



Yapısal Özellikleri Kullanan Parçacık Filtresi İle Uzun Süreli Nesne Takibi

Haluk DİLMEN^{1*}, Muhammet Fatih TALU¹

¹İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Malatya/TÜRKİYE

Öz

Nesnelerin uzun süreli takip edilmesi eski bir araştırma konusu olmasına rağmen araştırmacıların hala aktif olarak ilgisini çeken ve hakkında birçok çalışma yapılan araştırma konularının başında gelmektedir. Bu çalışmada tahminsel yöntemler arasında adı anılan, durum uzay değişkenlerinden yararlanarak takip konusunu ilgilendiren dinamikleri modelleyen parçacık filtresi ile nesne takibi gerçekleştirilmiştir. Parçacık filtresinde, parçacık ağırlıklarının belirlenmesinde kullanılan ölçüm modelinde yenilikler sunularak nesnenin yapısal özelliklerinin kullanıldığı SSIM benzerlik katsayısı ile birlikte adaptif histogram eşitlemesi ve nesne merkez bölgesinin ağırlıklandırılması temeline dayanan yeni bir ölçüm modeli geliştirilmiştir. Yapılan deneysel sonuçlar, önerilen nesne takip yönteminin klasik takip performansını en az %18.59 oranında arttırdığı gözlemlenmiştir.

Makale Bilgisi

Başvuru: 19/04/2016

Kabul: 18/11/2016

Anahtar Kelimeler

Parçacık Filtresi
Uzun Süreli Nesne Takibi
İmge Benzerlik Metrikler
Modeli Bilinmeyen
Nesnelerin Takibi

Long Time Object Tracking Using Structural Features With Particle Filter

Abstract

Although long time tracking is an old research subject, it is still among the research subject actively attracting the attention of researchers and it is one of the research topic many studies conducted about. Object tracking with particle filter, known to be among stochastic methods, models dynamics related to tracking subjects by taking advantage of state space variables, implemented in this study. Presenting improvements in the measurement models used to determine the weight of the particles, a new measurement model based on structural features of similarity coefficients used by SSIM with adaptive histogram equalization and weighting center of the object has been developed in The Particle Filter. Experimental results show that the proposed measurement model in object tracking increase classical tracking performance by at least %18.59.

Keywords

Particle Filter
Long Time Object
Tracking
Image Similarity Metrics
Model Free Object
Tracking

1. GİRİŞ (INTRODUCTION)

Nesne takibi uygulamaları güvenlik, insan bilgisayar etkileşimi, video indeksleme, performans analizleri, akıllı ara yüzler, video sıkıştırma gibi geniş bir alanda güncel ve etkili çözümler üretmek adına araştırma konularına yön verici nitelikte çalışmalarda bulunmaktadır. Hakkında pek çok çalışma yapılmış, problemlere yeni yaklaşımlar ve çözümler getirilmiş olmasına rağmen nesne takibi problemi takip edilecek nesnenin ve dış etkenlerin durumuna göre oldukça karmaşık, zaman ve işlem gücü gerektiren bir duruma bürünebilir. Bahsedilen zorluklardan dolayı uygulama geliştiriciler zorlukları kısmen dahi olsa azaltmak adına nesne takibini iyi tanımlanmış ve iyi kontrol edilen ortamlar içinde gerçekleştirme yoluna başvurmuşlardır [1]. Bu durum takip işleminde kullanılan modelin yüksek oranda doğrusal olduğu varsayımını kabul etmekle birlikte uygun şartlar altında başarılı olabilecek yöntemleri sonuç vermiştir. Bununla beraber takip işleminde kullanılan modellerin tümünün doğrusal olduğunu varsaymak gerçek hayat için çoğu zaman mümkün değildir. Önerilen takip algoritmaları sınırlı durum ve koşul altında başarı sağlarken doğrusal olmayan sistemler ile karşılaşıldığında takip işleminde hatalar meydana gelmektedir. Bu çalışmada doğrusal olmayan sistem yapısına uygun nesne takibi konusunda başarı ile uygulanmış olan

*İletişim yazarı, e-mail: hdilmen2@gmail.com

PF incelenerek takip performansını arttırmak adına yeni öneriler ve yaklaşımlar sunulmuştur. Deneysel sonuçlarda önerilen yöntemlerin takip performansında en az %18.59 oranında iyileştirme içeren sonuçlar verdiği tespit edilmiştir.

1.1. Parçacık Filtresi İle Nesne Takibi (Object Tracking With Particle Filter)

Literatürde PF ile nesne takibi konusunda önerilmiş algoritmalarla sıklıkla karşılaşmak mümkündür [2]. [3] numaralı çalışmada, PF kullanılarak nesne takibi gerçekleştirilen yöntemler incelenmiş ve okuyucuların bilgisine sunulmuştur. Genellikle nesne takibi probleminde karşılaşılan durumlar, doğrusal olmayan sistem modeline sahip, Gauss dağılımı ile ifade edilemeyen, tek veya çok modlu yapılar olarak karşımıza çıkmaktadır[4]. PF ile nesne takibi sırasında uygulamaya özel yapının ve özniteliklerin kullanılması takip performansı açısından önemli yere sahiptir. Uygulamaya özel yapıyı belirleyen kriterlerin başında ise takip sırasında kullanılacak olan nesneye ait özniteliklerin belirlenmesi ve kullanımı sayılabilir. PF ile nesnelerin takibinde yaygın olarak kullanılan özelliklerden biri nesnenin renk bilgisidir. Renk bilgisi nesnenin poz değişiklikleri, yarı kapanma ve öteleme gibi durumlarda ayırt edici bilgiler sunduğundan takip sırasında tercih edilebilecek önemli bir özniteliktir. Bu amaçla [5] ile yapılan çalışmada nesne 8x8x8 lik RGB histogram demeti olarak ifade edilerek nesneye ait sonrasal dağılım 200 adet parçacık kullanılarak takip işlemi gerçekleştirilmiştir. Bununla birlikte [6]'de RGB renk histogramları yerine HSV renk uzayına ait 10x10 HS ve 10 demet V kanalına ait histogram demetlerinin kullanımı yönünde çalışma yürütülmüştür. Sabit olarak belirlenen histogram demetleri yerine bu işlemi otomatik olarak yapacak model seçimi önerilerini [7]'de görmek mümkündür. Her ne kadar renk bilgisi uygulamalarda en çok tercih edilen ayırt edici özellik olarak karşımıza çıksa da nesne ve arka plan renklerinin aynı olması, karmaşık arka plan imgelerinde kaybolmalar ve nesnenin görünüşündeki bozulmalar söz konusu olduğunda yanlış tahminlere sebebiyet veren durumlar oluşmaktadır. Problemin çözümü adına araştırmacılar nesneyi tek bir öznitelik ile takip etmek yerine birden fazla özneliğin takip edildiği yapılara yönelmişlerdir [8]. Bu takip edilen özneliklere örnek vermek gerekirse renk bilgisine ilave olarak hareket, ses, nesnenin kenar bilgisi, şekil, çevrit gibi nesneye ait ayırt edici öznitelikler sıralanabilir.

