla çalıştırılıp sonuçları incelenmiştir. Hedef nesnenin tam olarak engele maruz kaldığı veya ani hareket ettiği koşullardaki video dizileri için ise ortalama kayma algoritması ve önerilen yöntem ayrıca çalıştırılmıştır.

Hedef nesneye ait histogram Epanechnikov çekirdek fonksiyonu kullanılarak belirlenmiştir. Dolayısıyla ortalama kayma iterasyonları düzgün kesit fonksiyonu (uniform profile) ile hesaplanmıştır.

Çizelge 4.1. Kullanılan videolar ve özellikleri

Video Dizisi	Hedef Nesne	Zorluklar	Format	Çerçeve Sayısı
Panda Sequence	Oyuncak panda	Işık şiddetinin değişmesi	320 X 240 25fps	412
Person Sequence	İnsan	Kamera hareketi, nesne hareketi, tam katı olmayan nesne	320 X 240 25fps	947
Person Partially Occluded Sequence	İnsan	Kamera hareketi, kısmi engelleme	320 X 240 25fps	305
Cup Sequence	Kupa	Kamera hareketi, nesne hareketi, arka planın değişmesi, hedef nesnenin arka planla benzer olması, ölçek değişikliği	320 X 240 25fps	629
Rubik's Cube Sequence	Rubik küp	Kamera hareketi, hedef nesnenin kameraya bakan yüzünün değişmesi	320 X 240 25fps	716
Ball Sequence	Futbol topu	Kamera hareketi, nesne hareketi, hızlıca yön değiştirme	320 X 240 25fps	602
Juice Box Sequence	Meyve suyu kutusu	Kamera hareketi, hızlıca yön değiştirme, ölçek değişikliği	320 X 240 25fps	404
Person Floor Sequence	İnsan	Kamera hareketi, nesne hareketi, tam katı olmayan nesne, nesnenin tam engellenmesi	320 X 240 25fps	388
Person Fully Occluded Sequence	İnsan	Kamera hareketi, nesne hareketi, tam katı olmayan nesne, nesnenin tam engellenmesi	320 X 240 25fps	453

#### **4.1. Ortalama Kayma Algoritmasının Nesne Takibinde Kullanılması**

Bu bölümde ortalama kayma algoritması çizelge 4.1.'de özellikleri belirtilen panda sequence, person sequence, person partially occluded sequence, cup sequence, Rubik's cube sequence videoları için çalıştırılmıştır.

##### **4.1.1. Panda Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı**

Bu videoda hedef nesne bir oyuncak pandadır. Karşılaşılan zorluk ortamdaki ışık şiddetinin aniden değişmesidir. Bu videoya ait takip sonuçları şekil 4.1.'de gösterilmiştir. Şekil 4.2.'de ise takip işlemi boyunca elde edilen Bhattacharyya katsayısı gösterilmektedir. Bhattacharyya katsayısı ortamdaki ışık şiddetinin hızlı değiştiği çerçevelerde farklılık göstermektedir. Ortamdaki ışık şiddeti, 1-95 çerçeveleri arasında sabit kalırken, 96-214 çerçeveleri arasında azalmıştır, 215-298 çerçeveleri arasında ışık şiddeti fazlaca artarken, 299-412 çerçeveleri arasında ışık şiddeti tekrardan azalmıştır. Şekil 4.3.'te ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Ortalama kayma algoritması hedef nesne olan oyuncak pandayı başarılı bir şekilde takip edebilmiştir.

##### **4.1.2. Person Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne insandır. Karşılaşılan zorluklar: Kameranın hareket etmesi, nesnenin; eklemlili yapıda olması, kendi etrafında dönmesi ve ölçeğinin değişmesidir. Nesnenin renk özelliğine ait histogramı hesaplanırken, nesnenin eklemlili yapısından ötürü arka plan bilgisi de hesaplanan histogramda yer alır. Bu da takip işlemi olumsuz etkiler. Bu videoya ait takip sonuçları şekil 4.4.'te gösterilmiştir. 1 numaralı çerçevede seçilen nesne 1-440 çerçeveleri arasında kendi etrafında dönmektedir, dolayısıyla nesnenin görünümü değişmektedir. Şekil 4.6.'ya bakılırsa 475-500 numaralı çerçeveler arasında nesnenin gerçek y pozisyonu ve hesaplanan y pozisyonu arasında açık bir sapma vardır bunun nedeni, arka planda hedef nesne de var olan siyah renk tonunun bulunmasıyla ilgilidir. Şekil 4.5.'te takip işlemi boyunca elde edilen Bhattacharyya katsayısı, şekil 4.6.'da ise hedef

nesnenin x ve y koordinatlarına ait gerçek değer ve ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Ortalama kayma algoritması bu video dizisi için başarılı sonuçlar üretmiştir.

#### **4.1.3. Person Partially Occluded Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne insandır. Karşılaşılan zorluklar: Kameranın hareket etmesi ve nesnenin sahne tarafından kısmi engellenmesidir. Bu videoya ait takip sonuçları şekil 4.7.'de gösterilmiştir. 1 numaralı çerçevede seçilen nesne 145-215 numaralı çerçeveler arasında sahne tarafından kısmi olarak engellendiği için Bhattacharyya katsayısı şekil 4.8.'de görüldüğü gibi bu çerçeveler arasında azalmıştır. Şekil 4.9.'da ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Ortalama kayma algoritması bu video dizisi için başarılı sonuçlar üretmiştir.

#### **4.1.4. Rubik's Cube Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne bir rubik küptür. Karşılaşılan zorluklar: Kameranın hareketi ve nesnenin yüzeylerinin farklı renk dağılımına sahip olmasıdır. Bu videoya ait takip sonuçları şekil 4.10.'da gösterilmiştir. 1 numaralı çerçevede seçilen nesnenin, üç yüzeyi de görüntü düzleminde görünürken, 228 numaralı çerçevede kameranın dönmesiyle beraber iki yüzeyi görünürdür. Nesne 228. çerçevede farklı görünüme sahip olmasından ötürü şekil 4.11.'de görüldüğü gibi Bhattacharyya katsayısı minimum seviyededir. Şekil 4.12.'de ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Ortalama kayma algoritması bu video dizisi için başarılı sonuçlar üretmiştir.

#### **4.1.5. Cup Sequence Videosu İçin Ortalama Kayma Algoritmasının Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne bir kupadır. Karşılaşılan zorluklar: Kameranın hareketi, nesnenin hareketi, arka planın hedef nesneyle benzer



renk özelliklerine sahip olması ve nesne ölçeğinin değişmesidir. Bu videoya ait takip sonuçları şekil 4.13.'te gösterilmiştir. 1 numaralı çerçevede hedef nesne seçilerek takip işlemine başlanmıştır. 166 numaralı çerçevede hedef nesnenin kameradan uzaklaşması, parlaklığının değişmesi ve az da olsa dönmesi şekil 4.14'te görüldüğü gibi minimum Bhattacharyya katsayısının bu çerçevede üretilmesine yol açmıştır. 203-237 numaralı çerçeveler arasında ise arka plan hedef nesneyle benzer renk özelliklerine sahiptir. Şekil 4.13.'te görüldüğü gibi 231 numaralı çerçevede, fazlaca arka plan görüntüsü de hedef nesne olacak şekilde takip sonucu üretilmiştir. Şekil 4.15.'te hedef nesnenin x ve y koordinatlarına ait gerçek değer ve ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Ortalama kayma algoritması bu videoda hedef nesne olan kupayı başarılı bir şekilde takip edebilmiştir.

## **4.2. Önerilen Yöntem ve Ortalama Kayma Algoritmasının Nesne Takibinde Kullanılması**

Bu bölümde önerilen yöntemin ortalama kayma algoritmasına üstünlüğünü sergileyebilmek için çizelge 4.1.'de özellikleri belirtilen ball sequence, juice box sequence, person floor sequence, person fully occluded sequence videoları kullanılmıştır. Önerilen yöntemin üstünlükleri irdelenmiştir.

### **4.2.1. Ball Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne bir futbol topudur. Karşılaşılan zorluklar: Kameranın hareketi, hareketli nesne, nesnenin kameraya yaklaşıp uzaklaşması ve nesnenin hızlı bir şekilde hareket etmesidir. Bu video için ortalama kayma algoritması ve önerilen yöntem 2 kez kıyaslanmıştır. İlk kıyaslama hedef nesneye ait ölçek bilgisinin değiştirilmediği, 2. kıyaslamaysa ölçek bilgisinin güncellendiği koşullarda yapılmıştır.

Ölçeklendirme göz önüne alınmadığı durum için 1 numaralı çerçevede seçilen hedef nesnenin standart ortalama kayma algoritması kullanılarak elde edilen takip sonuçları şekil 4.16.'da sunulmuştur. Nesne, 183-190 çerçeveleri arasında ani hareket ettiği için standart ortalama kayma algoritması nesneyi 188. çerçevede tamamiyle kaybedip videonun sonuna kadar takip edememektedir. Kameranın da hareketinden

dolayı şekil 4.17’de 188. çerçeveden sonra Bhattacharyya katsayısına ait değerlerde büyük salınımlar gözlenmektedir. Şekil 4.18.’de ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve standart ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Yine 188. çerçeveden sonra hedef nesneye ait hesaplanan pozisyon ve gerçek pozisyon değerleri arasında büyük fark bulunmaktadır. Standart ortalama kayma algoritması bu video için başarılı sonuç üretememiştir.

Buna karşılık ölçeklendirme işlemi göz önüne alınmadan önerilen yöntemle elde edilen takip sonuçları şekil 4.19.’da sunulmuştur. Hedef nesne olan futbol topunun ani hareket etmesine rağmen, önerilen yöntemdeki Kalman filtresine ait hareket modeli sayesinde nesne videonun sonuna kadar başarılı bir şekilde takip edilebilmiştir. Şekil 4.20.’de takip işlemi boyunca elde edilen Bhattacharyya katsayısına ait değerler sunulurken şekil 4.21.’de ise hedef nesnenin x ve y koordinatlarına ait gerçek değerler ve önerilen yöntemin hesapladığı değerler sunulmuştur.

Ölçeklendirmenin göz önüne alındığı durum için standart ortalama kayma algoritmasının takip sonuçları şekil 4.22.’de sunulmuştur. Şekil 4.23.’te Bhattacharyya katsayısına ait değerler sunulurken şekil 4.24.’te ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve standart ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Standart ortalama kayma algoritması nesnenin ani hareket ettiği 183-190 çerçeveleri arasında nesneyi gözden kaçırmaktadır ve bu çerçevelerden sonra nesneyi takip edememektedir.

Buna karşılık önerilen yöntemin takip sonuçları şekil 4.25.’te görülmektedir. Nesnenin ani hareket etmesine rağmen, önerilen yöntemin nesneyi başarılı bir şekilde takip edebildiği gözlenmektedir. Şekil 4.26.’da takip işlemi boyunca elde edilen Bhattacharyya katsayısına ait değerler sunulurken şekil 4.27.’de ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve önerilen yöntemin hesapladığı değerler sunulmuştur.

#### **4.2.2. Juice Box Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne bir meyve suyu kutusudur. Karşılaşılan zorluklar: Kameranin ani olarak hareket etmesi ve nesne ölçeğinin değişmesidir. Standart ortalama kayma algoritmasına ait takip sonuçları şekil 4.28.'de gösterilmiştir. Kameranin 240-250 çerçeveleri arasında hedef nesneden z eksenini boyunca uzaklaşması ve ani hareket etmesinden ötürü standart ortalama kayma algoritması nesneyi gözden kaçırıp takip edememektedir. Şekil 4.29.'da Bhattacharyya katsayısına ait değerler sunulurken şekil 4.30.'da ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve standart ortalama kayma algoritmasının hesapladığı değerler sunulmuştur.

Önerilen yöntemin takip sonuçları ise şekil 4.31.'de sunulmuştur. Kameranin ani hareket etmesine rağmen videonun sonuna kadar önerilen yöntem hedef nesneyi başarılı bir şekilde takip edebilmiştir. Şekil 4.32.'de takip işlemi boyunca elde edilen Bhattacharyya katsayısı değerleri sunulurken şekil 4.33.'te ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve önerilen yöntemin hesapladığı değerler sunulmuştur.

#### **4.2.3. Person Floor Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne insandır. Karşılaşılan zorluklar: Kameranin hareketi, sahnedeki ışık şiddeti dağılımının farklı olması, nesnenin; hareketi, eklemli yapıda olması ve tam engele maruz kalmasıdır. Standart ortalama kayma algoritmasının takip sonuçları şekil 4.34.'te sunulmuştur. Hedef nesne 292-297 çerçeveleri arasında tam olarak engele maruz kalmıştır. Şekil 4.35.'te Bhattacharyya katsayısına ait değerler sunulurken, 4.36.'da ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve standart ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Nesne tam engele maruz kaldıktan sonra algoritma hedef nesneyi kaybetmiştir.

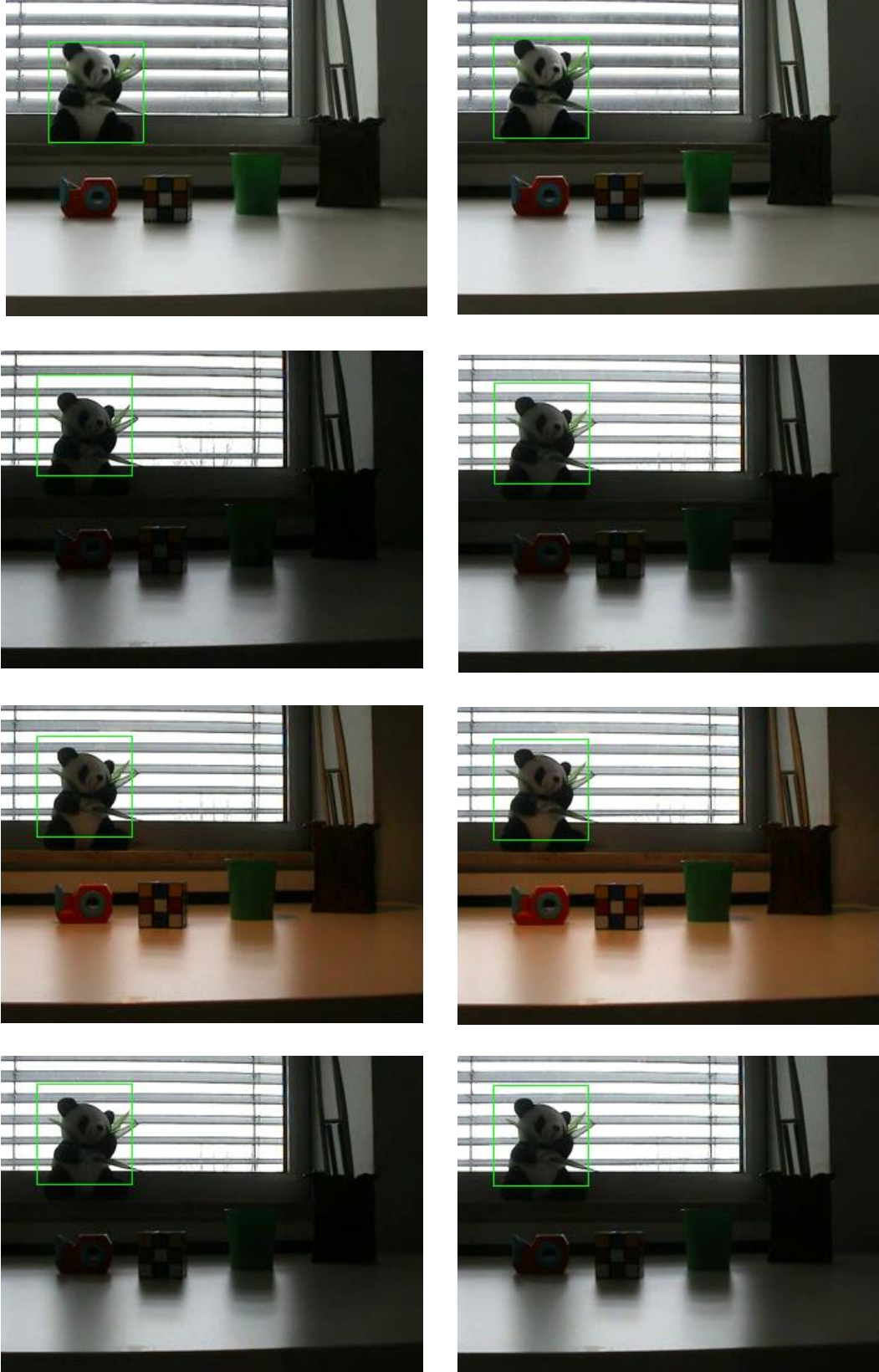
Önerilen yöntemin takip sonuçları ise şekil 4.37.'de sunulmuştur. Nesnenin tam olarak engellenmesine rağmen videonun sonuna kadar önerilen yöntem hedef

nesneyi başarılı bir şekilde takip edebilmiştir. Şekil 4.38.'de Bhattacharyya katsayısına ait değerler sunulurken şekil 4.39.'da ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve önerilen yöntemin hesapladığı değerler sunulmuştur. Şekil 4.39.'daki y eksenini için hedef nesneye ait gerçek pozisyon ve önerilen yöntemin hesapladığı pozisyon bilgisi bazı çerçeveler arasında sapma göstermektedir. Aynı durum şekil 4.36.'da da gözlenmektedir. Hedef nesnenin kameradan z eksenini boyunca uzaklaşması sonucu görüntü düzleminde kapladığı alan küçülmektedir. Bu videoda ise ölçekleme işlemi yapılmadan takip işlemi gerçekleştirilmiştir. Bunun nedeni ortalama kayma algoritması eklemli yapıdaki nesnelere ve en-boy farkı fazla olan nesnelere için ölçeklendirme işlemi yapıldığı durumda iyi sonuçlar üretememektedir.

