

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TÜNELLERDE GÜVENLİĞİN SAĞLANMASI AMACIYLA
ARAÇ HAREKETLERİNİN ANALİZİ

YÜKSEK LİSANS TEZİ

Şevket Umut ÇAKIR

KASIM 2009
TRABZON

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**TÜNELLERDE GÜVENLİĞİN SAĞLANMASI AMACIYLA
ARAÇ HAREKETLERİNİN ANALİZİ**

Bilgisayar Mühendisi Şevket Umut ÇAKIR

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“Bilgisayar Yüksek Mühendisi”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 09.10.2009
Tezin Savunma Tarihi : 06.11.2009**

**Tez Danışmanı : Prof. Dr. Vasıf V. NABİYEV
Jüri Üyesi : Yrd. Doç. Dr. Murat EKİNCİ
Jüri Üyesi : Doç. Dr. Ali GANGAL**

Enstitü Müdürü : Prof. Dr. Salih TERZİOĞLU

Trabzon 2009

ÖNSÖZ

Günümüzde karayollarında güvenliğin sağlanması önemli bir sorun olarak görülmektedir. Karayolları tünellerinde gerçekleşen kazalar otoyolların diğer kısımlarında gerçekleşen kazalara oranla daha ağır olabilmektedir. Dolayısıyla tünellerde yaşanan kazalar ve bu kazaların sonucundaki kayıplar bu sorunun önemini daha da artırmaktadır.

Çalışmada hızla kullanımı artan, karayolları tünellerinde kazalara sebep olacak durumların en kısa zamanda tespit edilmesi için otomatik araç takip sistemi geliştirmek amaçlanmıştır. Araçların hareketlerinin analizi amacıyla çeşitli nesne algılama ve nesne takip yöntemleri probleme uygulanıp sonuçları değerlendirilmiştir.

Bu çalışmada danışmanlığımı üstlenen değerli hocam Prof. Dr. Vasif Vagifoğlu NABIYEV'e ilgi, alaka ve yardımlarından dolayı, değerli hocam Yrd. Doç. Dr. Murat EKİNCİ 'ye desteğinden dolayı teşekkürü bir borç bilirim. Ayrıca her zaman yanımda olan aileme ve dostlarıma da destekleri için teşekkür ederim.

Şevket Umut ÇAKIR
Trabzon 2009

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	VI
SUMMARY.....	VII
ŞEKİLLER DİZİNİ.....	VIII
SEMBOLLER DİZİNİ.....	X
1. GENEL BİLGİLER.....	1
1.1. Mevcut Olay Algılama Sistemleri ve Trafik Parametreleri Çıkaran Sistemler.....	1
1.2. Otomatik Olay Algılama.....	3
1.2.1. Görüntü Dizisindeki Nesneleri Algılama.....	4
1.2.1.1. Nokta Algılayıcıları.....	4
1.2.1.1.1. Moravec Köşe Noktası Algılama Algoritması.....	4
1.2.1.1.2. Harris ve Stephens Köşe Noktası Algılama Algoritması.....	5
1.2.1.1.3. Shi ve Tomasi Köşe Noktası Algılama Algoritması.....	7
1.2.1.2. Bölütleme.....	7
1.2.1.2.1. Ortalama Kaydırma Algoritması ile Kümeleme.....	7
1.2.1.3. Danışmanlı Sınıflandırıcılar.....	7
1.2.1.3.1. Adaptif Hızlandırma.....	8
1.2.1.3.2. Destek Vektör Makineleri.....	9
1.2.1.4. Arkaplan Modelleme.....	10
1.2.1.4.1. İlk Çerçeve Arkaplan Modeli.....	11
1.2.1.4.2. Ardışıl Fark Arkaplan Modeli.....	11
1.2.1.4.3. Ortalama Arkaplan Modeli.....	12
1.2.1.4.4. Adaptif Arkaplan Modeli.....	12
1.2.1.4.5. En Çok Tekrarlılık Oranı.....	13
1.2.1.4.6. Temel Bileşen Analizi Yardımıyla Arkaplan Modelinin Belirlenmesi.....	13
1.2.1.4.6.1. Temel Bileşen Analizi.....	13
1.2.1.4.6.2. Temel Bileşen Analizi Metodolojisi.....	14