PF ile nesne takibinde uygulama geliştiricilerin önem vermeleri gereken bir diğer önemli husus ise parçacıkların ağırlıklandırılmasında kullanılan ölçüm modeli yardımıyla benzerlik tahmininin yapılmasıdır. (8)'de görüleceği üzere benzerlik tahmini $p(z_k|x_k^i)$ parçacıkların ağırlıklandırılmasında kullanılan bir terim olarak karşımıza çıkmaktadır. Takip sırasında parçacık ağırlıklarının etkilediği iki önemli konu vardır. Dolayısı ile benzerlik tahmini bu iki konuyla doğrudan ilişkilidir.

Birincisi parçacık ağırlıkları parçacıkların yeniden örneklenmesinde kullanılmaktadır. Yeniden örnekleme sayesinde iyi olan parçacıkların sayısının artırılması ve kötü olanların yok edilmesi veya sayısının azaltılması hedeflenmektedir. İyi olan parçacıkların yeniden örneklenmesi nesneye ait takip işleminin zamanla performansının iyileştirilmesi sonucunu doğuracaktır.

İkincisi tüm parçacıklara ait ağırlık değerleri olasılık yoğunluk fonksiyonu türünden ifade edildiğinden dolayı parçacıkların ağırlıklarının ortalama değeri durum tahminini yani nesnenin yörüngesine ait tahmin değerini ifade etmiş olacaktır [9].

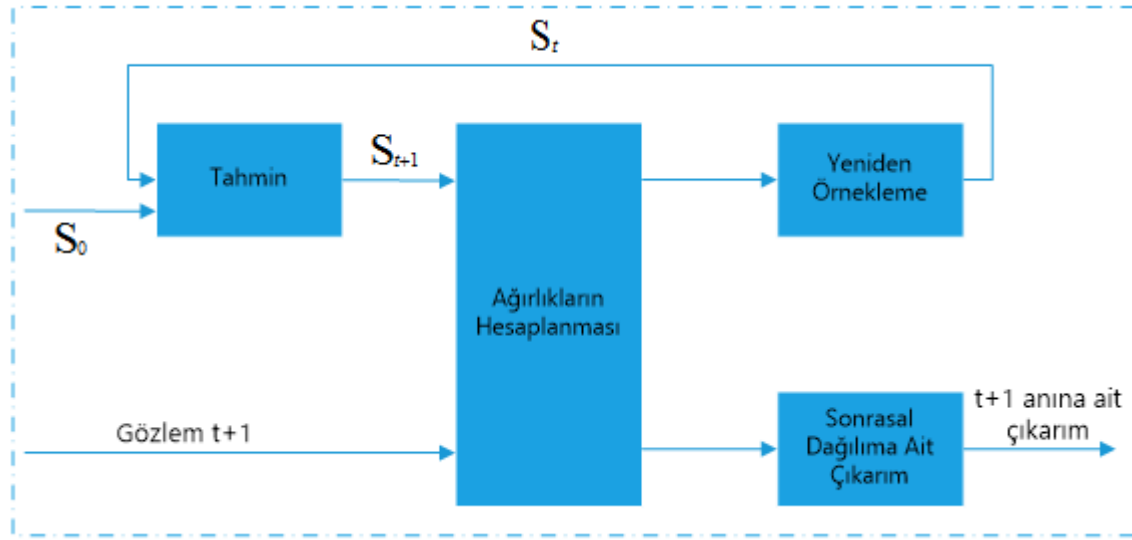
Benzerlik tahmini en fazla hesap maliyeti gerektiren bölüm olduğundan gerçek zamanlı sistemlerin tasarımı aşamasında dikkat edilmesi gereken konular arasında yer almaktadır. Bu çalışmada PF, Optik Akış yöntemi ile birlikte kullanılmıştır. Nesnenin hareket ve konum belirleme probleminde optik akış kabul gören yöntemler arasındadır [10]. Modern optik akış temelli hesaplama yöntemleri gerçek zamanlı ve yüksek doğrulukta tahminler sunduğundan dolayı görüntü işleme konusunda iyi sonuçlar vermektedir. Optik akış yönteminin PF ile kullanımı adına [11]'de PF ile kullanılabilecek optik akış temelli üç ayrı gözlem modeli sunulmuştur. PF ile el takibi yapan [12]'te optik akış yöntemi kullanarak hareket modelinin yapısı izah edilmiştir. PF'nin temel çalışma prensiplerini anlama adına ön hazırlık olarak yapay olarak üretilen iki boyutlu sinyaller üzerinde parçacık filtresi ile takip işlemi gerçekleştirilmiştir [13]. PF ile nesne takibi iki boyutlu sinyaller üzerinde PF'nin uygulanması işleminden çok daha karmaşık ve hesaplama maliyeti gerektiren koşulları barındırmaktadır. İki boyutlu sinyallerin PF ile takibi sırasında (1) ile gösterilen x_k durum vektörümüz nesnenin konum bilgisinden (x,y) oluşmakta iken, nesne takibi sırasında x_k durum vektörümüz tasarıma bağlı olarak nesnenin konum bilgisine ilave olarak hız, ivme, renk, ölçek, rotasyon vb. gibi değişkenleri ihtiva edebilmektedir. Durum vektörünün boyutundaki artış doğal olarak hesaplama

maliyetinde artışları sonuç verecektir. Karmaşıklığından kasıt ise, durum vektörünün tasarıma bağlı olarak içeriğinin belirlenmesi ile ilgili çalışmaları ifade etmek için kullanılmıştır.

Makalenin geri kalan kısmı şu bölümlerden oluşmaktadır: 2. Bölümde PF'nin genel yapısı, 3. Bölümde bu çalışmayı motive eden nedenler, 4. Bölümde önerilen sistemin var olan yapılardan farkı, 5. Bölümde önerilen sistem ile ilgili karşılaştırma sonuçları ve performans değerlendirmesi, 6. Bölüm sonuç ile ilgili değerlendirmeler, 7. Bölüm makalede kullanılan kısaltmalar ve 8. Bölümde çalışmada yararlanılan kaynaklar ile ilgili bilgiler sunulmuştur.

2. PARÇACIK SÜZGEÇİ GENEL YAPISI (BASIC STRUCTURE OF PARTICLE FILTER)

Şekil 1'de genel yapısı gösterilen PF doğrusal olmayan sistemlerin tahmininde başarı ile kullanılmaktadır.