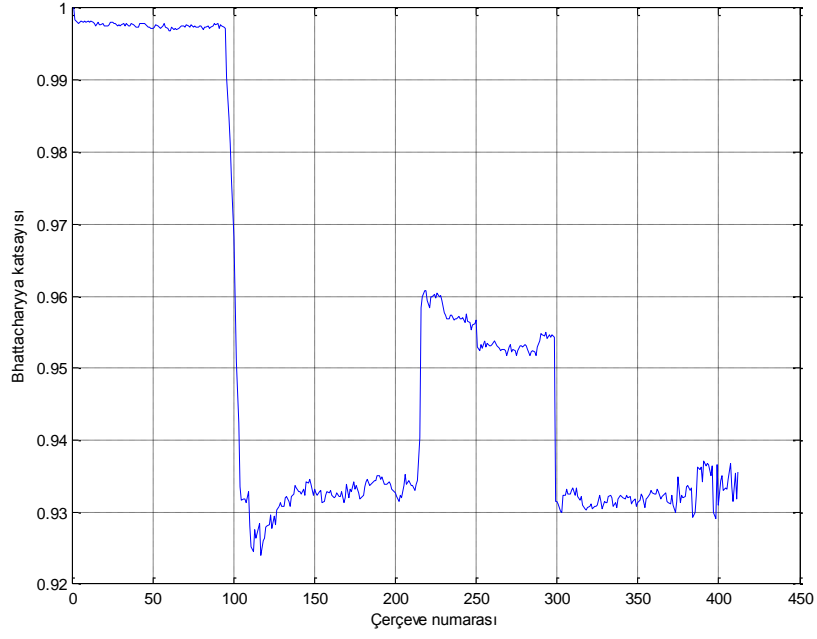
#### **4.2.4. Person Fully Occluded Sequence Videosu İçin Ortalama Kayma Algoritması ve Önerilen Yöntemin Kullanımı**

Bu videoda takip edilmesi istenen hedef nesne insandır. Karşılaşılan zorluklar: Kameranın hareketi, nesnenin; hareketli, eklemli yapıda olması ve tam engele maruz kalmasıdır. Takip işlemi nesne 58. çerçevede seçilerek başlatılmıştır. Standart ortalama kayma algoritması kullanılarak elde edilen takip sonuçları şekil 4.40.'ta sunulmuştur. Hedef nesne 143-154 çerçeveleri arasında tam olarak engele maruz kalmıştır. Şekil 4.41.'de Bhattacharyya katsayısına ait değerler sunulurken şekil 4.42.'de ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve standart ortalama kayma algoritmasının hesapladığı değerler sunulmuştur. Nesne tam engele maruz kaldıktan sonra algoritma hedef nesneyi kaybetmiştir.

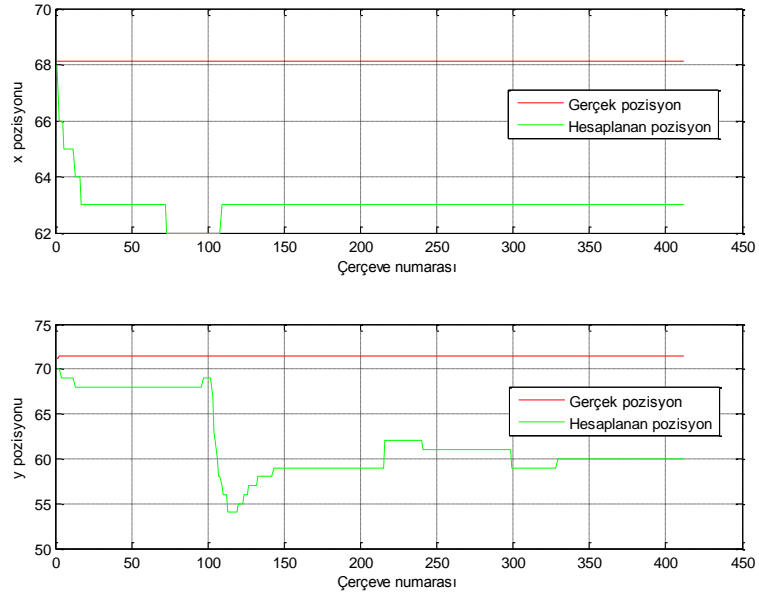
Önerilen yöntemin takip sonuçları ise şekil 4.43.'te sunulmuştur. Nesnenin tam olarak engellenmesine rağmen önerilen yöntem hedef nesneyi başarılı bir şekilde takip edebilmiştir. Şekil 4.44.'te Bhattacharyya katsayısına ait değerler sunulurken şekil 4.45.'te ise hedef nesnenin x ve y koordinatlarına ait gerçek değer ve önerilen yöntemin hesapladığı değerler sunulmuştur.



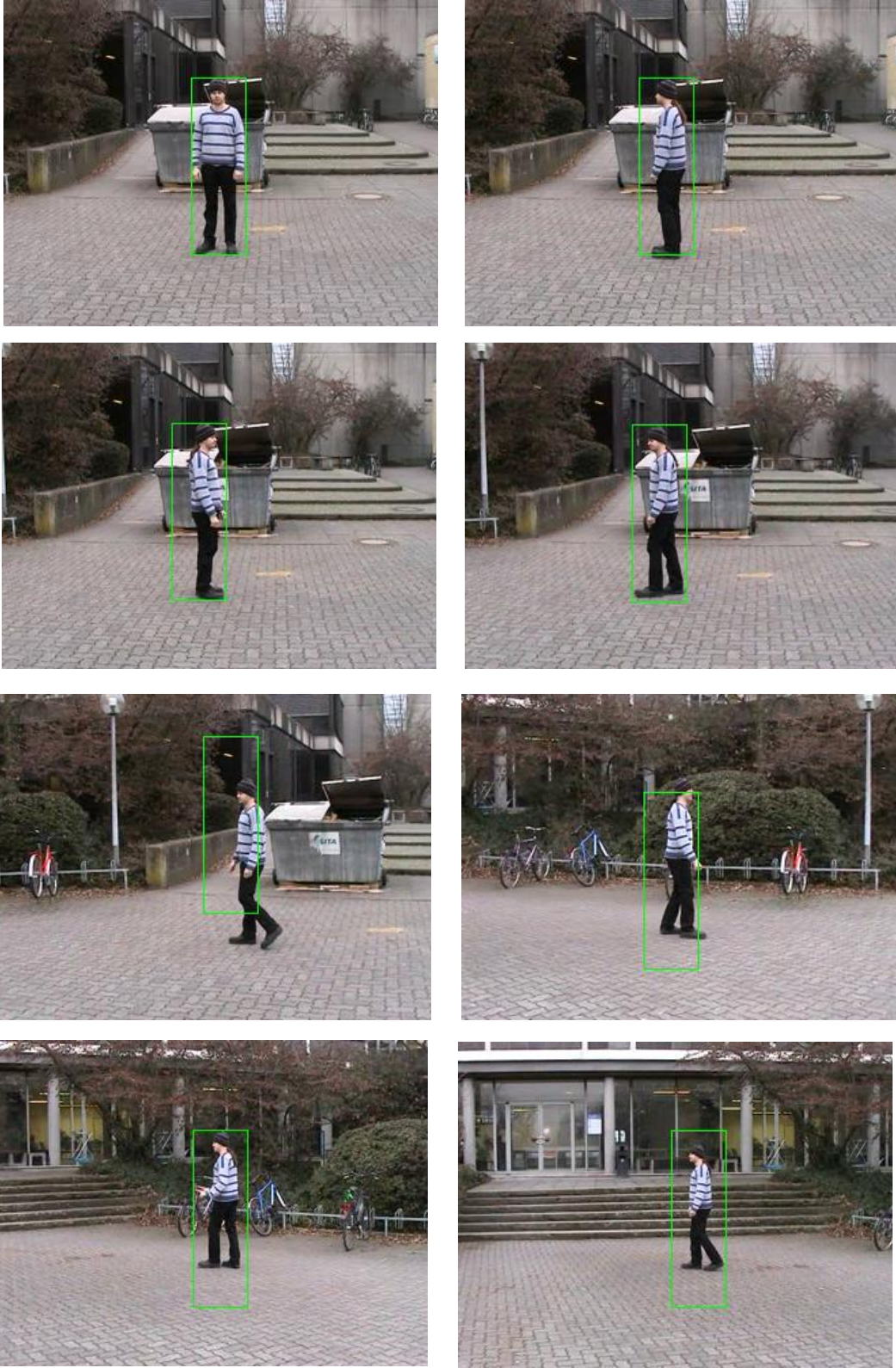
Şekil 4.1. Panda sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 50, 110, 170, 220, 260, 310, 412 numaralı çerçevelere ait takip sonuçları



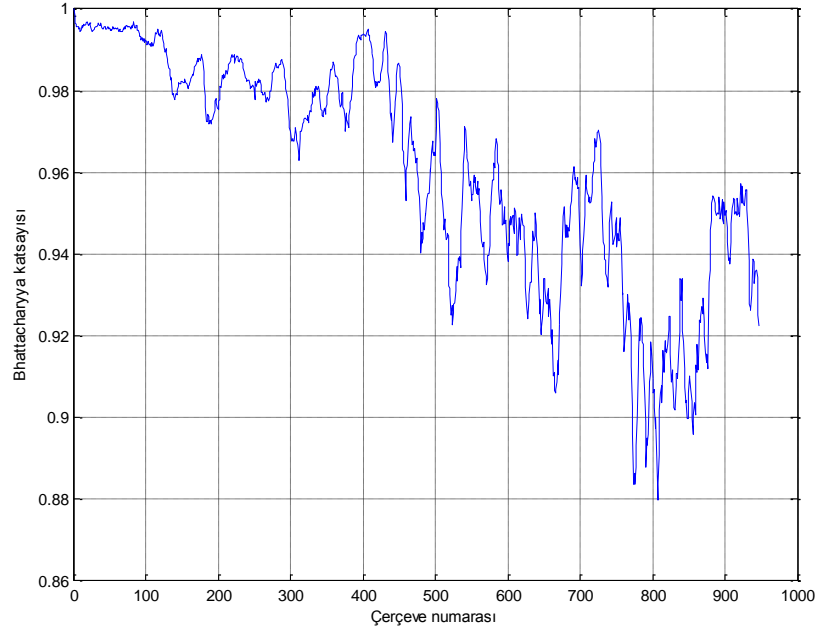
Şekil 4.2. Panda sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri



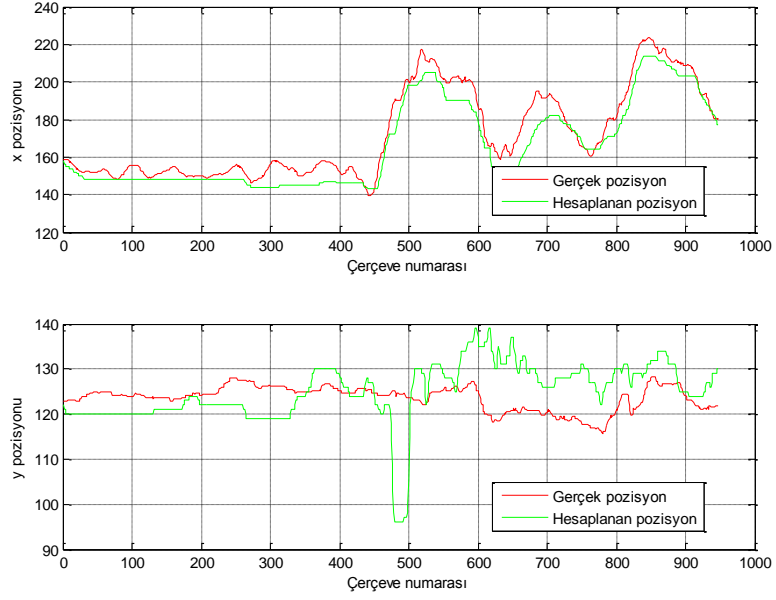
Şekil 4.3. Panda sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri



Şekil 4.4. Person sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 200, 350, 450, 480, 650, 750, 947 numaralı çerçevelere ait takip sonuçları