1.2.1.4.6.3.	Temel Bileşen Analizinin Arkaplan Modellemeye Uygulanması.....	16
1.2.1.4.7.	Gaussian Karışım Modeli Kullanarak Arkaplan Modelini Belirleme.....	18
1.2.1.4.8.	Ağırlık Merkezi Arkaplan Modeli.....	20
1.2.2.	Nesne Takibi.....	21
1.2.2.1.	Nesne Temsili.....	22
1.2.2.1.1.	Noktalar.....	22
1.2.2.1.2.	Basit Geometrik Şekiller.....	22
1.2.2.1.3.	Nesne Silueti ve Sınırları.....	23
1.2.2.1.4.	Eklemlı Şekil Modelleri.....	23
1.2.2.1.5.	İskelet Modelleri.....	23
1.2.2.2.	Nesne Takibi İçin Özellik Seçimi.....	24
1.2.2.2.1.	Renk Özellikleri.....	24
1.2.2.2.2.	Kenar Özellikleri.....	25
1.2.2.2.3.	Optik Akış Özellikleri.....	25
1.2.2.2.4.	Doku Özellikleri.....	25
1.2.2.3.	Ortalama Kayma Algoritması.....	26
1.2.2.4.	Rastlantısal Kestirim.....	27
1.2.2.4.1.	Durum-Uzayı Modelleri.....	27
1.2.2.4.2.	Gözlemci Tasarım Problemi.....	29
1.2.2.4.2.1.	Ölçüm ve Sistem Gürültüsü.....	30
1.2.2.5.	Kalman Filtresi.....	30
1.2.2.5.1.	Ayrık Kalman Filtresi.....	31
1.2.2.5.1.1.	Kestirim Yapılacak Sistem.....	31
1.2.2.5.1.2.	Kalman Filtresinin Hesapsal Tabanı.....	32
1.2.2.5.1.3.	Ayrık Kalman Filtresi Algoritması.....	33
1.2.2.6.	Mahalanobis Uzaklığı.....	35
1.2.2.7.	Veri İlişkilendirmesi.....	36
1.2.2.8.	Çoklu Hipotez Takibi Algoritması.....	38
1.2.2.8.1.	Güzergâh ve Hipotez Değerlendirmesi.....	40
2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME.....	42
2.1.	Giriş.....	42
2.2.	Nesne Algılama Yöntemlerinin Tünel Görüntülerine Uygulanması.....	44
2.2.1.	İlk Çerçeve Yöntemi.....	44

2.2.2.	Ardışıl Fark Yöntemi.....	45
2.2.3.	Ortalama Yöntemi.....	45
2.2.4.	Adaptif Arkaplan Yöntemi.....	46
2.2.5.	ETO Algoritması.....	47
2.2.6.	PCA ile Arkaplan Modelleme Yöntemi.....	48
2.2.7.	GMM Kullanarak Arkaplan Modelleme Yöntemi.....	49
2.3.	Yatay ve Dikey Kesit Değerleri ile Nesne Algılama.....	51
2.4.	Nesne Takibi.....	52
2.4.1.	Ortalama Kaydırma Algoritmasının Uygulanması.....	52
2.4.2.	Görüntüdeki Yol Bölgesinin Poligonlar ile Seçilmesi.....	57
2.4.3.	Kalman Filtresinin Probleme Uygun Modellenmesi.....	59
2.4.4.	Kalman Filtresinin Araç Verilerine Uygulanması.....	61
2.4.5.	Temel Bileşenler ile Güzergâh Aralığının Belirlenmesi.....	63
2.4.6.	Çoklu Hipotez Takibinde Güzergâh-Gözlem Sonucu İlişkilendirilmesi.....	65
3.	SONUÇLAR.....	68
4.	ÖNERİLER.....	70
5.	KAYNAKLAR.....	71
ÖZGEÇMİŞ		

ÖZET

Motorlu araçların kapalı mekânlarda kullanılması sırasında kazaların oluşması, çok ciddi sorunlar doğurması nedeniyle kazalara sebebiyet verebilecek olayların önceden tahmin edilerek saptanması gerekmektedir. Karayolları tünellerinde gerçekleşen kazalar otoyolların diğer kısımlarında gerçekleşen kazalara oranla daha ağır olabilmektedir. Özellikle kazalardan sonra oluşan yangınlar vahim sonuçlar doğurabilmektedir.