Şekil 1. PF'nin genel yapısı

En genel ifadesiyle filtreleme işlemi, geçmiş durum ve en son geçerli ölçüm değeri kullanılarak sistemin gizli olan en son durum tahmin değerinin elde edilmesi işlemi olarak tanımlanabilmektedir [14]. Durum tahmini $p(x_t|x_{t-1}, z_{0:t})$ şeklinde olasılık yoğunluk fonksiyonu olarak ifade edilip tasarıma bağlı olarak, t anındaki x_t durum ve z_t gözlem verileri farklı öğeler olarak seçilebilirler. z_t gözlem verisi gürültülü ölçümleri içeren sensör ve donanımlardan okunan verilere karşılık gelmektedir. PF ile hedeflenen bir sistemin analizi için en az iki modele gereksinim duyulmaktadır. (1) ve (2) nolu denklemler sırasıyla bahsedilen zamana bağlı durum geçiş ve ölçüm modeline ait eşitlikler olarak kabul edilirler.

$$x_k = f_k(x_{k-1}, v_{k-1}) \quad (1)$$

x_k , v_{k-1} sırasıyla sistemin k anındaki durum değerini belirten bir vektör, sistem gürültüsünü gösteren vektör olarak bilinmektedirler. x_k durum vektörünü oluşturan değişkenlerinin gizli ve gözlemlenemeyen değişkenlere sahip olduğu ve durum ile ilgili bilgilerin yalnızca gürültülü ölçümler içeren ölçüm modeli (2) ile elde edilebildiği varsayılmaktadır. f_k ise zamana bağlı, doğrusal olmayan durum geçiş fonksiyonunu belirtmektedir.

$$z_k = h_k(x_k, n_k) \quad (2)$$

z_k ölçüm modeli, h_k ölçüm fonksiyonunu ve n_k ise ölçüm gürültüsünü gösteren ölçüm fonksiyonuna ait vektörler olarak adlandırılmaktadırlar. Bir sonraki durum değerine ait tahmin işlemi önceki durum ve yeni ölçüm değeri ile gerçekleştirilmeye çalışıldığından problem Bayes Teoremine uygun niteliğe dönüşmüş olacaktır. Hedeflenen sonrasal dağılım " $p(x_k|z_{1:k})$ " Bayes yaklaşımıyla rekürsif olarak iki adımdan oluşan hesaplama adımıyla gerçekleştirilebilmektedir.

Birinci adım tahmin adımı olarak adlandırılıp $k-1$ anında hedeflenen $p(x_k|z_{1:k-1})$ dağılımını, (3) ile ifade edilen eşitlikte $p(x_{k-1}|z_{1:k-1})$ değeri kullanılarak elde etmeyi amaç edinmektedir.

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})d_{x_{k-1}} \quad (3)$$

$p(x_{k-1}|z_{1:k-1})$ değeri Bayes Kuralı'nın rekürsif yapısında dolayı, $p(x_k|x_{k-1})$ değeri ise (1) ile belirtilen eşitlik yardımıyla hesaplanan değerlerdir.

Güncelleme adımı ise bir önceki adım olan tahmin adımından elde edilen değer ve yeni ölçüm (z_k) değerleri kullanılarak (4) yardımıyla x_k üzerinde sonrasal dağılımın hesaplanması olarak ifade edilmektedir.

$$p(x_k|z_{1:k}) \propto p(z_k|x_k)p(x_k|z_{1:k-1})d_{x_{k-1}} \quad (4)$$

Filtreleme işleminin doğrusal olmayan sistemler için uygulanmasında tahmin ve güncelleme adımlarını analitik olarak yerine getirilmesi her zaman mümkün olmayıp problemin çözümü adına tahminsel yaklaşım yöntemini benimseyen Monte Carlo Yöntemleri tercih edilmektedir. SIS, Ardışık Monte Carlo Örnekleyicileri arasında bu konuda en basit yapıda olan metot olarak anılmaktadır. SIS'e ait temel mantık $k - 1$ anındaki sonrasal dağılımı $p(x_{0:k-1}|z_{1:k-1})$ ağırlıklandırılmış örnekleme seti $\{x_{0:k-1}^i, w_{k-1}^i\}_{i=1}^N$ veya parçacık ile rekürsif olarak güncellenerek bir sonraki k anındaki sonrasal dağılım $p(x_{0:k}|z_{1:k})$ elde edilmeye çalışılmaktadır. Parçacıklara ait ağırlıklar $\sum_i w_k^i = 1$ şeklinde normalize edilmektedir. En son durumu ifade eden k anındaki sonrasal dağılıma ait gösterim (5)'te sunulmuştur.

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_{0:k} - x_{0:k}^i) \quad (5)$$

Hedef dağılımdan örnekleme yapmanın zor olduğu durumlarda taslak dağılımdan örnek seçilmesi yoluna gidilmiştir. SIR olarak adlandırılan PF'nin bu türevinde hedef dağılım p_x yerine, taslak dağılım q_x 'ten örnekler seçilerek durum tahmin yapılmaktadır. Örneklerin $q(x_{0:k}|z_{1:k})$ ile ifade edilen önem yoğunluk fonksiyonundan seçildikleri düşünüldüğünde parçacık ağırlıkları (6) ile ifade edilen eşitlik yardımıyla hesaplanabilir.

$$w_k^i \propto \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})} \quad (6)$$

Ağırlıkların güncellendiği denklem ile işlem yapabilmek için sisteme ait sonrasal dağılımı $p(x_{0:k}|z_{1:k})$; $p(x_{0:k-1}|z_{1:k-1})$, $p(z_k|x_k)$, $p(x_k|x_{k-1})$ ile belirtilen olasılıksal formdan oluşan bileşenlerle ifade etmek gerekmektedir. Elde edilen yeni ölçüm değerleri kullanılarak Bayes Kuralı yardımıyla ağırlıklar (7) ile belirtilen eşitlik şeklinde ifade edilebilmektedir.

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \quad (7)$$

Daha da ileri gidip önem yoğunluk fonksiyonunun $q(x_k|x_{0:k-1}, z_{1:k}) = q(x_k|x_{k-1}, z_k)$ yapısında olduğu kabulü yapılırsa ağırlık güncelleme işlemi tüm geçmiş yerine sadece bir önceki durum ve son ölçüm değerleri ile hesaplanabilecek hale dönüştürülmüş olacaktır. k anındaki parçacıklara ait ağırlıklar, $k-1$ anındaki ağırlıklar kullanılarak (8) ile belirtilen eşitlik yardımıyla hesaplanabilecektir.