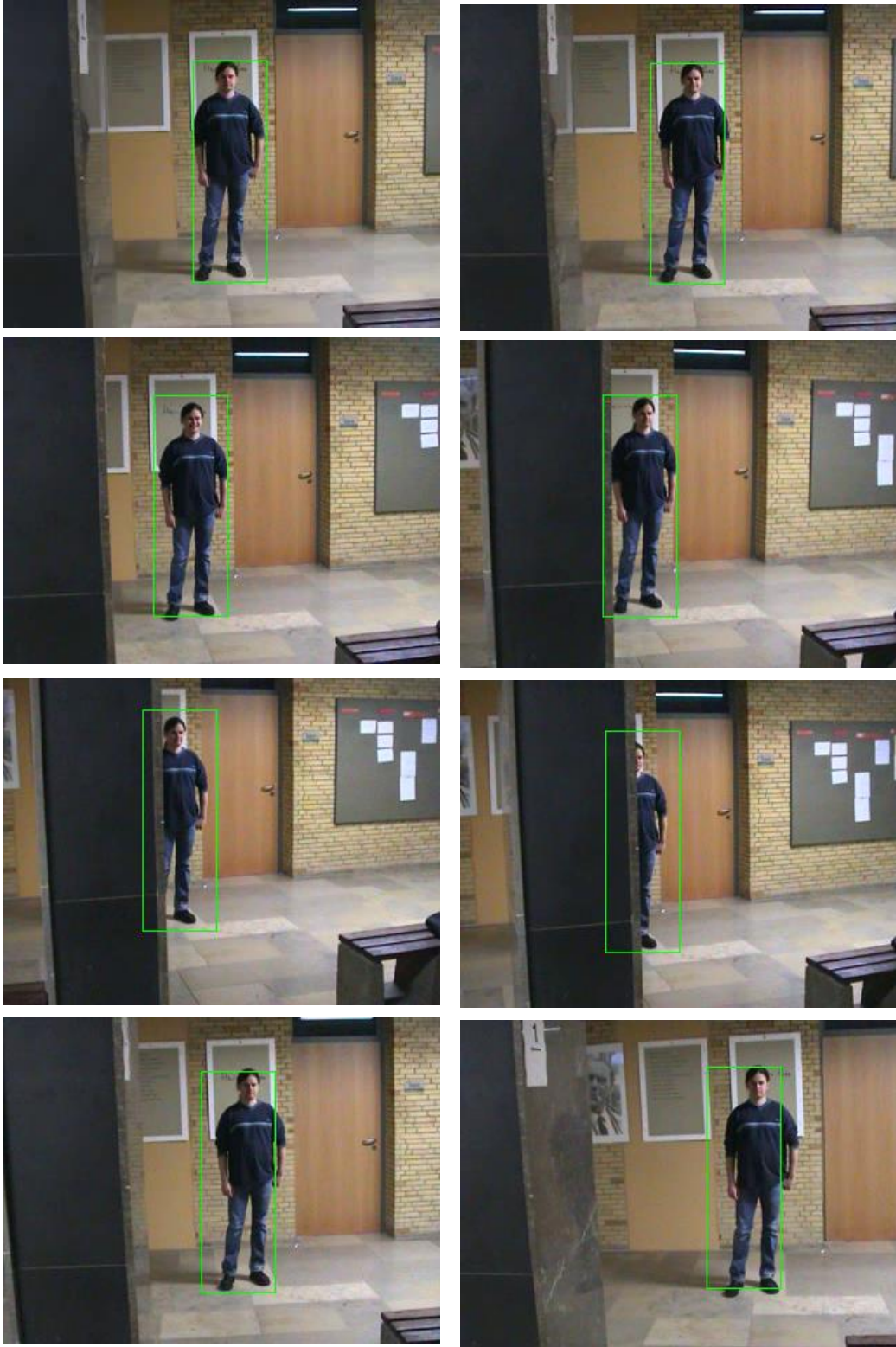


Şekil 4.5. Person sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri

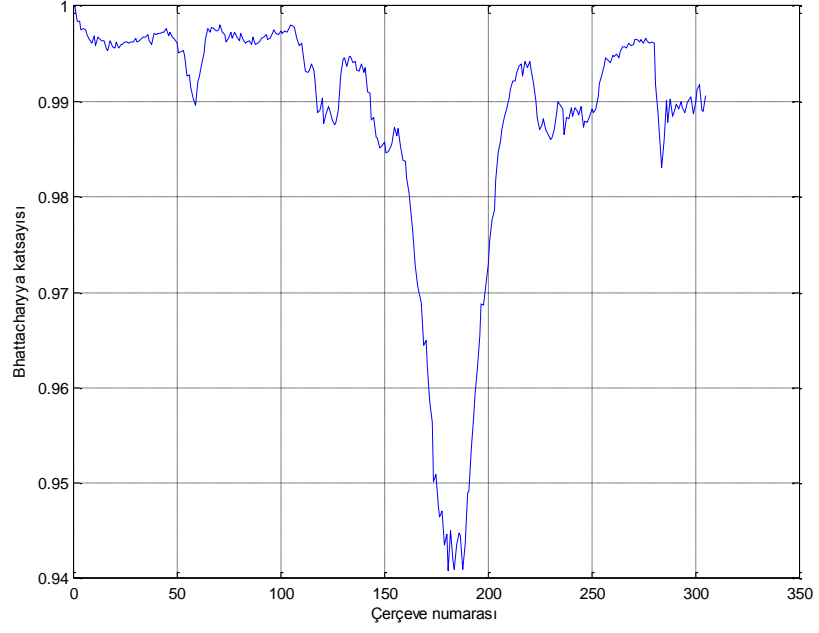


Şekil 4.6. Person sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri

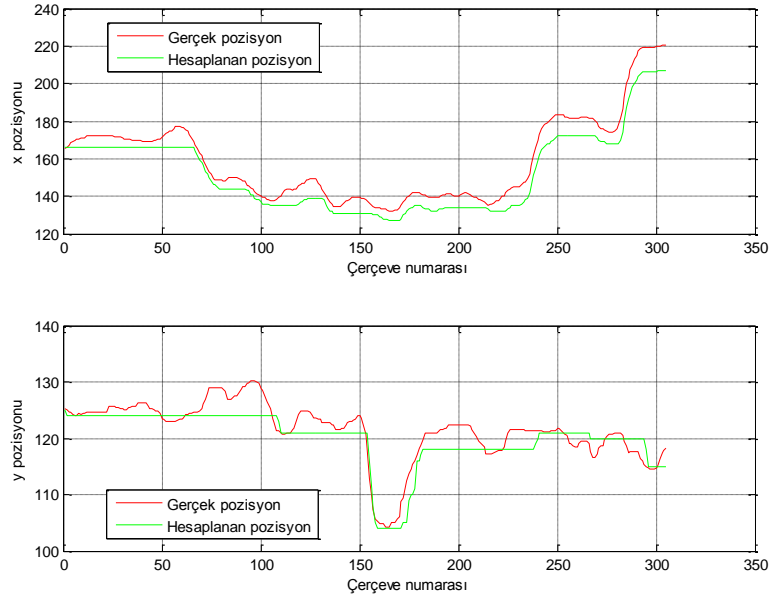




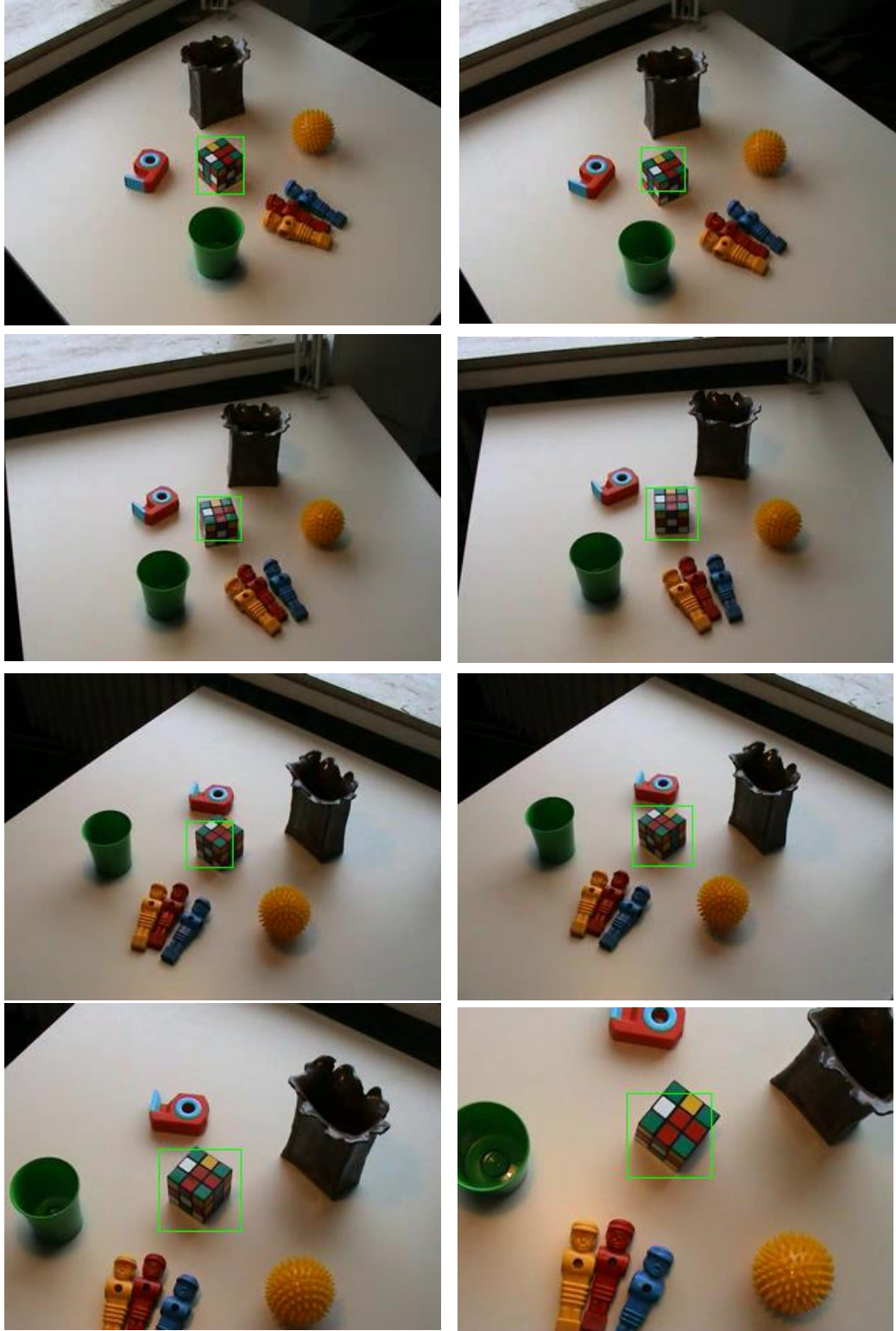
Şekil 4.7. Person partially occluded sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 50, 100, 140, 160, 190, 255, 305 numaralı çerçevelere ait takip sonuçları



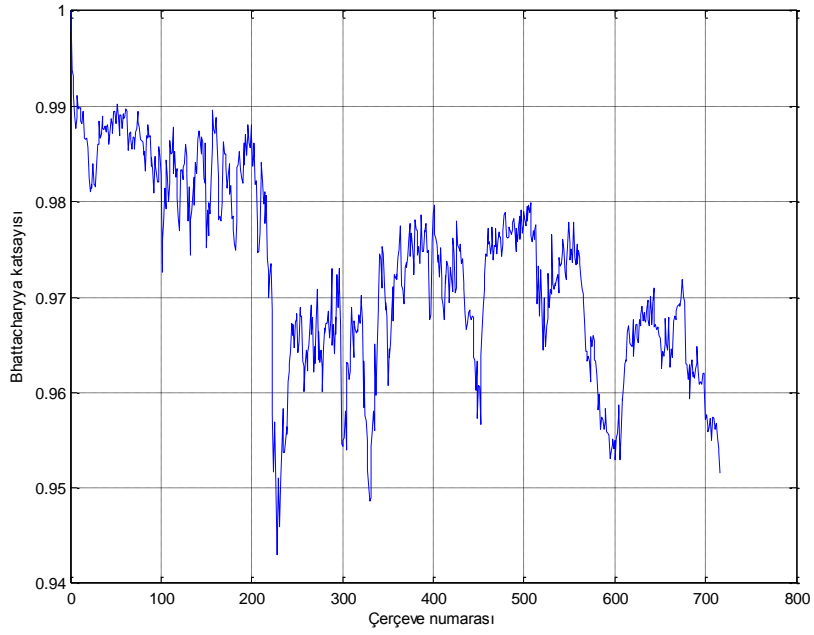
Şekil 4.8. Person partially occluded sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri



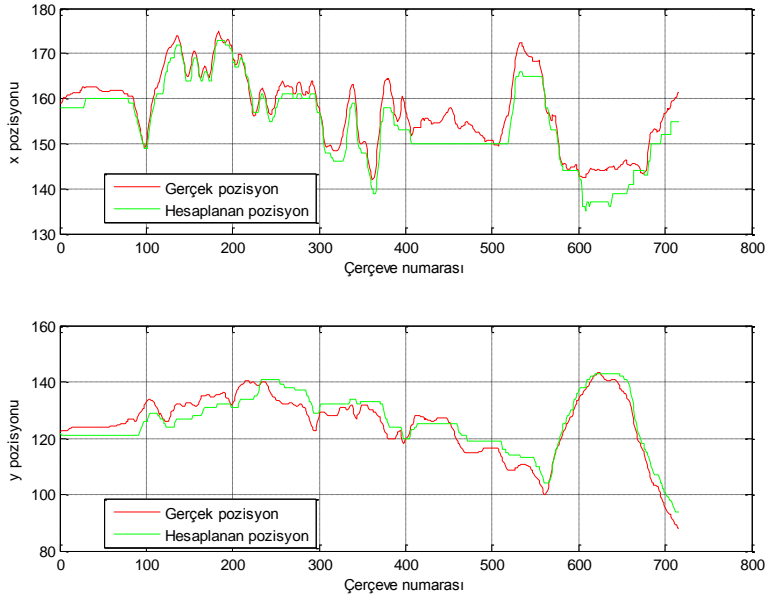
Şekil 4.9. Person partially occluded sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri



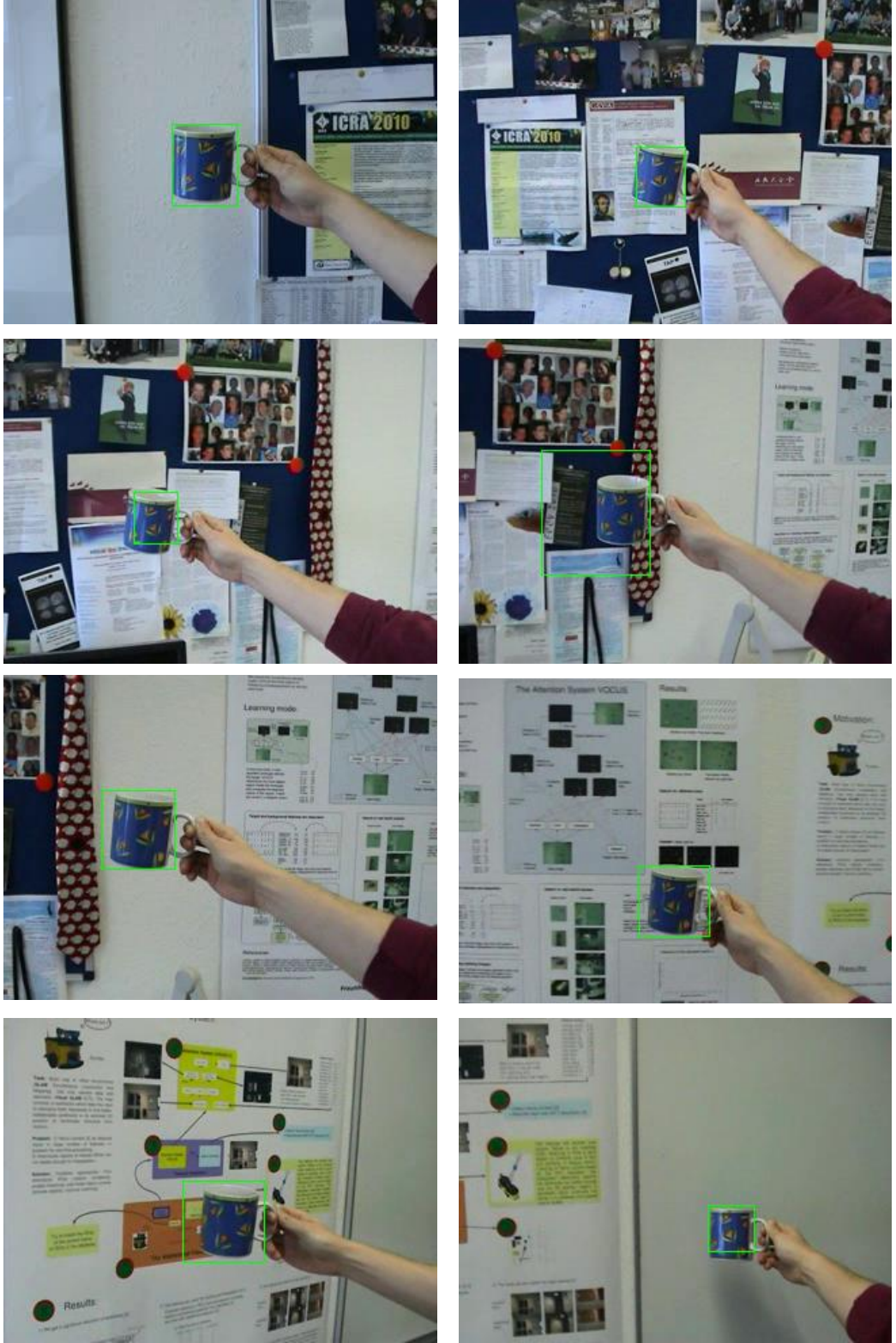
Şekil 4.10. Rubik's cube sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 101, 228, 300, 452, 500, 600, 716 numaralı çerçevelere ait takip sonuçları



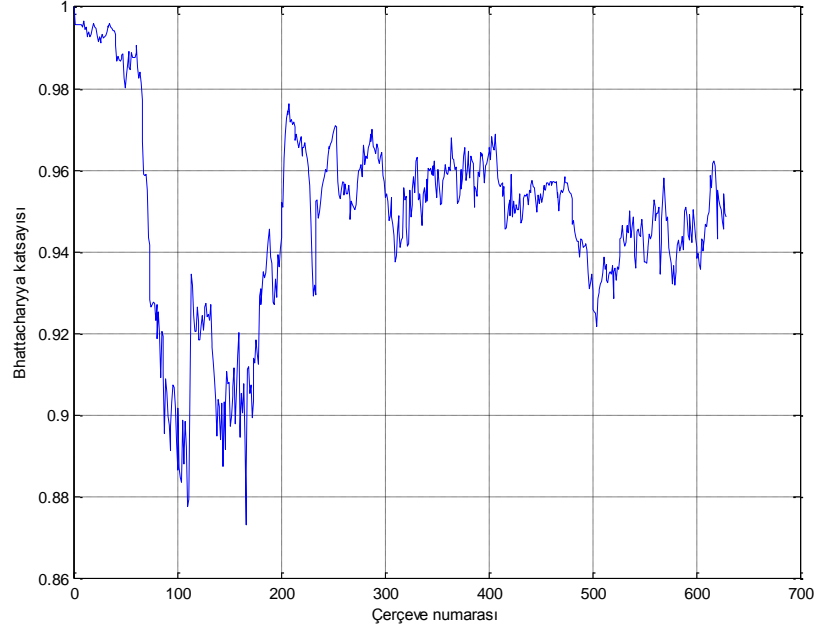
Şekil 4.11. Rubik's cube sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri



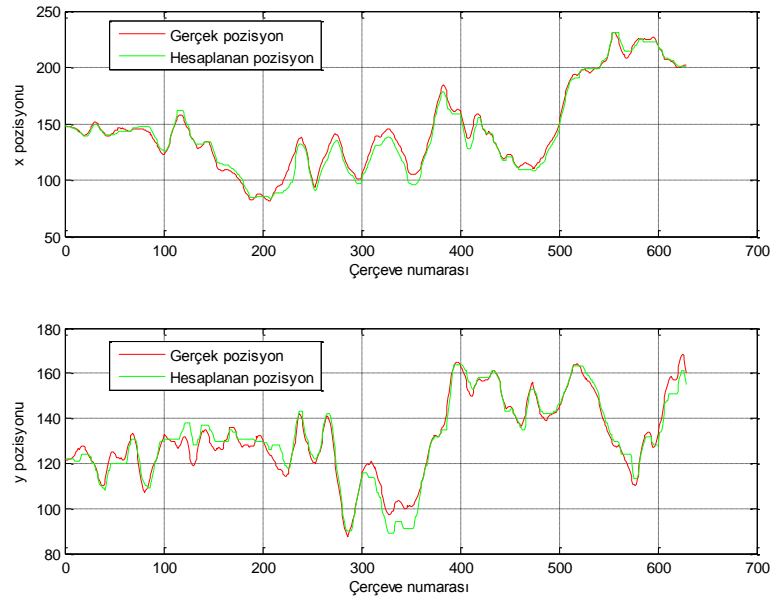
Şekil 4.12. Rubik's cube sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri



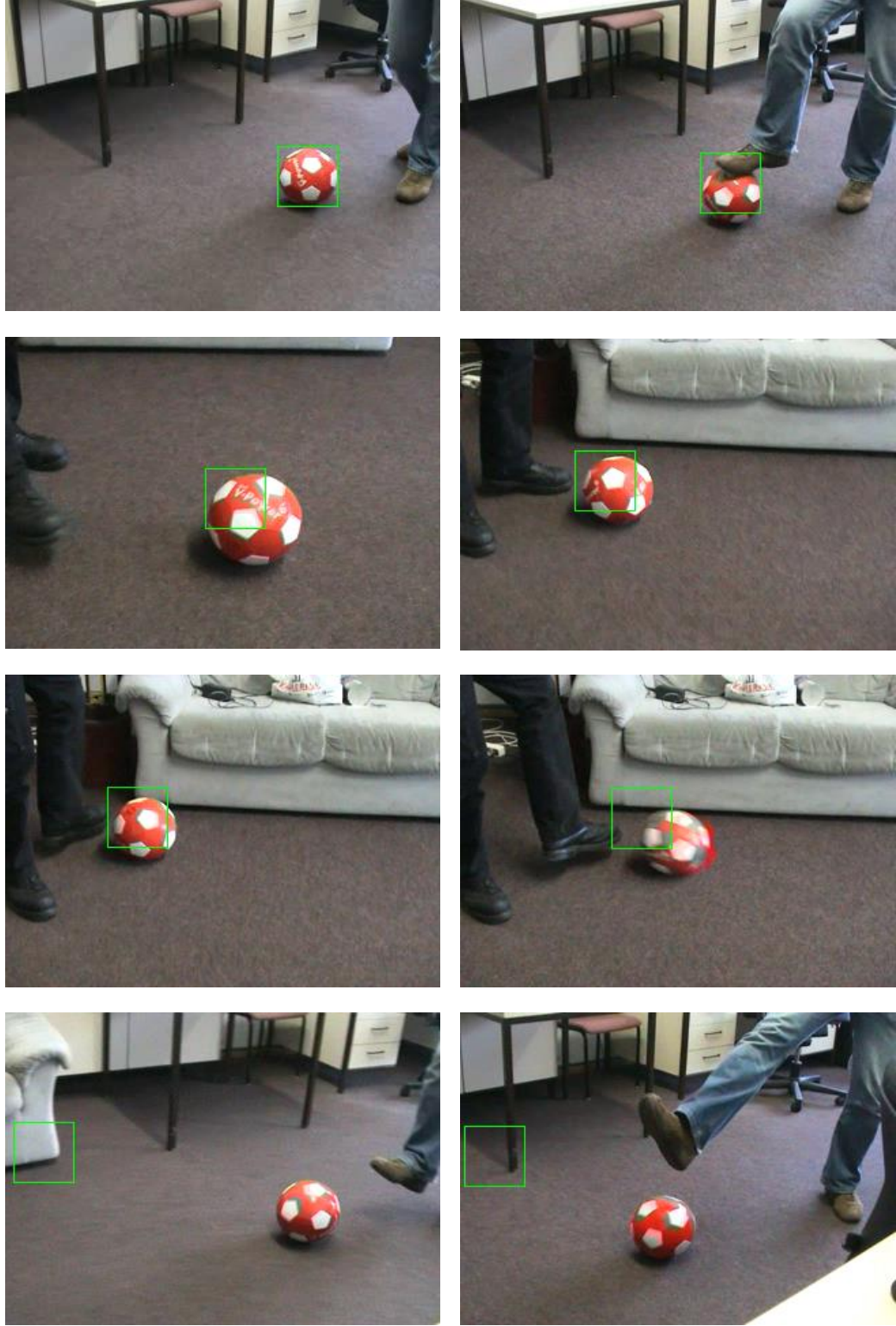
Şekil 4.13. Cup sequence videosu için soldan sağa ve yukarıdan aşağıya sırasıyla 1, 111, 166, 231, 300, 400, 504, 629 numaralı çerçevelere ait takip sonuçları



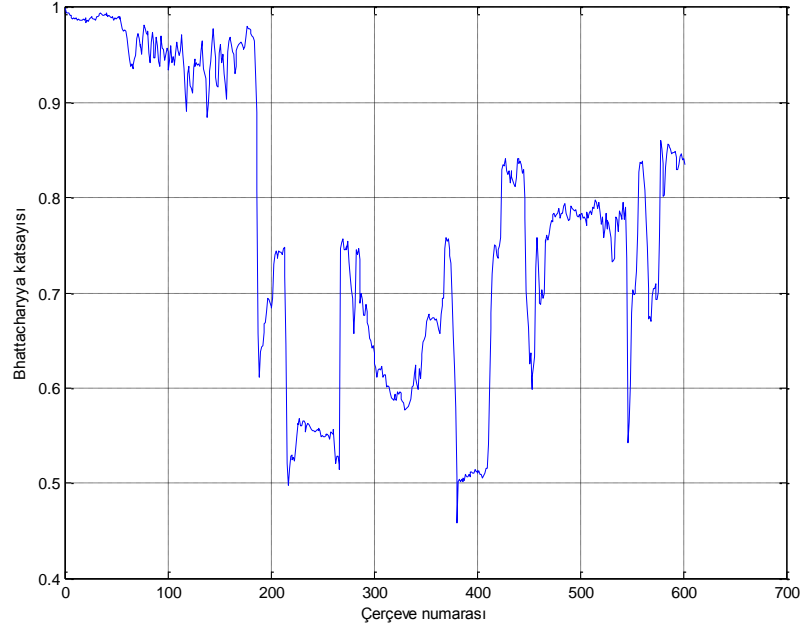
Şekil 4.14. Cup sequence videosuna ait takip işlemi boyunca Bhattacharyya katsayısı değerleri



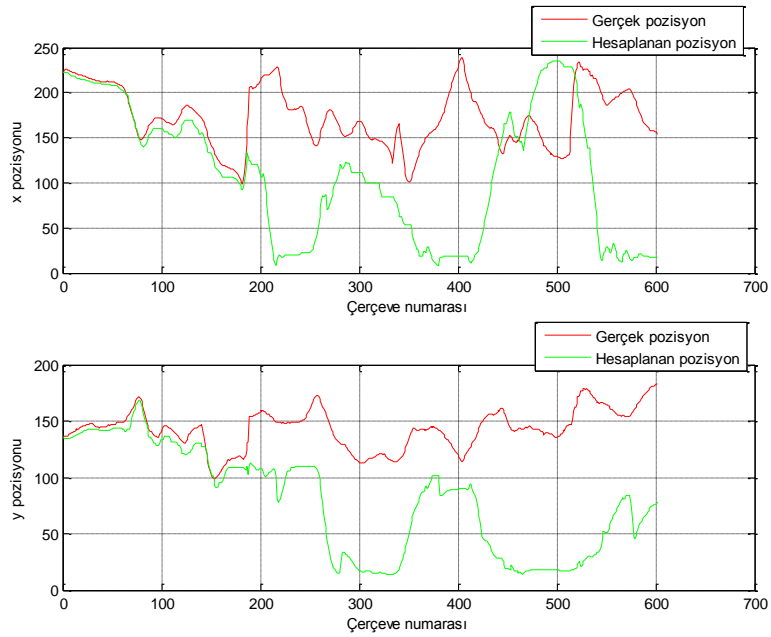
Şekil 4.15. Cup sequence videosu için gerçek pozisyon ve hesaplanan pozisyon bilgileri



Şekil 4.16. Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları

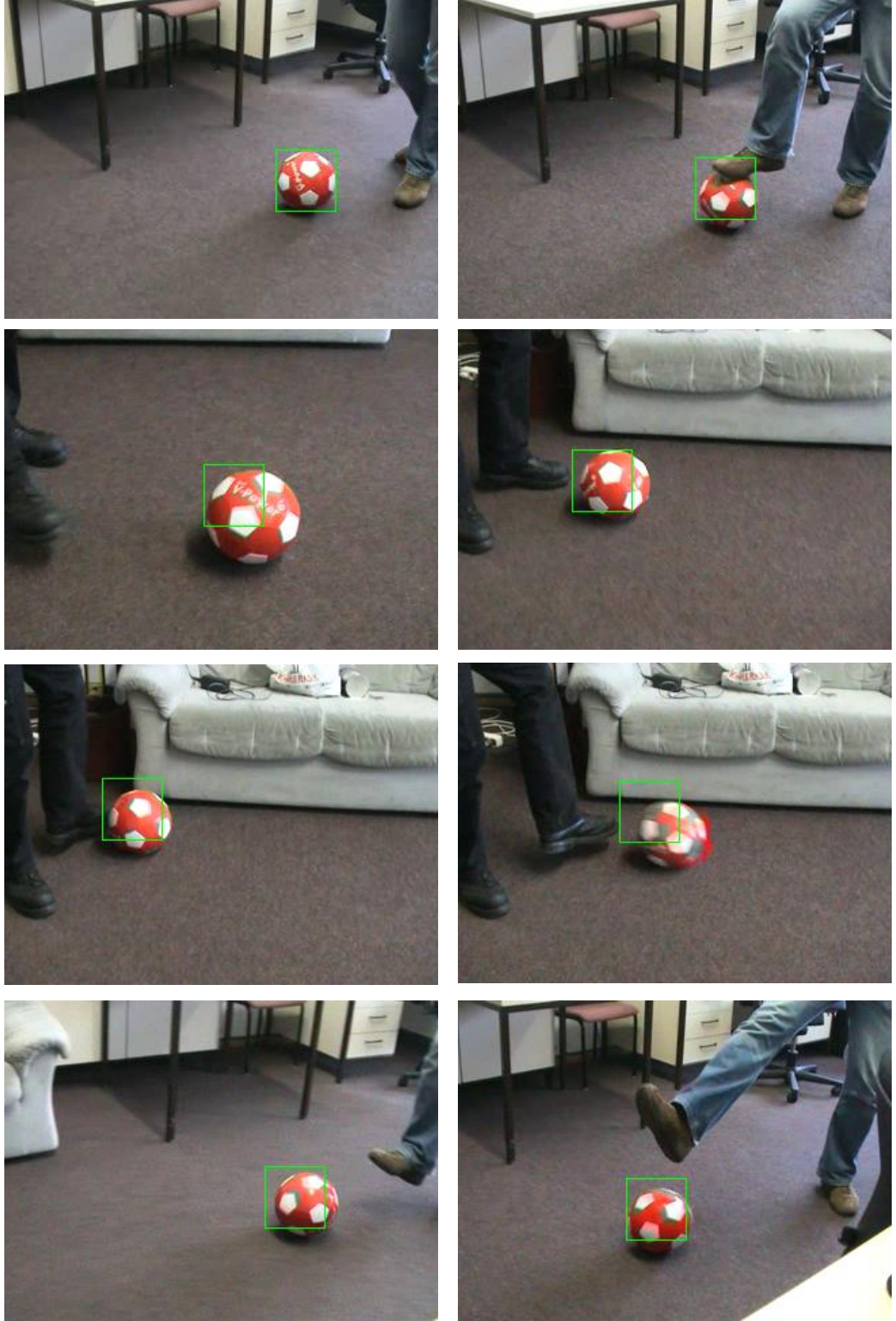


Şekil 4.17. Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri

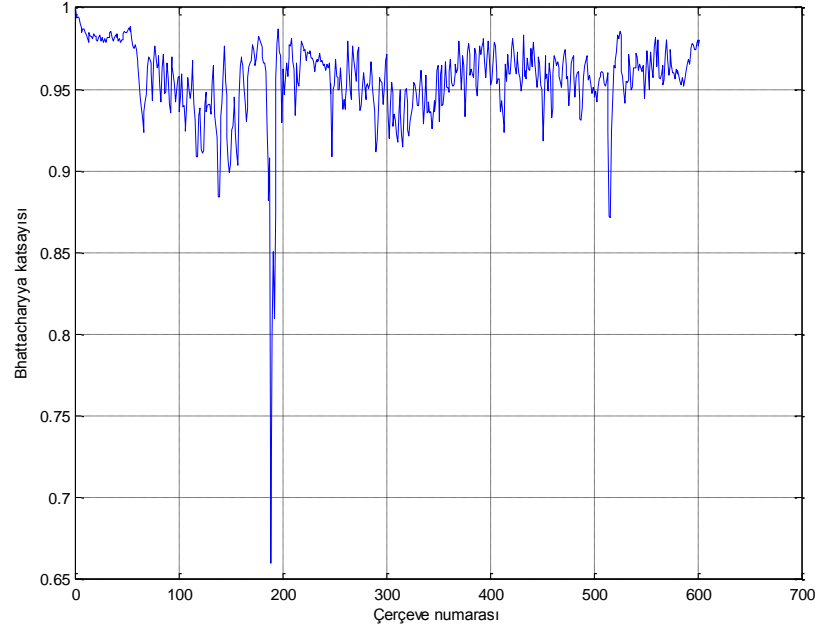


Şekil 4.18. Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapılmadığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri