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \quad (8)$$

Filtrelenmiş sonrasal dağılımımızı

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \quad (9)$$

formunda yazıp parçacık sayımızı sonsuza gidiyor kabul ettiğimizde sonrasal dağılımın en yakın değeri elde edilmiş olunacaktır. Bu çalışmada PF nesne takibi işleminde kullanıldığından dolayı parçacıkları hedef nesneyi ifade eden durum vektörünü içeren ağırlıklandırılmış set olarak ifade etmek mümkündür. Yöntemimizde nesneye ait durum bilgisi; nesnenin konumu (merkez noktası), nesnenin hareket hızı ve nesneyi çevreleyen dikdörtgenin yükseklik ve genişlik bilgilerinden oluşmaktadır. Dolayısı ile her bir parçacık takip edilmek istenen nesneyi temsil eden hipotez olarak düşünülebilir. Sonrasal dağılım ağırlıklandırılmış parçacıklar türünden örneklenmiş olmaktadır.

3.ÇALIŞMAYI MOTİVE EDEN NEDENLER (REASONS MOTIVATED THIS WORK)

Genel olarak nesnenin durum tahminini yapmayı amaç edinen yöntemleri deterministik ve tahmini yöntemler olarak iki bölüme ayırmak mümkündür. Deterministik metotlar nesne takibinin doğrusal yapıda olduğu tezini savunmuş olup doğrusal olmayan fonksiyonları doğrusal yapıya çevirme yoluna gitmişlerdir [15]. Bu amaçla yaygın olarak kullanılan yöntem Taylor açılımı olarak karşımıza çıkmaktadır. Deterministik nesne takibi konusu ve yöntemleri bu makale çalışmasının alanının dışında kaldığından dolayı bu konuda detaylara girilmeyecektir. Bu çalışmada ilgi alanını sınırlandırmak ve çözüme odaklanmak amacıyla çalışmalar PF'de kullanılan ölçüm modeli üzerinde yoğunlaştırılmış olup daha iyi sonuçlar üretebilecek ölçüm modeli önerilmiştir. Ölçüm modeli video imgeleri içinde hedef ve aday görüntü kümeleri arasındaki benzerlik oranını bulmak adına karşılaştırmalardan oluşmaktadır. PF ile nesne takibi gerçekleştiren uygulamaların pek çoğunda renk veya gri seviye histogram seviyelerinin karşılaştırmasına dayalı ölçüm modellerinin tercih edildiği görülmektedir. Bunun altında yatan ana sebep metotların basit ve hesaplama maliyetinin düşük olması olarak gösterilebilir. Histogram temelli karşılaştırmalar video imgelerindeki ışık, kontrast ve renk değişimlerinin fazla olduğu durumlarda bir takım yanlış hesaplamalara ve takip edicinin hatalı neticeler vermesine neden olmaktadır. Ölçümleri daha sağlıklı yapma adına renk ve histogram temelli karşılaştırmaları temel alan klasik yöntemler yerine gelişmiş yöntem ve yaklaşımlar ortaya konulmuştur[7][16][17][18]. [7],[16] ile belirtilen çalışmada nesne ve aday görüntü kümeleri seyrek gösterim olarak ifade edilen şablonlar haline dönüştürülerek, hedef ve adaylar arasındaki izdüşümler içinde en küçük hatayı veren aday nesne doğru tahmini göstermiş olarak kabul edilmektedir. [17] ile ifade edilen çalışmada takip işlemine başlamadan elde edilen sabit görünüm modeli yerine, çevrimiçi olarak görünüm modeli inşa eden yaklaşım benimsenmiştir. Bu sayede nesneye ait takip sırasında ortaya çıkan şekil, özel ışık durumları gibi nesnenin görünümü etkileyen faktörlere adaptasyon gerçekleştirilmiş olmaktadır. [18] ile gerçekleştirilen çalışmada ölçüm modeli inşa edilirken nesneye ait yapısal özelliklerinde değerlendirildiği çalışma gerçekleştirilmiştir. Önerdiğimiz yöntem [18] ile belirtilen yaklaşıma ilave özellikler katarak daha iyi takip sonucunun elde edilmesinde kullanılmıştır. Bu bölümün geri kalan kısmında önerdiğimiz yöntemin alt yapısına ait temel bilgiler genel olarak sunulmuştur.

Ardışıl video imgeleri veya durağan imgelerin karşılaştırılmasında klasik yöntemlere göre daha yüksek başarıma sahip SSIM yöntemi son yıllarda araştırmacıların ilgisini çekmiştir [19]. SSIM, imgeler arasındaki benzerliği ortaya çıkarmak için ortalama parlaklık ve renk değişimi bilgilerini bir fonksiyon ile değerlendirmektedir.

$$SSIM(x, y) = [I(x, y)]^\alpha \cdot [C(x, y)]^\beta \cdot [S(x, y)]^\gamma \quad (10)$$

$$I(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad (11)$$

$$C(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (12)$$

$$S(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (13)$$

$\mu_x, \mu_y, \sigma_x, \sigma_y$, ve σ_{xy} değişkenleri, X ve Y imge veya sinyalleri için sırasıyla yerel ortalamalar, standart sapmalar ve kovaryans değeri olarak hesaplanmaktadır. c_1, c_2, c_3 değerleri ise denklemlerde sıfıra bölünmekten kaçınma adına kullanılan küçük değerler olarak bilinirler. α, β, γ katsayıları ise her bir terimi ağırlıklandırmak amacıyla kullanılan katsayılar olup, eşit ağırlıklandırma amacıyla tamamı 1 olarak seçilebilir. $c_3 = c_2/2$ özel durumu göz önünde bulundurulduğunda benzerlik katsayısını 14 ile gösterilen ifadeye indirgemek mümkündür.