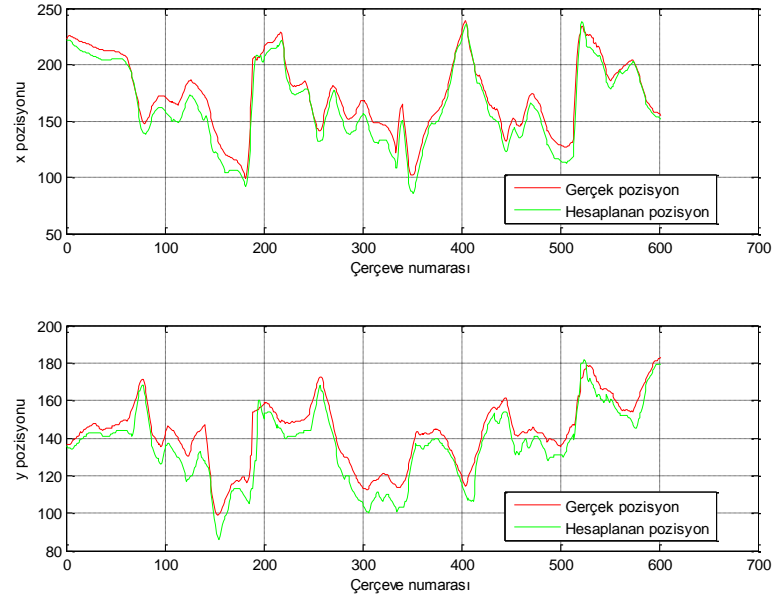




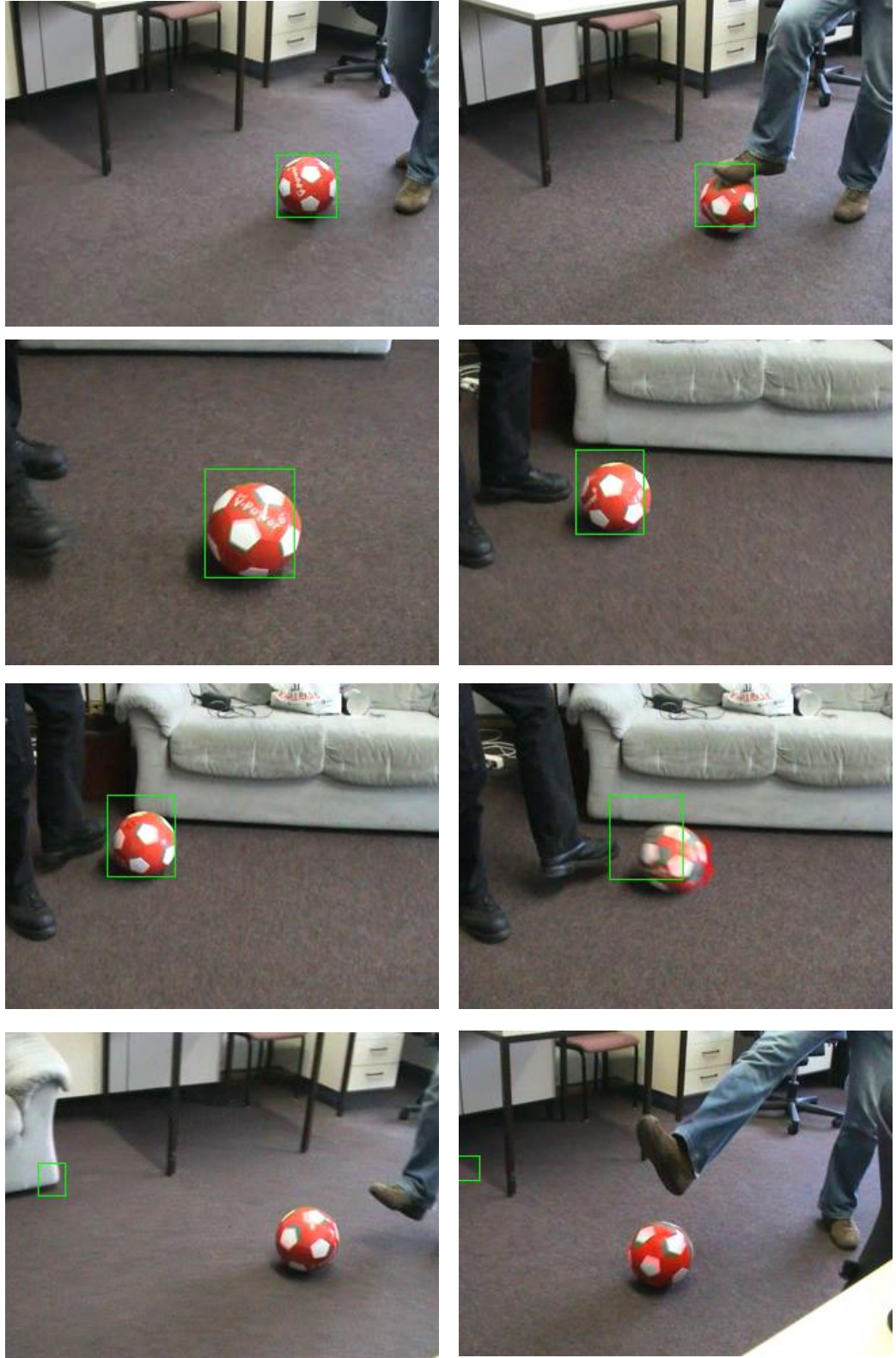
Şekil 4.19. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları



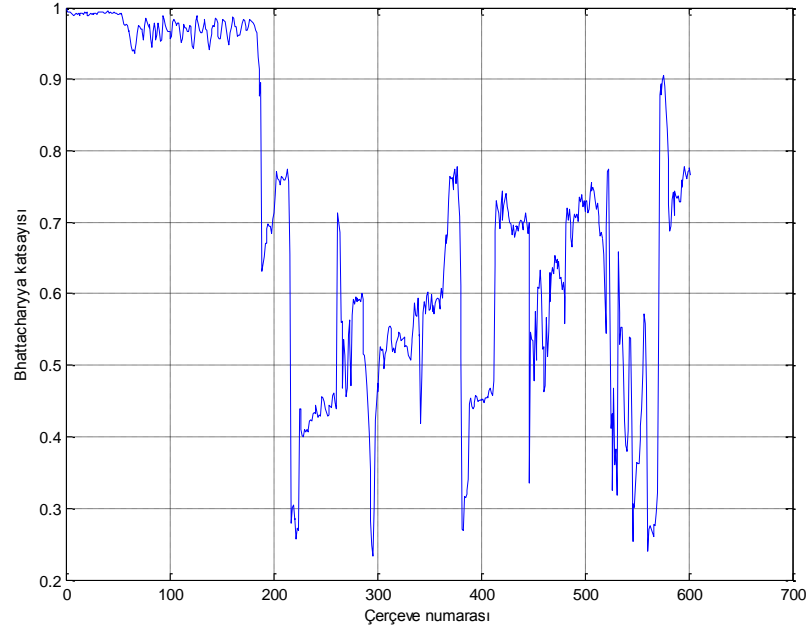
Şekil 4.20. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri



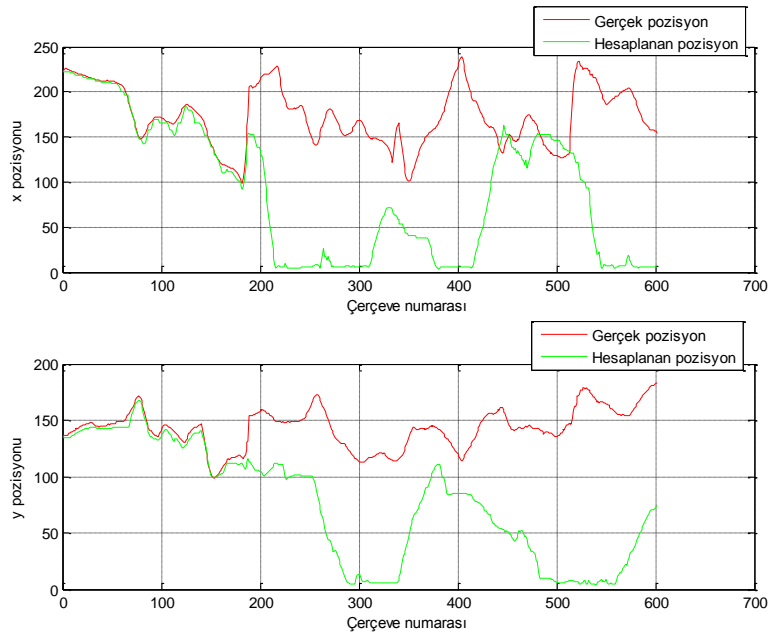
Şekil 4.21. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapılmadığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



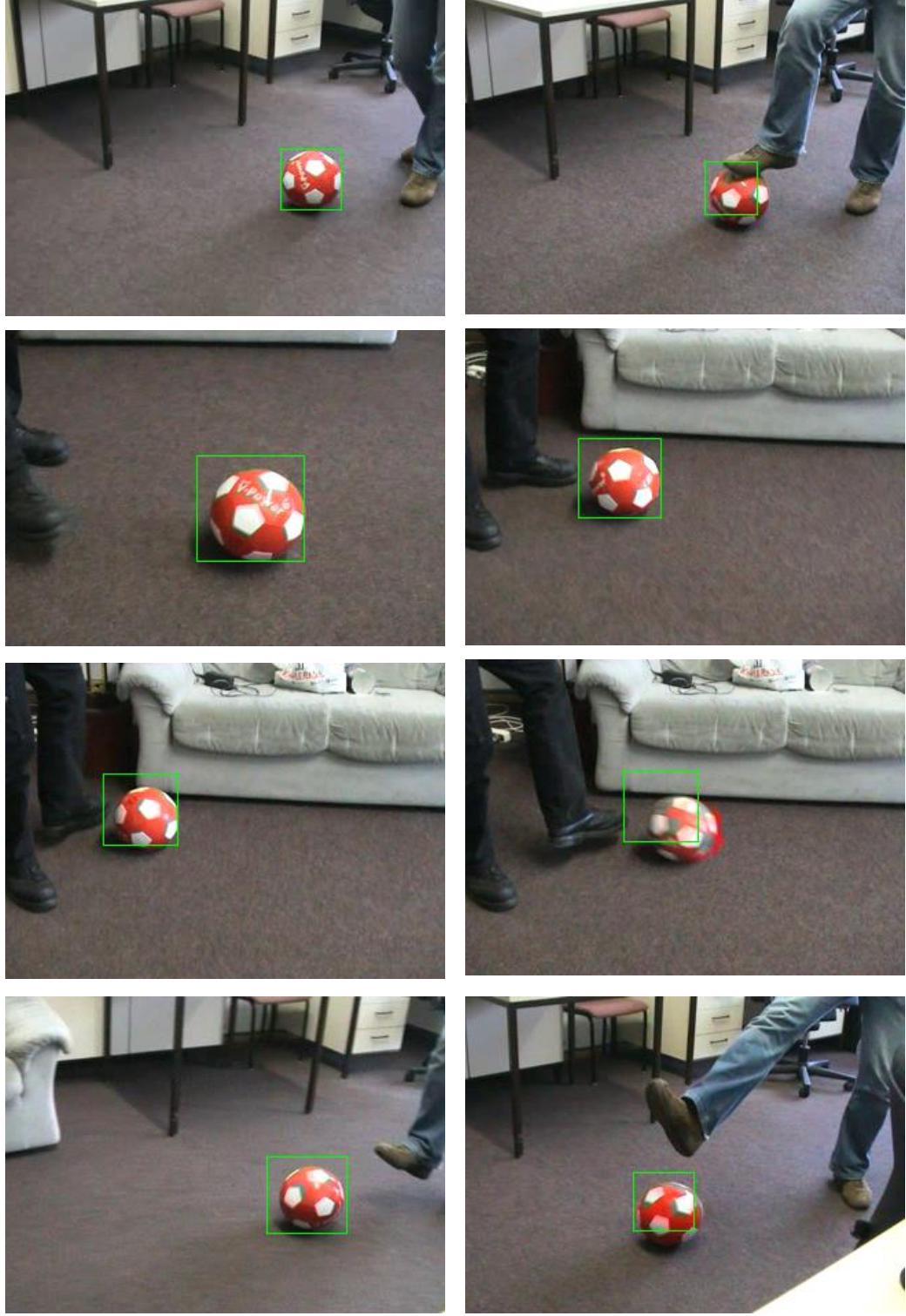
Şekil 4.22. Ball sequece videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapıldığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçvelere ait takip sonuçları



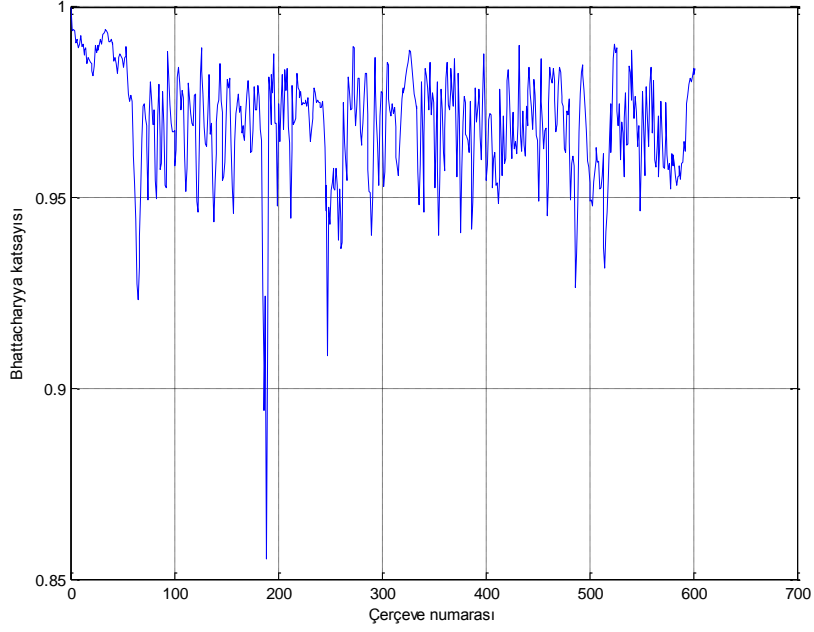
Şekil 4.23. Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapıldığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri



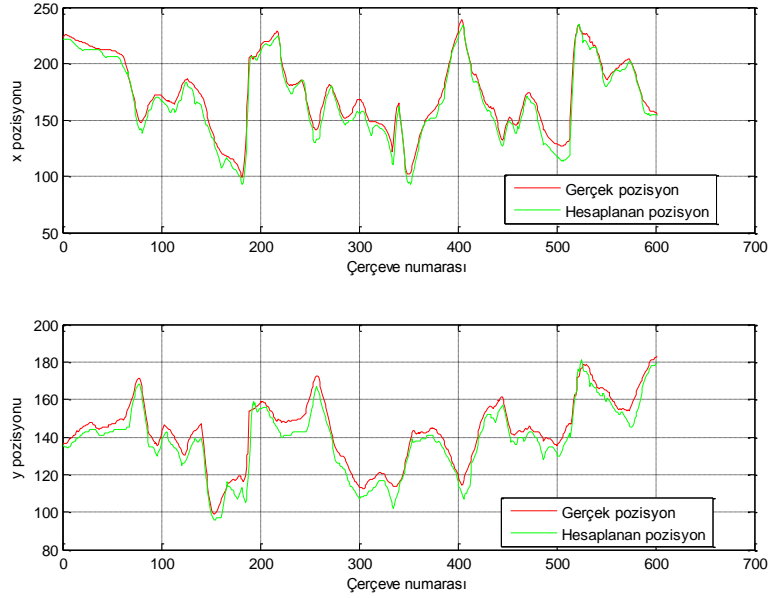
Şekil 4.24. Ball sequence videosu için standart ortalama kayma algoritması kullanarak ve ölçeklemenin yapıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



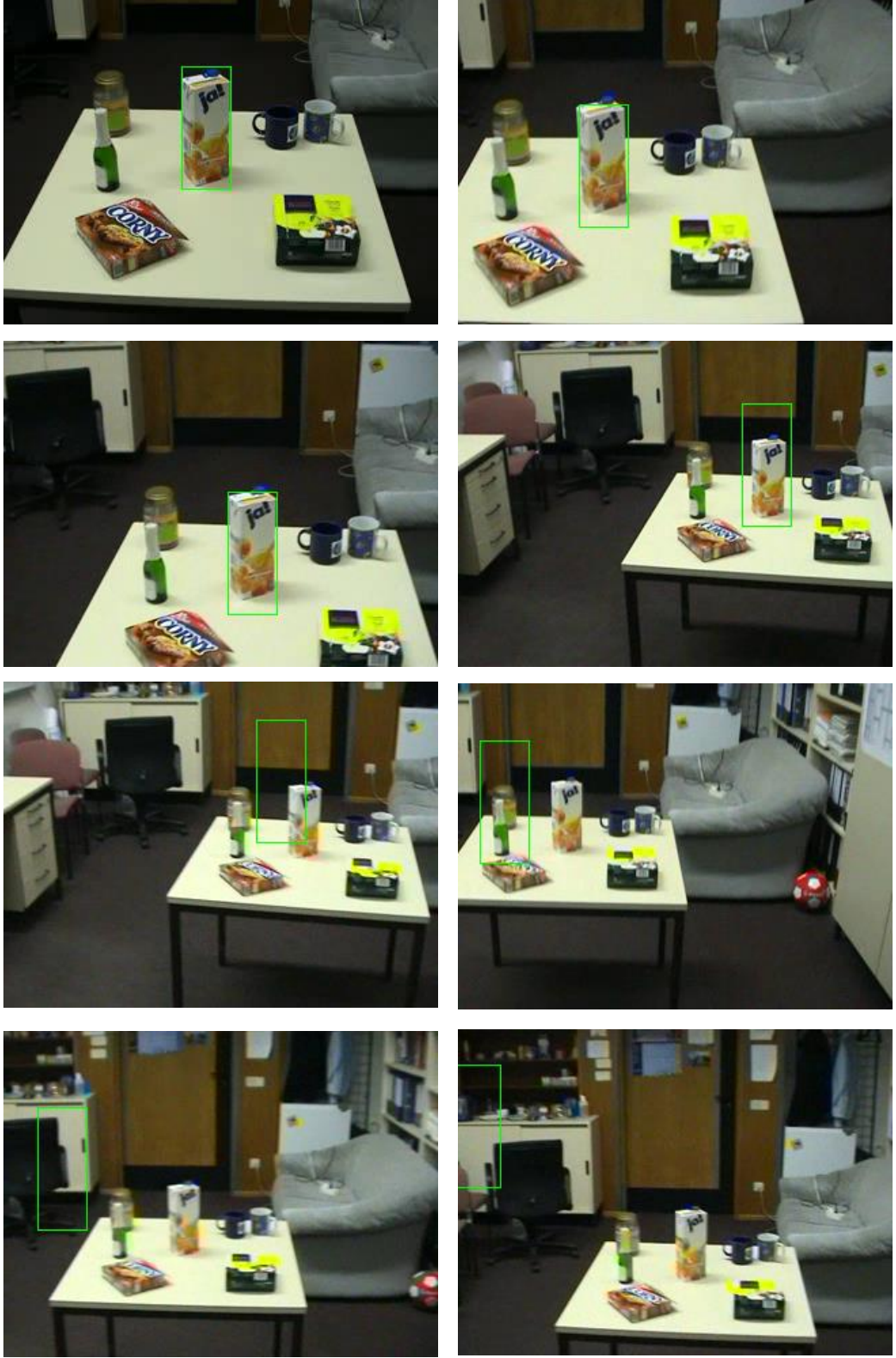
Şekil 4.25. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı soldan sağa ve yukarıdan aşağıya sırasıyla 1, 64, 130, 170, 180, 187, 212, 252 numaralı çerçevelere ait takip sonuçları