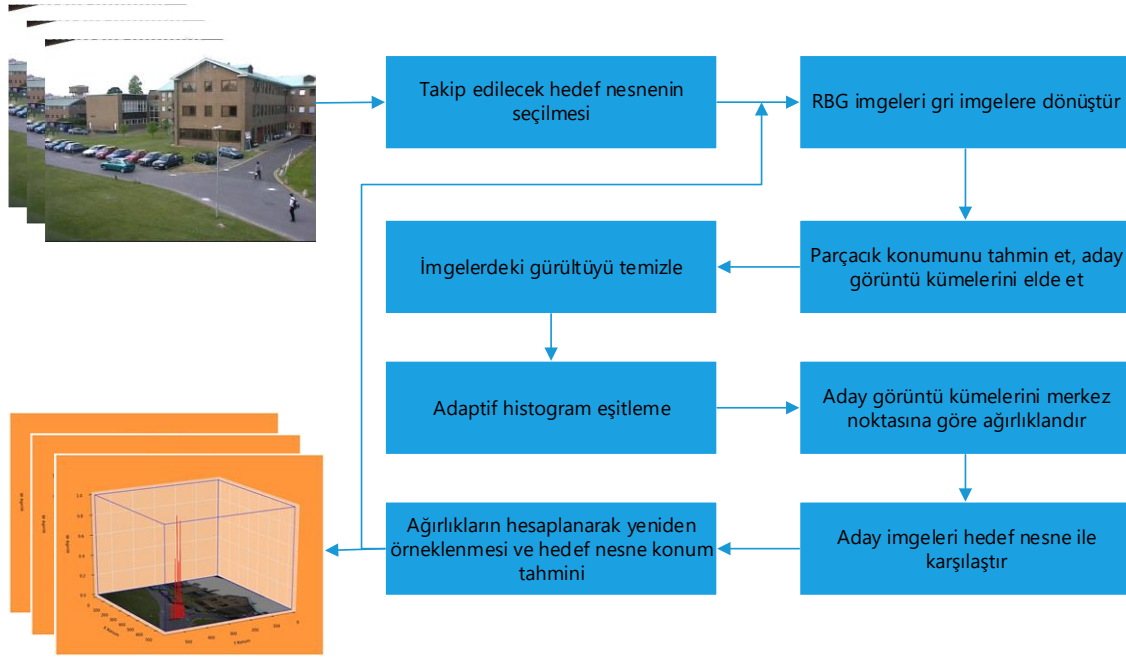
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (14)$$

SSIM benzerlik değerinin PF yöntemi ile birlikte nesne takip probleminin çözümünde kullanılması başarılı sonuçlar sağlamıştır [18]. (10) ile ifade edilen eşitlikte, imgeler arasındaki benzerlik katsayısı hesaplanırken üç ayrı terimden elde edilen sonuçlar çarpılmaktadır. Çarpım sonucunun yüksek çıkması, (x,y) merkezli görüntü parçalarının bir birine benzediğini, düşük çıkması ise benzemediği anlamını taşımaktadır.

Klasik nesne takibi yöntemlerinin aksine, bu çalışmada, görüntü benzerliğini yüksek bir başarımla ortaya çıkarabilen geliştirilmiş SSIM benzerlik katsayısının PF çatısı altında kullanılması önerilmektedir. Bir sonraki bölümde önerilen yöntemin detayları verilmektedir.

4. ÖNERİLEN YÖNTEM (PROPOSED METHOD)

Uzun süreli takip başarısı elde edebilmek hedefiyle, parçacık filtresinin ölçüm fonksiyonunda imge bloklarının SSIM yöntemiyle eşleştirilmesine dayalı yeni bir karşılaştırma yöntemi geliştirilmiştir. Önerilen yöntemin aşamalı akış diyagramı Şekil 2'de verilmektedir.



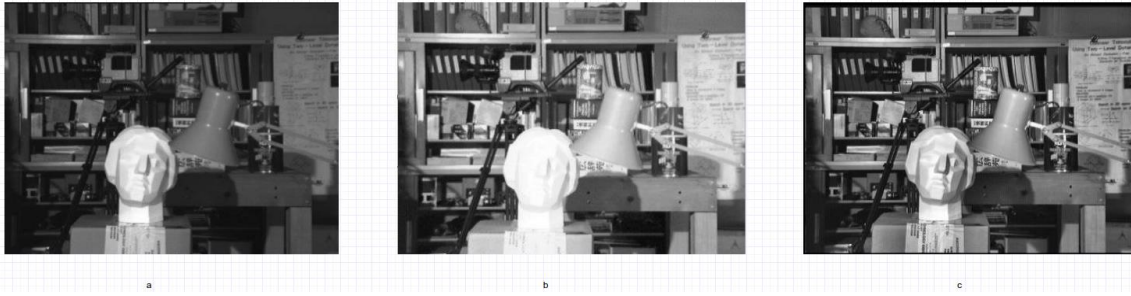
Şekil 2. Önerilen PF takip yöntemi ve ölçüm metodu

Nesne takibine ait durum vektörümüz $X_k = \{x_k, y_k, u_k, v_k\}$ ile ifade edilip, x_k, y_k aday nesne konum bilgisini; u_k, v_k ise x ve y yönündeki hızları temsil etmektedir. Nesne hareketi birinci derece otoregresif model göz önünde bulundurularak (15) ile gösterilen eşitlikteki gibi modellenmiştir.

$$x_k = Ax_{k-1} + \eta_{k-1} \quad (15)$$

A durum geçiş matrisini; η_{k-1} ise Gaussian olarak sistem gürültüsünü temsil etmektedir.

Önerilen yöntemin ilk aşamasında kullanıcı tarafından takip edilecek hedef nesnenin başlangıç konumu belirlenir ve nesne üzerine parçacıklar yayılır. Karşılaştırmaları gri imgeler üzerinde yaptığımızdan dolayı RGB imgeler gri imgelere dönüştürülür. Her bir parçacıktan aday imge parçaları elde edilir. Hedef ve aday görüntü parçaları arasında yüksek doğrulukta karşılaştırma yapabilmek amacıyla görüntülerdeki gürültü etkisinin ortadan kaldırılması için Gauss Filtresi kullanılmış ve giriş görüntüleri yumuşatılmıştır. Bir sonraki adımda gürültüden arındırılmış imgeler adaptif histogram eşitlemesi yöntemine tabi tutulmuştur. Adaptif histogram eşitlemesi yerel bölgelerdeki doku ve desen bilgilerini daha iyi koruduğundan dolayı karşılaştırma performansını doğrudan etkileyen faktör olarak karşımıza çıkmaktadır [20]. Örnek imgeler üzerinde yapılan Adaptif histogram eşitleme karşılaştırma sonuçlarına ait görseller Şekil 3'te paylaşılmıştır.