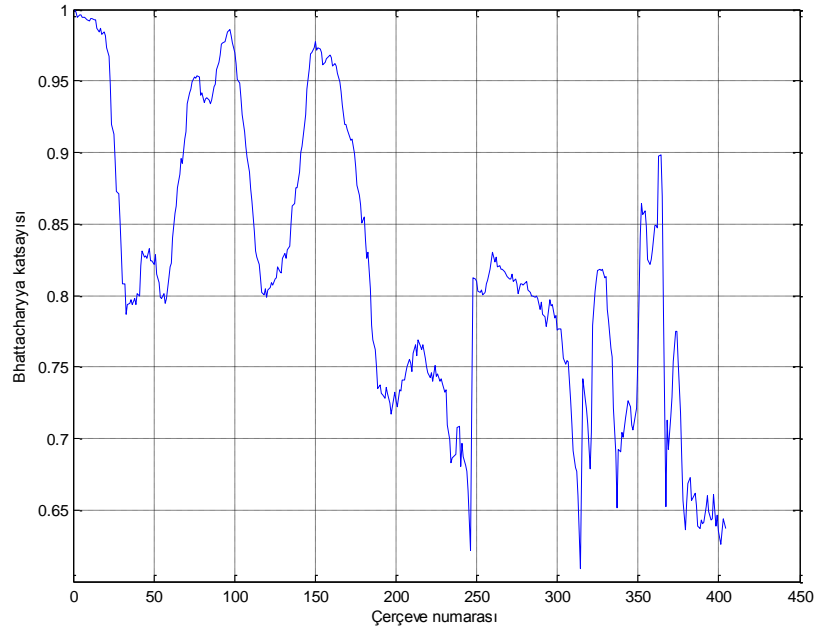
Şekil 4.26. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı durumda takip işlemi boyunca Bhattacharyya katsayısı değerleri



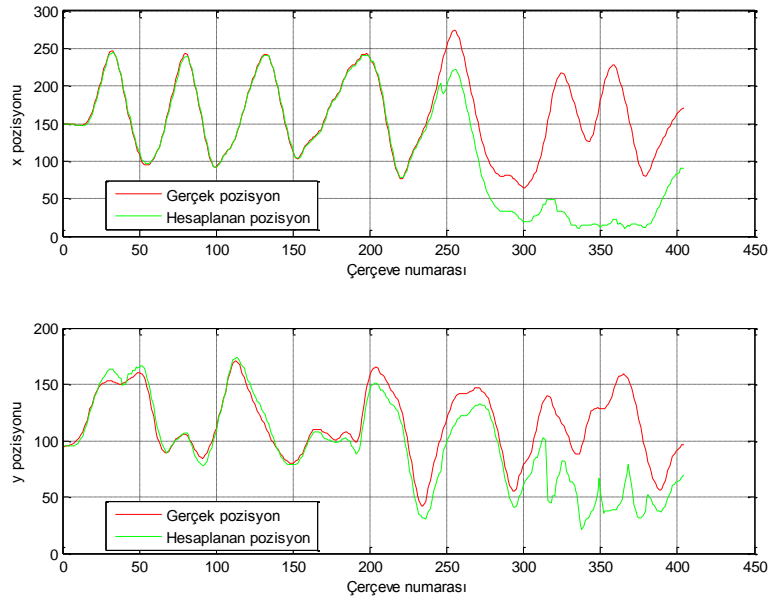
Şekil 4.27. Ball sequence videosu için önerilen yöntemi kullanarak ve ölçeklemenin yapıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



Şekil 4.28. Juice box sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 60, 120, 190, 246, 285, 314, 367 numaralı çerçevelere ait takip sonuçları

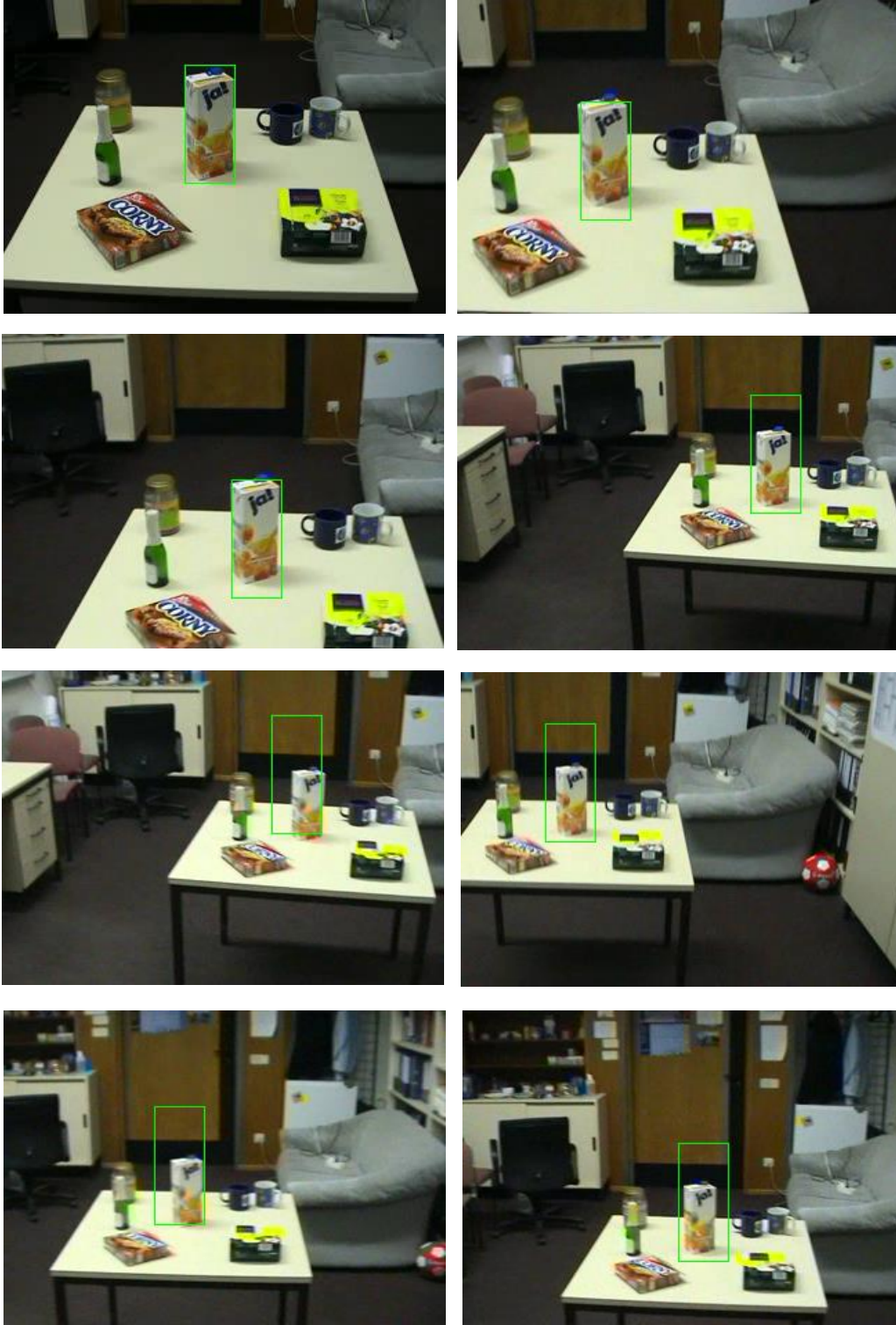


Şekil 4.29. Juice box sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri

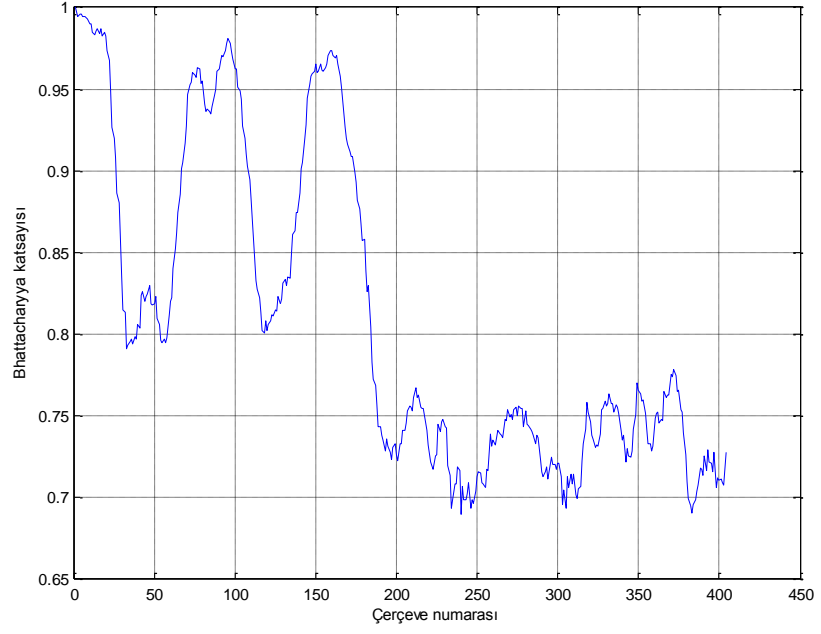


Şekil 4.30. Juice box sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri

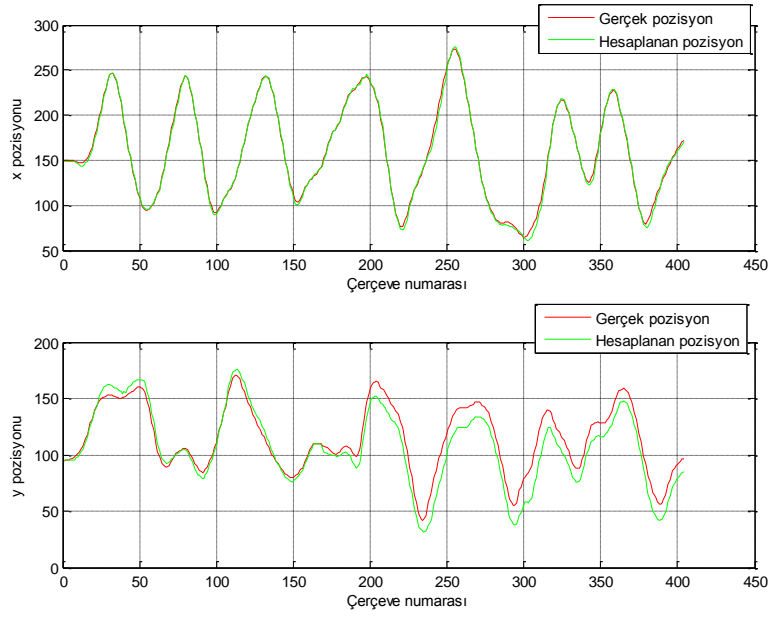




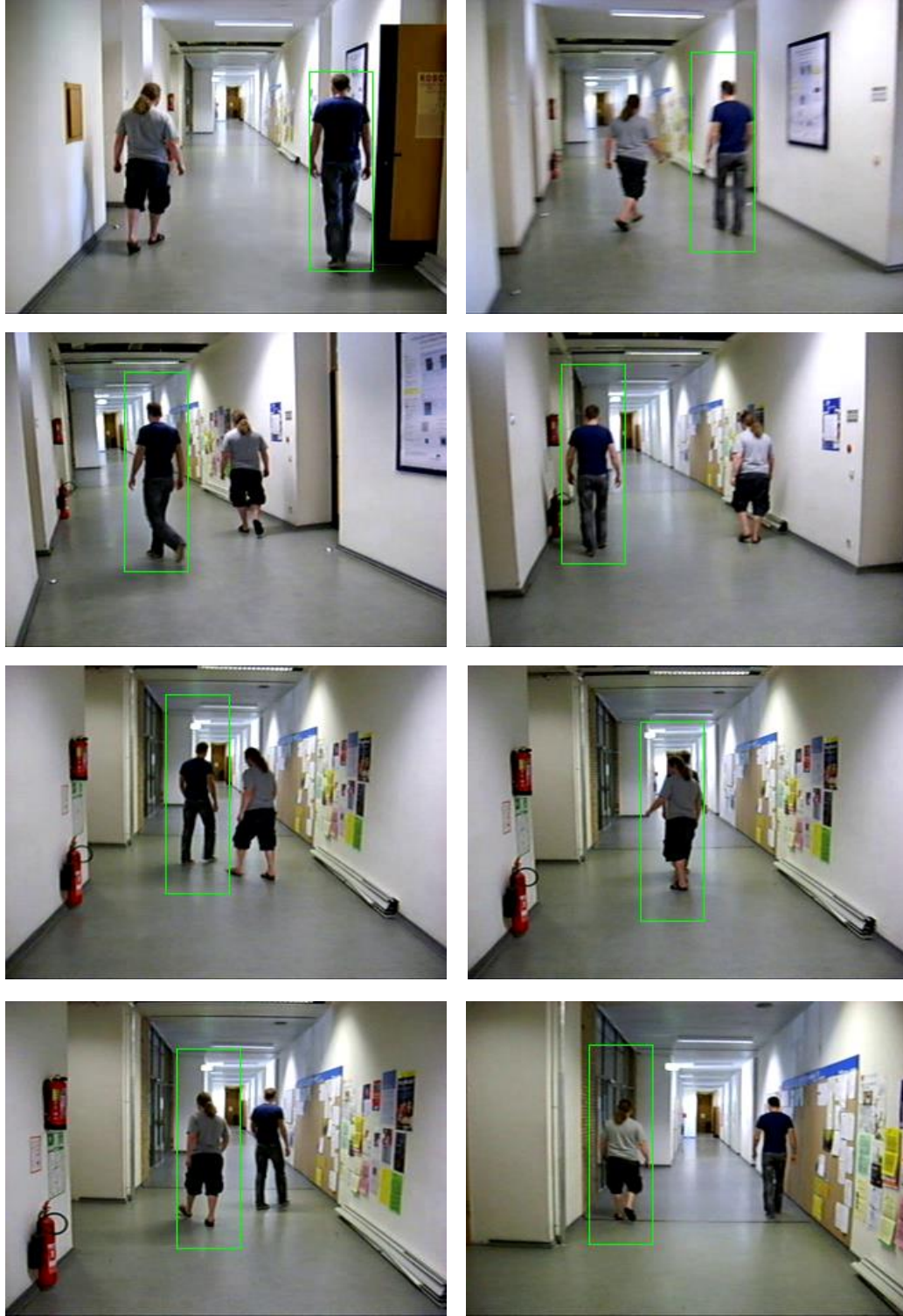
Şekil 4.31. Juice box sequence videosu için önerilen yöntem kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 60, 120, 190, 246, 285, 314, 367 numaralı çerçevelere ait takip sonuçları



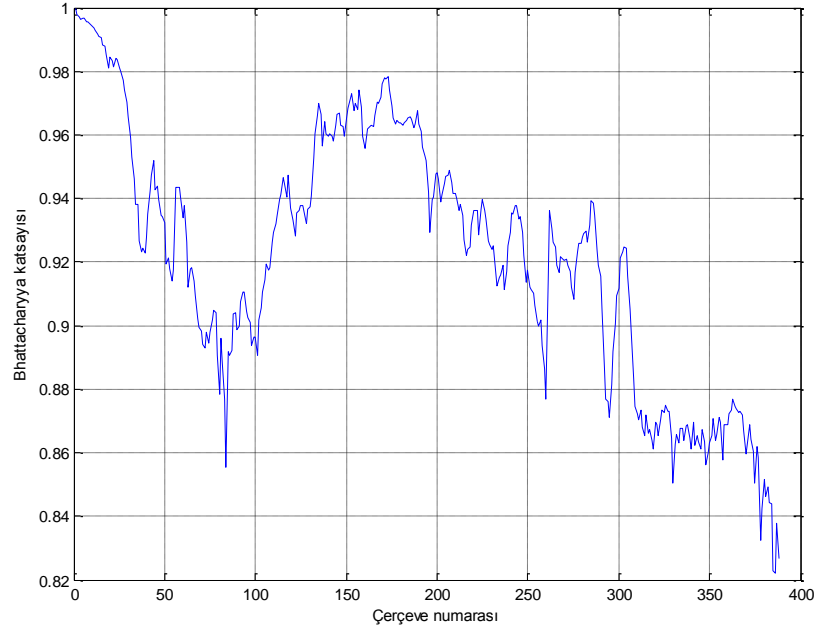
Şekil 4.32. Juice box sequence videosu için önerilen yöntem kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri



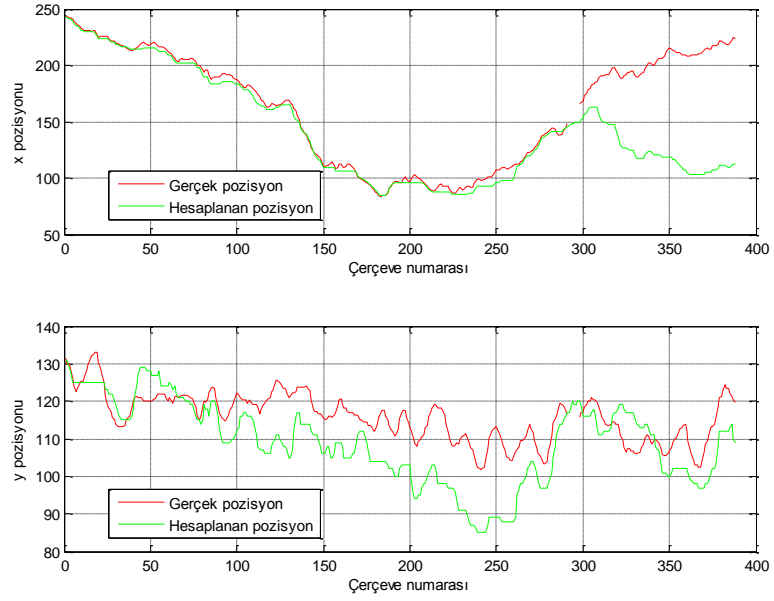
Şekil 4.33. Juice box sequence videosu için önerilen yöntem kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



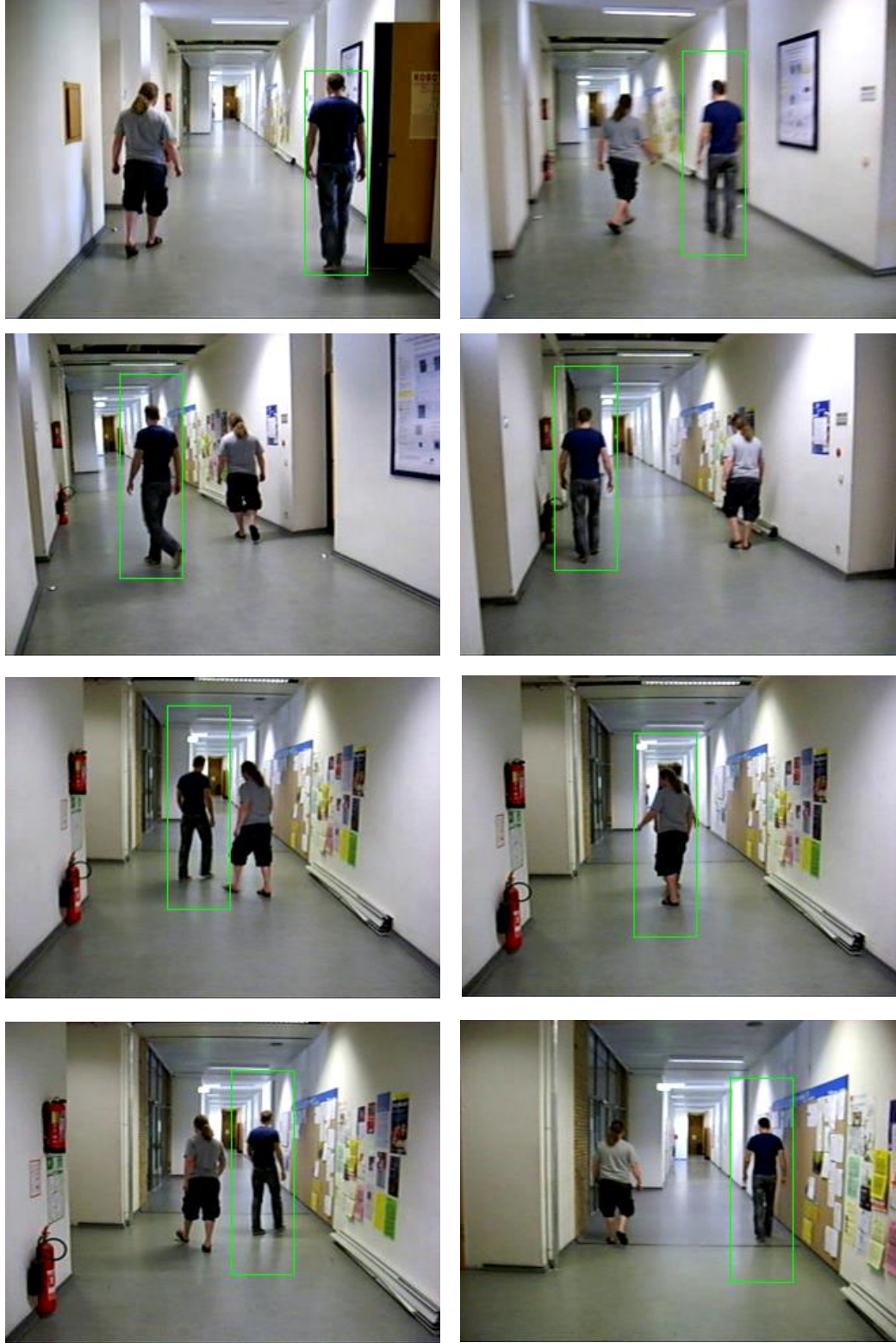
Şekil 4.34. Person floor sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 84, 150, 210, 280, 295, 315, 388 numaralı çerçevelere ait takip sonuçları



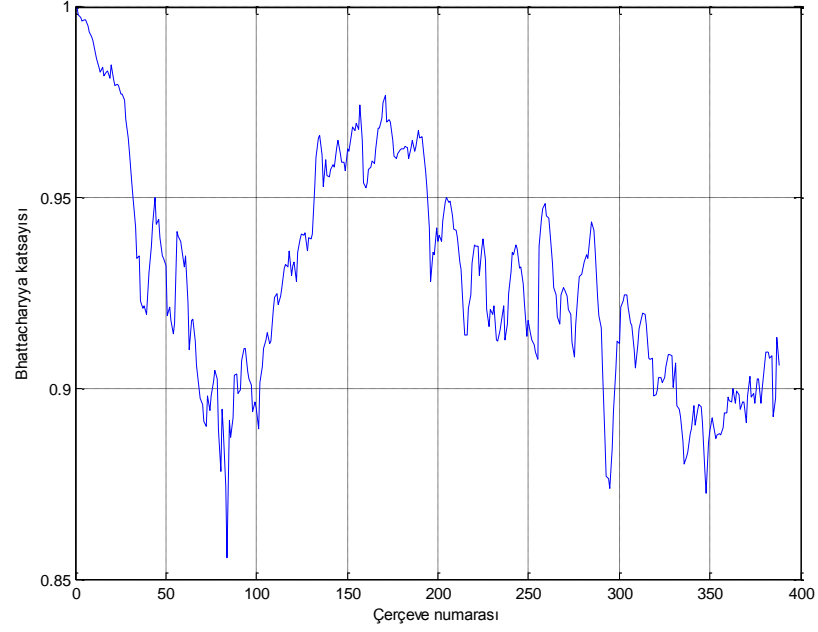
Şekil 4.35. Person floor sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri



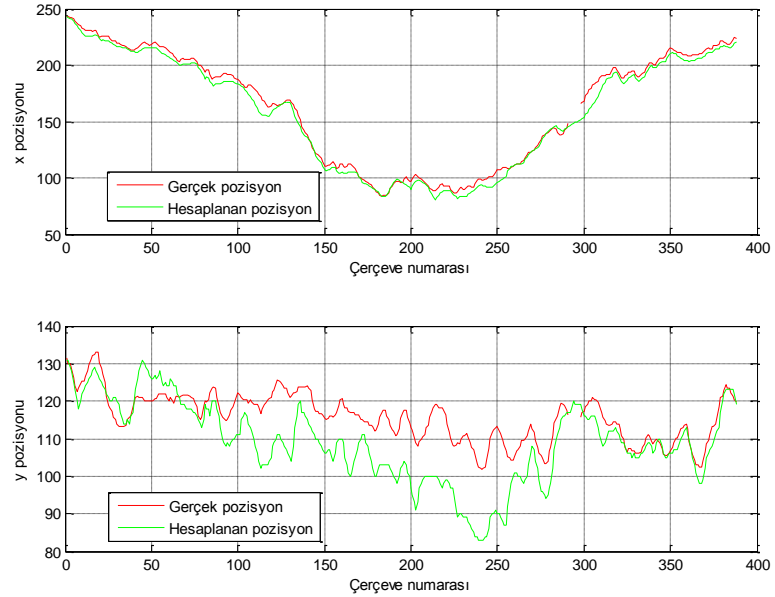
Şekil 4.36. Person floor sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



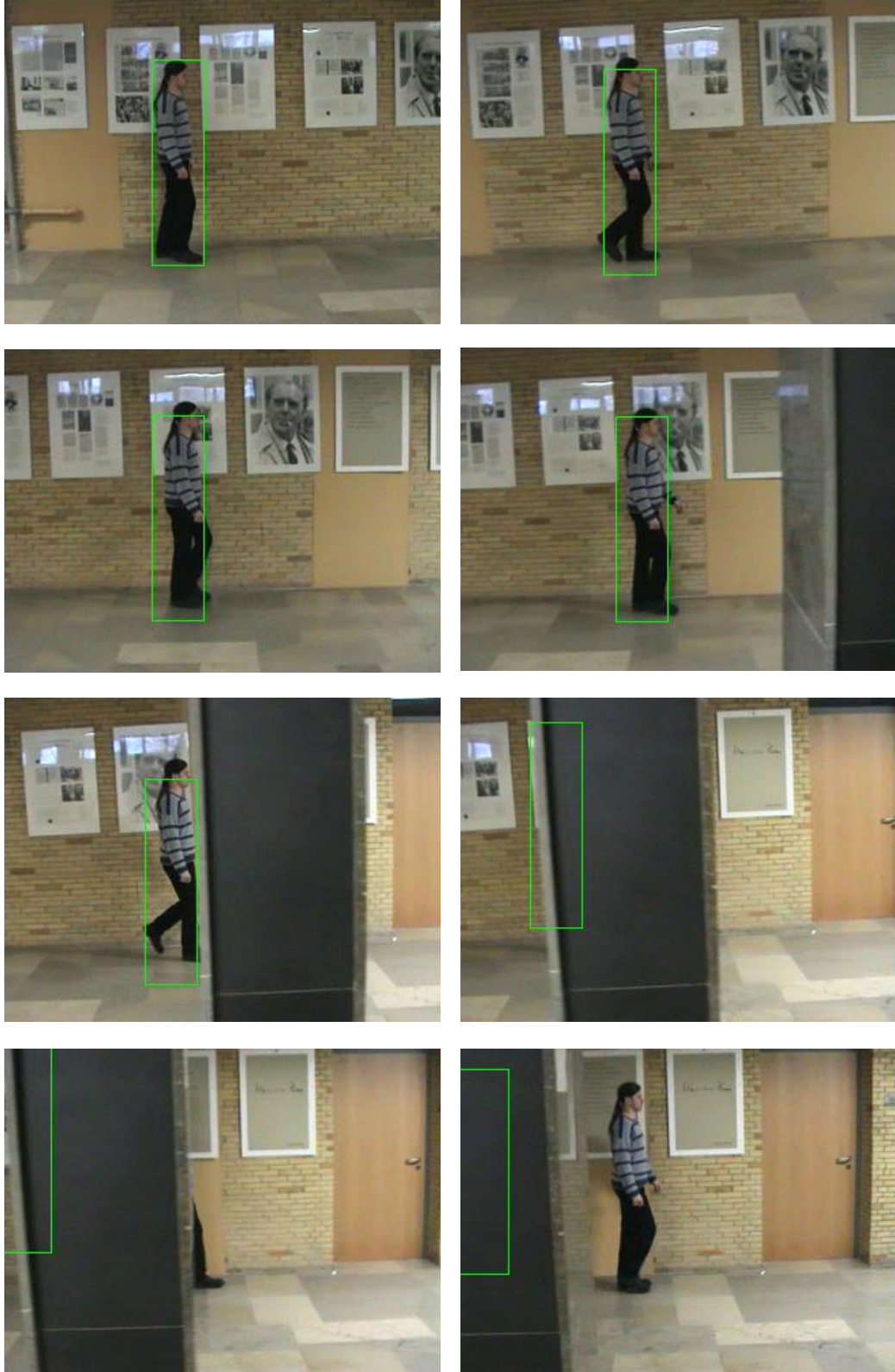
Şekil 4.37. Person floor sequence videosu için önerilen yöntem kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 1, 84, 150, 210, 280, 295, 315, 388 numaralı çerçevelere ait takip sonuçları



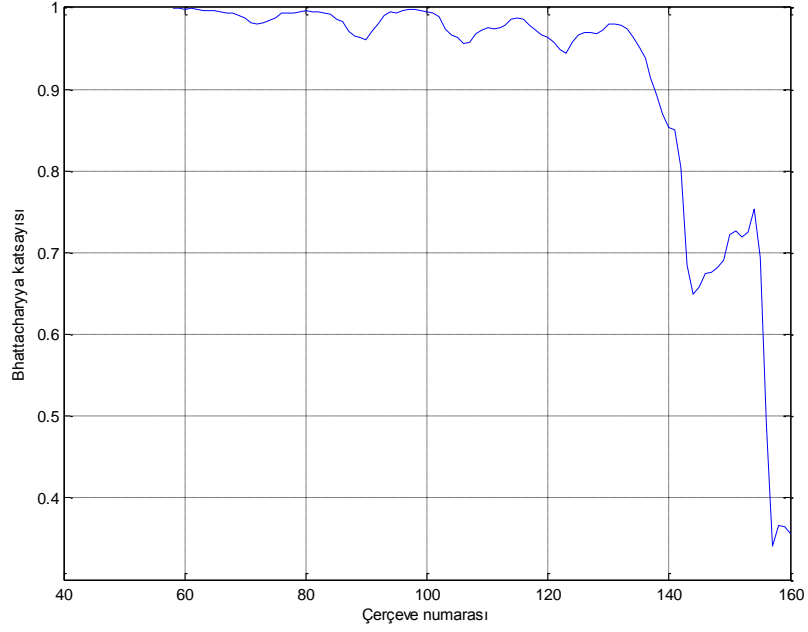
Şekil 4.38. Person floor sequence videosu için önerilen yöntemi kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri



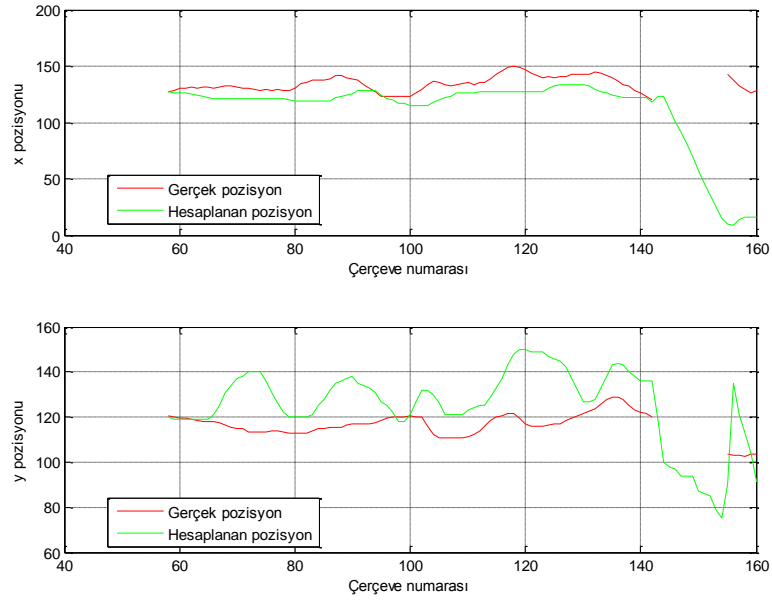
Şekil 4.39. Person floor sequence videosu için önerilen yöntemin kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri



Şekil 4.40. Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanarak soldan sağa ve yukarıdan aşağıya sırasıyla 58, 95, 113, 130, 141, 149, 154, 160 numaralı çerçevelere ait takip sonuçları