Şekil 3. (a) Orijinal imge, (b) Global histogram eşitleme, (c) Adaptif histogram eşitleme

Önerilen yönteme ait sonraki aşama dikdörtgen çerçeve içinde konumlandırılan hedef imgelerin merkeze uzaklıklarına göre ağırlıklandırılması işlemi kapsamaktadır. İmge takibi sırasında merkeze yakın noktada yer alan pikseller kenarlara doğru yer alan piksellerden daha önemlidir. Bu amaçla merkeze yakın piksellerin kenar bölgedeki piksellere göre daha fazla ağırlıklandırılması amacıyla imge, imge boyutunda geniş Gauss fonksiyonundan geçirilir ve imge blokları SSIM metodu kullanılarak benzerlik oranı $S(a, b)$ elde edilir. a ve b değişkenleri karşılaştırılan imge bloklarını temsil etmektedir. Karşılaştırma sonucunda hesaplanan benzerlik oranı kullanılarak, farklılık katsayısı (16) yardımıyla elde edilir.

$$D(a, b) = 1 - S(a, b) \quad (16)$$

Fonksiyona ait varyans değeri OpenCV metodlarındaki varsayılan değer olarak seçilip, Gauss çekirdeğinin büyüklüğü temel alınarak hesaplanmaktadır ve karşılaştırılan imge bloklarında sabit olarak tutulmaktadır. Farklılık katsayısına ait sonuçlar $[0,1]$ aralığında değişmektedir. Uzaklık metriği kullanılarak parçacıkları ağırlıklandırmada kullanılacak olabilirlik katsayısı $p(z|x)$, (17) ile belirtilen eşitlik vasıtasıyla elde edilir.

$$p(z|x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(D(a,b))^2}{2\sigma^2}} \quad (17)$$

$D(a, b)$ (16) ile hesaplanan farklılık katsayısını, σ ise varyans değerini ifade etmektedir. Her bir parçacık karşılaştırma sonucunda elde edilen değerlere göre ağırlıklandırıldıktan sonra tüm parçacıkların ağırlıkları normalize edilmektedir. Bu sayede nesne durum bilgisini belirleyecek olasılık yoğunluk dağılımı parçacıkların ağırlıkları cinsinden ifade edilmektedir. Parçacıkların ortalama değeri ise bize tahmin edilen nesne konumu hakkında bilgi vermektedir.

5. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Bu bölümde önerilen yönteme ait sonuçları içeren karşılaştırmalı sonuçlar paylaşarak önerilen yöntemin mevcut yöntemlere karşı olan avantaj ve dezavantajları ifade edilmektedir. Önerilen yönteme ait değerlendirme sonuçları nesne takip yöntemlerinin karşılaştırmalı sonuçlarının yer aldığı [21]'de belirtilen video görüntüleri üzerinde uygulanmıştır. Klasik nesne takip yöntemleri ile önerilen yöntemin sonuçlarını kıyaslamalı bir şekilde karşılaştırmak için Python programlama dili tercih edilmiş ve OpenCV 3.0 kod geliştirme kütüphaneleri kullanılmıştır. Python ve OpenCV tercihinin nedeni, ihtiyaçlara cevap verecek seviyede çok fazla açık kaynak kütüphane içermesi ve önerilen takip yönteminin gerçek zamanlı çalışmalar için geliştirilmesine uygun yapıya olmasıdır.

Yapılan deneysel çalışmalar iki alt başlık altında değerlendirilmektedir. İlk bölümde, renk histogramlarının hesaplanması ve karşılaştırılmasını kullanan klasik PF ile nesne takip yöntemleri, diğeri nesneye ait yapısal özelliklerin kullanıldığı ölçüm modeli ile nesne takibi yapabilen yeni PF yöntemidir.

Her iki başlık altındaki uygulamalar, Intel Pentium Core I5, 3.6 GHz işlemci ve 8GB RAM donanımlarına sahip bilgisayarda gerçekleştirilmiştir.

5.1. Performans Değerlendirmesi (Performans Evaluation)

Bu çalışmada takip edilecek nesne, merkez noktası ve nesneyi çevreleyen dikdörtgen olarak ifade edilmiştir. Takip sırasında kullanılan video imgeleri Object Tracking Benchmark adı verilen doğru nesne takip sonuçlarının yer aldığı çalışmadan elde edilmiştir [21]. [21]'de belirtilen çalışmaya ait önemli özelliklerden biri farklı durum ve senaryolara ait olabilecek pek çok video imgesini bünyesinde barındırıp,

video imgelerini kategoriler halinde araştırmacıların kullanımına sunmuş olmasıdır. Bu sayede araştırmacılara ölçme ve değerlendirme sonuçlarını doğru analiz etme imkânı sunulmaktadır. Sonuçların karşılaştırılması ve performans değerlendirmesi amacıyla en eski ve yaygın olarak kullanılan nesne merkez noktasına göre hata ölçümü yapan değerlendirme esas alınmıştır[22]. Her bir görüntü çerçevesi için elde edilen hata değerini RMSE kullanarak sonuç olarak sunmak mümkündür. Bu kapsamda RMSE değeri (18) ile gösterilen eşitlikteki gibi hesaplanabilir.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N \|X_t^G - X_t^T\|^2} \quad (18)$$

N çerçeve sayısını, X_t^G doğru sonuçlara ait merkez noktasını, X_t^T yöntemimizce tahmin edilen merkez noktasını belirtmektedir. X_t^G değeri [21] numaralı çalışmada kullanıcılar için mevcut olup değerlendirmelerde kullanılmak için araştırmacılar ile paylaşılmıştır. Yöntemimizi değerlendirmek için [21]'de belirtilen beş örnek video (Suv, Girl, Boy, FaceOcc1, Walking) üzerinde nesne takip yöntemlerinin performansları elde edilmiş ve tablolar halinde sunulmuştur. Adaletli bir kıyaslama için algoritmalar aynı parametrelerde koşturulmuştur. Bu parametreler; parçacık sayısı 70, standart sapma değeri 0.01 ve iterasyon sayısı 10 olarak belirlenmiştir. Her bir takip işleminin hata oranı, 10 iterasyon sonucunda elde edilen RMSE değerlerinin ortalaması olarak değerlendirilmiştir. Bu sayede parçacıkların takip başlangıcı sırasında rastgele oluşturulmasından dolayı kaynaklanabilecek olası hatalar en aza indirgenmiştir.

5.2. Renk Bilgisi İle Takip (Tracking With Color Information)

Parçacıklara ait olabilirlik değerlerinin hesaplanmasında ölçüm modeli olarak karşılaştırılan imgelere ait renk histogramları elde edilmiş olup, hedef ve aday imgelere ait histogram değerleri Bhattacharyya benzerlik katsayısı kullanılarak karşılaştırılmıştır. (19) numaralı denklemde ifade edilen Bhattacharyya benzerlik katsayısı karşılaştırılan imgelere ait histogramların benzerlikleri konusunda fikir verme amaçlı kullanılan uzaklık belirleme metriğidir.

$$D(h_1, h_2) = (1 - \sum_{i=1}^B \sqrt{h_{i,1} h_{i,2}}) \quad (19)$$

h_1 ve h_2 sırasıyla aday ve hedef görüntü parçalarının histogramları, ise histogram demeti sayısını göstermektedir. Benzerlik katsayısı (19)'de belirtilen fonksiyon yardımıyla parçacıkların ağırlıklandırılmasında kullanılarak, sonrasal dağılım parçacıkların ağırlığı türünden ifade edilmiştir. Renk temelli PF ile takip iki ayrı yöntem altında irdelenmiştir. Bunlardan ilki sadece renk bilgisini kullanan RT ile takip, diğeri ise filtrelenmiş imgeler kullanarak imge karşılaştırmaları sonucunda takip işlemini gerçekleştiren FRT yöntemidir. Karşılaştırma sonuçları Tablo 1, Tablo 2, Tablo 3'te paylaşılmıştır.