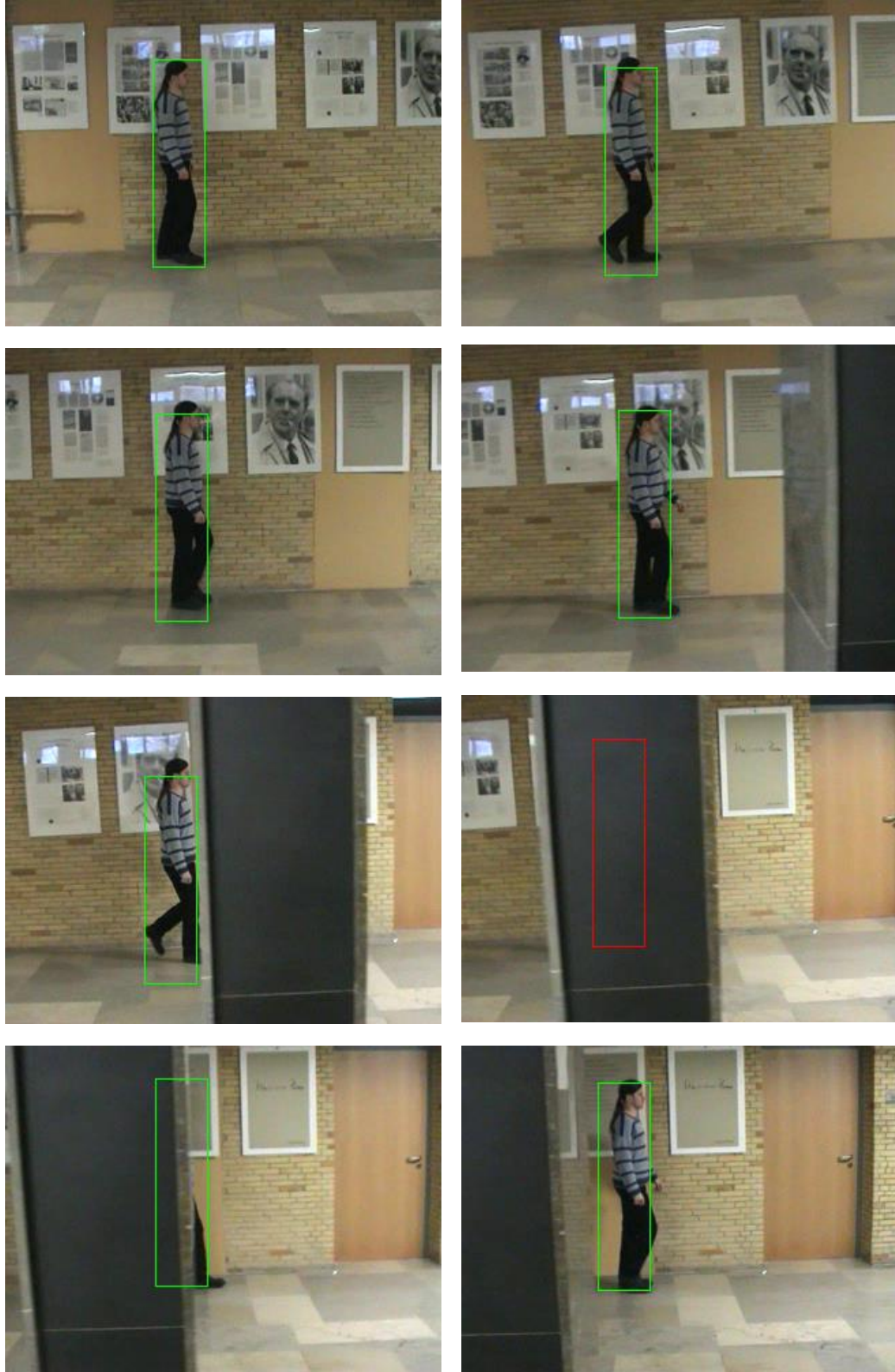


Şekil 4.41. Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanarak takip işlemi boyunca Bhattacharyya katsayısı değerleri

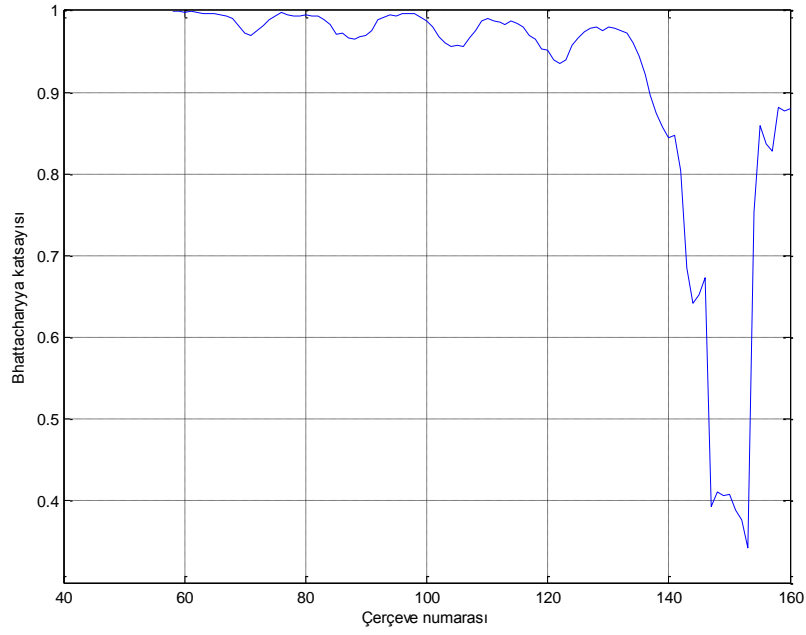


Şekil 4.42. Person fully occluded sequence videosu için standart ortalama kayma algoritması kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri

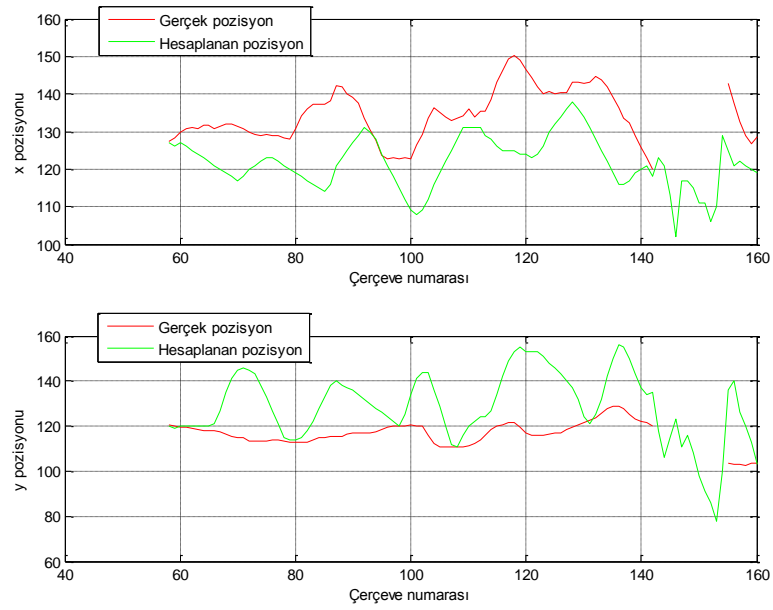




Şekil 4.43. Person fully occluded sequence videosu için önerilen yöntem kullanılarak soldan sağa ve yukarıdan aşağıya sırasıyla 58, 95, 113, 130, 141, 149, 154, 160 numaralı çerçevelere ait takip sonuçları



Şekil 4.44. Person fully occluded sequence videosu için önerilen yöntem kullanılarak takip işlemi boyunca Bhattacharyya katsayısı değerleri



Şekil 4.45. Person fully occluded sequence videosu için önerilen yöntem kullanıldığı durumda elde edilen hesaplanan pozisyon ve gerçek pozisyon bilgileri

## 5. SONUÇ

Bu çalışmada ortalama kayma algoritması ve Kalman filtresi birlikte kullanılarak nesne takip işlemi gerçekleştirilmiştir. Takip edilmesi hedeflenen nesne ardışık iki görüntü arasında ani yer değişikliği yaparsa ya da engele maruz kalırsa standart ortalama kayma algoritması iyi sonuçlar üretemeyip hedef nesneyi takip edememektedir. Önerilen yöntemde ise Kalman filtresi kullanılarak nesnenin hareket modeli de takip algoritmasıyla bütünleştirilmiştir. Hareket modelinin takip algoritmasına dahil edilmesi, hedef nesnenin ani hareket etmesi veya engele maruz kalması koşullarında başarılı sonuçlar üretilmesine yol açmıştır.

Önerilen yöntemde eşik değeri, ortalama kayma algoritmasından elde edilen Bhattacharyya katsayısıyla kıyaslanarak nesnenin engele maruz kalıp kalmadığına karar vermek için kullanılır. Eşik değerinin doğru ayarlanması takip sisteminin başarısını doğrudan etkilemektedir. Bundan ötürü, eşik değerinin otomatik olarak ayarlanacağı çalışmaların ileride yapılması planlanmaktadır.

## KAYNAKLAR

- [1] Y. Alper, J. Omar and S. Mubarak, *Object tracking: A survey*, **Acm computing surveys**, 38:4 (2006) 1-13.
- [2] G. R. Bradski, *Computer vision face tracking for use in a perceptual user interface*, **Intel Technology Journal**, (1998) 1-15.
- [3] A. J. Lipton, H. Fujiyoshi and R. S. Patil, Moving target classification and tracking from real-time video, Fourth IEEE Workshop on Applications of Computer Vision, Princeton, NJ, USA, Oct. 19-21, (1998), pp. 8-14.
- [4] I. Haritaoglu, D. Harwood and L. S. Davis, *W 4: Real-time surveillance of people and their activities*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 22:8 (2000), 809-830.
- [5] C. Shan, Y. Wei, T. Tan, and F. Ojardias, Real time hand tracking by combining particle filtering and mean shift, Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, May. 17-19, (2004), pp. 669-674.
- [6] F. Xu, X. Liu and K. Fujimura, *Pedestrian detection and tracking with night vision*, **IEEE Transactions on Intelligent Transportation Systems**, 6:1 (2005) 63-71.
- [7] S. Baysal and P. Duygulu, *Sentioscope: A Soccer Player Tracking System using Model Field Particles*, **IEEE Transactions on Circuits and Systems for Video Technology**, 99 (2015) pp. 1-14.
- [8] A. Lehuger, P. Lechat and P. Perez, An adaptive mixture color model for robust visual tracking, IEEE International Conference on Image Processing, Atlanta, GA, USA, Oct. 8-11, (2006), pp. 573-576.
- [9] C. R. Wren, A. Azarbayejani, T. Darrell ve A. P. Pentland, *Pfinder: Real-time tracking of the human body*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 19:7 (1997) pp. 780-785.
- [10] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan and M. Shah, *Visual tracking: An experimental survey*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 36:7 (2014) 1442-1468.

- [11] W. Yi, L. Jongwoo and Y. Ming-Hsuan, Online object tracking: A benchmark, Proceedings of the IEEE conference on computer vision and pattern recognition, Portland, OR, USA, June 23-28, (2013), pp. 2411-2418.
- [12] K. Fukunaga and L. D. Hostetler, *The estimation of the gradient of a density function, with applications in pattern recognition*, **IEEE Transactions on Information Theory**, 21:1 (1975) 32-40.
- [13] B. W. Silverman, *Density estimation for statistics and data analysis*, CRC press, New York, USA, 1986, p. 1-170.
- [14] Y. Cheng, *Meanshift, mode seeking, and clustering*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 17:8 (1995) 790-799.
- [15] D. Comaniciu and P. Meer, Robust analysis of feature spaces: color image segmentation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Porto Riko, Jun. 17-19, (1997), pp. 750-755.
- [16] D. Comaniciu and P. Meer, Mean shift analysis and applications, The Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, Sep. 20-27, (1999), pp. 1197-1203.
- [17] D. Comaniciu and P. Meer, Mean shift: *A robust approach toward feature space analysis*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 24:5 (2002) 603-619.
- [18] D. Comaniciu, V. Ramesh and P. Meer, *Kernel-based object tracking*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 25:5 (2003) 564-577.
- [19] D. Comaniciu, V. Ramesh and P. Meer, Real-time tracking of non-rigid objects using mean shift, IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, Jun. 13-15, (2000), pp. 142-149.
- [20] Y. Ukrainitiz and B. Sarel (2016), [http://www.wisdom.weizmann.ac.il/~vision/courses/2004\\_2/files/mean\\_shift/mean\\_shift.ppt](http://www.wisdom.weizmann.ac.il/~vision/courses/2004_2/files/mean_shift/mean_shift.ppt). (Son Erişim Tarihi: 12 Ocak 2016).
- [21] A. Yilmaz, K. Shafique, N. Lobo, X. Li, T. Olson and M. Shah, Target-tracking in FLIR imagery using mean-shift and global motion compensation, IEEE Workshop on Computer Vision Beyond Visible Spectrum, Hilton Head, SC, USA, May. 1-2, (2001), pp. 1-3.
- [22] J. Jeyakar, R. V. Babu and K. R. Ramakrishnan, *Robust object tracking with background-weighted local kernels*, **Computer Vision and Image Understanding**, 112:3 (2008) 296-309.

- [23] C. Yang, R. Duraiswami and L. Davis, Efficient mean-shift tracking via a new similarity measure, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, USA, June 20-25, (2005), pp. 176-183.
- [24] R. T. Collins, Mean-shift blob tracking through scale space, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, California, USA, June 18-20, (2003), pp. 234-240.
- [25] I. Leichter, M. Lindenbaum and E. Rivlin, *Mean shift tracking with multiple reference color histograms*, **Computer Vision and Image Understanding**, 114:3 (2010) 400-408.
- [26] Faragher R, *Understanding the basis of the Kalman filter via a simple and intuitive derivation*, **IEEE Signal processing magazine**, 29:5 (2012) 128-132.
- [27] G., Welch and G. Bishop, *An introduction to the kalman filter*, *Department of Computer Science*, University of North Carolina Press, USA, 2006, p. 1-16.
- [28] N. S. Peng, J. Yang and Z. Liu, *Mean shift blob tracking with kernel histogram filtering and hypothesis testing*, **Pattern Recognition Letters**, 26:5 (2005) 605-614.
- [29] E. V. Cuevas, D. Zaldivar and R. Rojas, *Kalman filter for vision tracking*, **Technical Report B 05-12**, (2005), 1-15.
- [30] A. H. Mazinan and A. Amir-Latifi, *Applying mean shift, motion information and Kalman filtering approaches to object tracking*, **ISA transactions**, 51:3 (2012) 485-497.
- [31] H. Jahandide, K. Mohamedpour and H. A. Moghaddam, *A hybrid motion and appearance prediction model for robust visual object tracking*, **Pattern Recognition Letters**, 33:16 (2012) 2192-2197.
- [32] S. K. Weng, C. M Kuo and S. K Tu, *Video object tracking using adaptive Kalman filter*, **Journal of Visual Communication and Image Representation**, 17:6 (2006) 1190-1208.
- [33] V. Karavasilis, C. Nikou and A. Likas, *Visual tracking by adaptive Kalman filtering and mean shift*. In *Artificial Intelligence: Theories, Models and Applications*, Springer, Berlin Heidelberg, Germany, 2010, p. 153-162.
- [34] H. T. Nguyen and A. W. Smeulders, *Fast occluded object tracking by a robust appearance filter*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 26:8 (2004) 1099-1104.

- [35] Z. Zhu, Q. Ji, K. Fujimura and Lee K, Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination, International Conference on Pattern Recognition, Quebec City, QC, Canada, Aug. 11-15, (2002), pp. 318-321.
- [36] D. Comaniciu and V. Ramesh, Mean shift and optimal prediction for efficient object tracking, International Conference on Image Processing, Vancouver, BC, USA, Sep. 10-13, (2000), pp. 70-73.
- [37] Bonn Benchmark on Tracking (2016), <http://www.iai.uni-bonn.de/~kleind/tracking/>, (Son Erişim Tarihi: 12 Nisan 2016).

## ÖZGEÇMİŞ

Ad Soyad: Mehmet Murat TURHAN

Doğum Yeri ve Tarihi: MUŞ-1985

Adres: İnönü Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği  
Bölümü, Merkez/MALATYA

E-Posta : [turhan.mmurat@gmail.com](mailto:turhan.mmurat@gmail.com)

Lisans: Fırat Üniversitesi Bilgisayar Mühendisliği Bölümü (2004-2009)

Tezden Türetilen Yayınlar/Sunumlar

- M. M. Turhan ve D. Hanbay, Mean shift based object tracking supported with adaptive Kalman filter, In Signal Processing and Communications Applications Conference (SIU), Malatya, Türkiye, 2015.