5.3 Yapısal Özellikler Kullanılarak Takip (Tracking By Using Structural Features)

Aday ve hedef nesnelere ait yapısal özellikler kullanılarak elde edilen takip sonuçları, nesnenin yalnızca renk bilgisini kullanarak takip işlemi gerçekleştiren yöntemlerden daha başarılı sonuçlar sunmuştur. Yapısal özellikler kullanılarak yapılan nesne takibindeki başarının temel nedenleri: İmgelere ait adaptif histogram eşitlemesi ile aday ve hedef imge parçalarındaki zıtlık değerinin artırılması, imge merkezine ait piksellerin önem derecelerinin artırılması esasına dayalı işlemlerden sonra SSIM ile karşılaştırılma yapılmasıdır. Durum vektörümüzü oluşturan bileşenlerden olan parçacık hız bilgisinin hesaplanmasında optik akış yönteminden yararlanılarak parçacıkların konum bilgilerinin güncellenmesi yoluna gidilmiştir. Tablo 1 ve Tablo 2'de takip için kullandığımız Suv ve Walking görüntü imgeleri için belirli görüntü çerçevelerine ait RMSE değerleri karşılaştırmalı olarak sunulmuştur. Bu sayede seçilen iki görüntü kümesine ait takip sonucunda oluşan toplam hata değeri yerine, takip boyunca hata değişimi gösterilmiştir. Tablo 3 ise görüntü imgelerinin tamamı için 10 iterasyon sonrasında ortalama hata değerlerini göstermektedir. Yapısal özellikleri kullanarak takip işlemi gerçekleştiren yöntemlerimiz üç ayrı başlık altında değerlendirilmiştir. Bunlardan SSIM sadece yapısal özelliklerin kullanıldığı yöntemi, SSIMFAHE yapısal özelliklerle birlikte imgelerin yumuşatıldığı ve adaptif histogram eşitlemesine tabi tutulduğu yöntemi, MASSIMAHE ise ilave olarak merkez piksellerin ağırlıklandırıldığı yöntemimizi belirtmektedir. Tablo 3'de yer alan verilerden anlaşılacağı üzere "MASSIMAHE" en iyi takip sonucu elde edilen yöntem olarak belirlenmiştir.

Tablo 1. Suv takibinde belirli çerçevelerden alınmış takip sonuçları

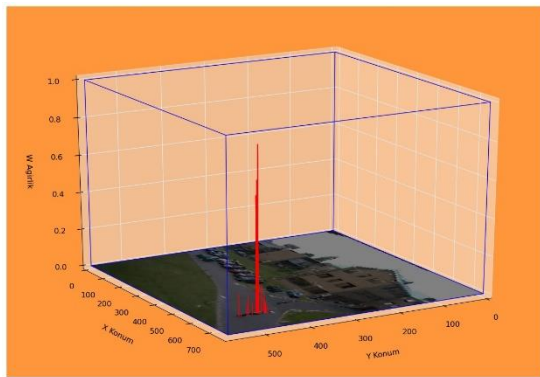
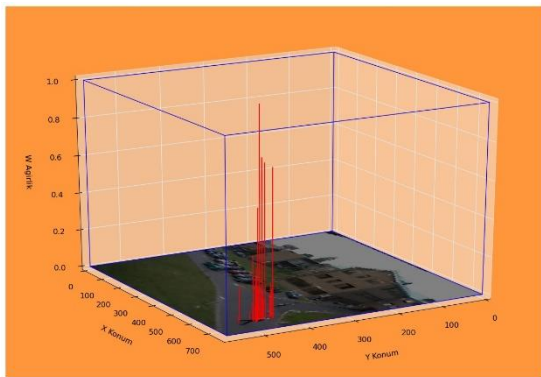
Suv	#75	#150	#300	#450	#600	#750	#900	#945
RT	12,32	14,37	14,29	13,94	15,94	15,98	15,77	16,24
FRT	12,34	15,71	13,70	13,03	14,94	15,10	15,42	15,68
SSIM	10,42	13,19	12,54	11,63	13,88	14,06	14,13	14,55
SSIMFAHE	10,17	10,09	11,30	10,93	13,30	13,16	13,21	13,70
MASSIMAHE	9,32	11,39	10,91	10,60	13,08	12,87	12,60	12,73

Tablo 2. Walking takibinde belirli çerçevelerden alınmış takip sonuçları

Walking	#50	#100	#150	#200	#250	#300	#350	#400
RT	14,07	14,01	13,06	13,46	13,63	14,75	18,17	20,68
FRT	14,41	11,89	11,27	12,39	12,66	13,26	15,38	18,10
SSIM	14,41	13,29	12,24	12,49	12,74	12,48	12,15	12,58
SSIMFAHE	15,17	13,63	12,26	11,62	11,11	10,64	10,00	10,29
MASSIMAHE	13,48	13,04	12,23	11,31	10,64	9,73	9,06	8,90

Tablo 3. Takip yöntemine ait ortalama hata oranları

	RT	FRT	SSIM	SSIMFAHE	MASSIMAHE
Suv	15,65	14,98	14,51	14,75	12,74
Girl	9,30	8,40	5,57	6,05	5,33
Boy	8,87	8,67	4,51	4,95	4,45
Face Occ1	30,44	28,84	10,66	10,69	10,57
Walking	22,05	19,21	12,71	9,82	8,89

**Şekil 4.** Walking video imgelerine ait parçacıkların konum ve ağırlık bilgileri.

Şekil 5’te görüleceği üzere bu çalışmada hedeflenen PF ile nesnelerin uzun süreli takibi adına ilerlemelerin kaydedildiği gözlemlenmiştir. %18.59 iyileştirme oranı, Tablo 3’de belirtilen MASSIMAHE ve RT’ye ait RMSE değerlerinin yüzdelik dilimde düşüşünü ifade etmek için kullanılmıştır. Minimum iyileştirmeyi ifade etmek amacıyla en az iyileştirmenin olduğu SUV imgeleri için MASSIMAHE ve RT değeri kullanılarak $\left(1 - \frac{RT}{MASSIMAHE}\right) \times 100$ eşitliği ile %18.59 değeri elde edilmiştir. Diğer test amaçlı seçilen görüntü kümeleri için hata oranındaki azalma hesaplandığında hatanın daha yüksek oranda azaldığı görülmüştür. Nesnenin tam ve kısmi kapanma gibi durumlarında önerilen yöntemlerimiz herhangi bir

müdahaleye gerek kalmadan nesne takibini gerçekleştirmeye devam etmiştir. Önerilen yöntemimize ait güçlü yönlerden bahsetmek gerekirse; Nesne takibi sırasındaki doğruluk oranındaki artış, Geliştirilen uygulamanın geliştirmeye ve değişik alanlarda kullanılmasına uygun oluşu örneğin insansız hava araçları ile nesne takibi, Uzun süreli nesne takibindeki başarısı olarak sıralanabilir. Şekil 5 takip edilen imgelere ait ekran görüntülerini göstermektedir.



Şekil 5. Nesne takip sonuçları

Her ne kadar takip hızı önerilen yöntem sayesinde azalsa da sistemin hızı kabul edilebilecek seviyede olup bu çalışmada hızla ilgili değerlendirmelerden ziyade takip doğruluğuna odaklanılmıştır. Turuncu dikdörtgen gerçek sonuçları, turkuaz ise önerilen yönteme ait takip sonucunu göstermektedir. Kırmızı ile gösterilen noktalar parçacık konum ve ağırlık bilgisini göstermektedir.

6. SONUÇ (RESULTS)

PF doğrusal olmayan sistemlerin tahmin ve takibi amaçlı kullanılabilecek yöntemler arasında önemli bir yere sahiptir. Bu çalışmada PF kullanarak video dosyalarında uzun zamanlı nesne takibi yapabilen yeni bir yöntem tanıtılmıştır. Önerilen nesne takibi yöntemi, sisteme ait sonrasal dağılımın yeni bir ölçüm metodu kullanarak elde edilmesi yaklaşımını benimsemiştir. Önerilen yöntem ile klasik yaklaşım tarzlarını karşılaştırmak amaçlı Python ve OpenCV kütüphaneleri kullanılarak nesne takip uygulaması geliştirilmiş olup, karşılaştırmalı sonuçlar paylaşılmıştır. Karşılaştırmalı sonuçlar önerilen yöntemin rakiplerine karşı en az %18.59 oranında iyileştirme sağladığını ortaya koymuştur. İleriye dönük olarak yapılacak çalışmalar: Nesnenin yapısal özelliklerini kullanarak takip sırasında gri renkli imgeler yerine renkli imgelerin kullanılması, Takip yönteminin gerçek zamanlı takip işlemine uygun hale getirilmesi ve Geliştirilen nesne takip yazılımının insansız hava araçları yardımıyla havadan otonom nesne takibi çalışmasında kullanılması hedeflenmektedir.

7. KISALTMALAR (ABBREVIATIONS)

PF: Parçacık Filtresi, **SIS:** Sıralı Önem Örneklemesi, **SIR:** Sıralı Önem Yeniden Örneklemesi, **RMSE:** Ortalama Hata Kareleri Toplamı Kökü, **RT:** Renk Temelli Parçacık Filtresi, **SRT:** Filtrelenmiş Renk Temelli Parçacık Filtresi, **SSIM:** Yapısal Benzerlik İndeksi İle Parçacık Filtresi, **SSIMFAHE:** Yapısal

Benzerlik İndeksi Filtrelenmiş Histogram Eşitlenmiş Parçacık Filtresi, **MASSIMAHE**: Merkez Ağırlıklandırılmış Yapısal Benzerlik İndeksi Histogram Eşitlenmiş Parçacık Filtresi.

KAYNAKLAR (REFERENCES)

- [1] M. Islam, C. Oh, and C. Lee, "Video Based Moving Object Tracking by Particle Filter," *Int. J. Signal Process. Image Process. Pattern*, vol. 2, pp. 119–132, 2009.
- [2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [3] G. M. Rao and C. Satyanarayana, "Visual Object Target Tracking Using Particle Filter: A Survey," *Int. J. Image, Graph. Signal Process.*, vol. 5, no. 6, pp. 57–71, 2013.
- [4] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull, "Object Tracking by Particle Filtering Techniques in Video Sequences," *Adv. Challenges Multisens. Data Inf. Process.*, vol. 8, pp. 260–268, 2007.
- [5] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.
- [6] C. Hue, J. Vermaak, and M. Gangnet, "Color-Based Probabilistic Tracking," pp. 661–675, 2002.
- [7] X. Jia and H. Lu, "Visual Tracking via Adaptive Structural Local Sparse Appearance Model," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1822–1829, 2012.
- [8] C. Chen, W. Tarng, and K. Lo, "An Improved Particle Filter Tracking System Based on Colour and Moving Edge Information," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 4, pp. 97–117, 2014.
- [9] K. Ng and E. Delp, "New models for real-time tracking using particle filtering," *IS&T/SPIE Electron. Imaging*, p. 72570B–72570B–12, 2009.
- [10] J. Chun and G. Shin, "Realtime Facial Expression Recognition from Video Sequences Using Optical Flow and Expression HMM," in *Journal of Korean Society for Internet Information*, 2014, pp. 55–70.
- [11] M. Lucena and J. M. Fuertes, "Optical flow-based observation models for particle filter tracking," pp. 135–143, 2015.
- [12] S. Belgacem, A. Ben-hamadou, and T. Paquet, "Hand Tracking Using Optical-Flow Embedded Particle Filter in Sign Language Scenes," pp. 1–8, 2012.
- [13] T. M. F. Dilmen H, "Tek Boyutlu Durum Uzay Değişkenlerinin Parçacık Filtresi Yöntemi İle Takibi Tracking One Dimension State Space Variables With Particle Filter Method," in *Signal Processing and Communications Applications Conference (SIU), 2015 23th, Malatya*, pp. 1513–1516, 2015.
- [14] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *Int Symp AerospaceDefense Sens. Simul Control.*, vol. 3, pp. 182–193, 1997.
- [15] D. L. and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *inProc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 121–130, 1981.
- [16] X. Mei and H. Ling, "Robust Visual Tracking using L1 Minimization," in *IEEE International Conference on Computer Vision*, no. Iccv, pp. 1436–1443, 2009.
- [17] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 125–141, 2007.

- [18]A. Łoza, L. Mihaylova, D. Bull, and N. Canagarajah, “Structural similarity-based object tracking in multimodality surveillance videos,” *Mach. Vis. Appl.*, vol. 20, no. 2, pp. 71–83, 2009.
- [19]Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [20]G. Yadav, S. Maheshwari, and A. Agarwal, “Contrast limited adaptive histogram equalization based enhancement for real time video system,” *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 2392–2397, 2014.
- [21]Y. Wu, J. Lim, and M.-H. Yang, “Online Object Tracking: A Benchmark,” *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2411–2418, 2013.
- [22]L. Cehovin, A. Leonardis, and M. Kristan, “Visual Object Tracking Performance Measures Revisited,” *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1261–1274, 2016.