Çalışmada hızla kullanımı artan karayolları tünellerinde kazalara sebep olacak durumların en kısa zamanda tespit edilmesi amacıyla otomatik araç takip sistemi geliştirilmesi hedeflenmiştir. Bunun gerçekleştirilmesi için tünel içinden alınan görüntülerin filtrelenmesi ve gürültülerden arındırılması, ham görüntü üzerinde nokta algılayıcılar, bölütleme, arkaplan modelleme yöntemlerinin gerçekleştirilerek nesne algılaması işlemleri yapılmıştır. Algılanan nesnelere nesne takibi için çeşitli yöntemler uygulanmıştır.

Çalışmanın uygulamaya konulabilmesi durumunda tünellerde yaşanan kaza oranında ciddi bir düşüş gözleneceği öngörülmektedir. Yapılan literatür taramalarında bu konuya yönelik yapılan çalışmalar olduğu tespit edilmiştir. Bu çalışmalardan faydalanarak ortaya daha etkin çalışabilecek yeni bir sistem koyulması düşünülmüştür.

Anahtar Kelimeler: Otomatik Olay Algılama, Nesne Takibi, Arkaplan Modelleme, Bilgisayarla Görme

SUMMARY

Analysis of Vehicles' Motion for the Purpose of Providing Tunnel Safety

Accidents occurring in tunnels by usage of vehicles in these environments can cause serious problems. Detecting incidents which may cause accidents can prevent loss of lives and properties. Accidents occurring in motorway tunnels may be more harmful when they are compared to the ones in highways. Especially effects of fire starting in tunnels following accidents are serious.

In our work, it is aimed that developing a system which can detect incidents automatically. For automatically detecting incidents, a vehicle tracking approach is proposed. Algorithms such as point detectors, segmentation and background modeling are used in order to achieve object detection task. Various object tracking methods are applied on the moving objects which are detected in previous steps.

In case of applying an automatic incident detection system to real world, it is estimated that accidents occurring in tunnels will decrease. By making a literature review presence of such systems are seen. With help of these works in literature, developing a more effective system is aimed.

Key Words: Automatic Incident Detection, Object Tracking, Background Modeling, Computer Vision

ŞEKİLLER DİZİNİ

		<u>Sayfa No</u>
Şekil 1.1.	Otomatik olay algılama sistemlerinin genel akış şeması.....	3
Şekil 1.2.	Nesne algılama yöntemlerinin sınıflandırılması.....	4
Şekil 1.3.	Özellikleri çıkartmak kullanılan dikdörtgen filtreler.....	9
Şekil 1.4.	ETO algoritmasında piksel değerlerinin modellenmesi.....	13
Şekil 1.5.	Nesne temsilleri.....	24
Şekil 1.6.	Kalman filtresinin bir döngüsünde yapılan işlem adımları.....	34
Şekil 1.7.	İki farklı güzergâh için güzergâh aralıkları ve elde edilen gözlem sonuçları.....	37
Şekil 1.8.	Örnek veri ilişkilendirmesi için oluşturulan hipotezler.....	39
Şekil 2.1.	Olay algılama sistemi için düşünülen sistemin genel yapısı.....	42
Şekil 2.2.	İlk çerçeve yöntemine göre nesne algılama.....	44
Şekil 2.3.	Ardışıl fark yöntemine göre nesne algılama.....	45
Şekil 2.4.	Ortalama yöntemine göre nesne algılama.....	46
Şekil 2.5.	Adaptif arkaplan yöntemine göre nesne algılama.....	47
Şekil 2.6.	ETO algoritmasına göre nesne algılama.....	48
Şekil 2.7.	PCA ile arkaplan modelleme yöntemine göre nesne algılama.....	49
Şekil 2.8.	GMM kullanarak arkaplan modelleme yöntemine göre nesne algılama.....	50
Şekil 2.9.	ETO, PCA ve GMM yöntemlerinin süre karşılaştırması.....	50
Şekil 2.10.	Yatay ve dikey kesit değeri ile nesne algılanmasına bir örnek.....	51
Şekil 2.11.	Örnek histogram dağılımı.....	53
Şekil 2.12.	Ortalama kaydırma algoritması ile çoklu nesne takibini gerçekleştirme için önerilen sistem yapısı.....	54
Şekil 2.13.	Tek araç için oluşturulan ortalama kaydırma olasılık resmi.....	56
Şekil 2.14.	İki araç için oluşturulan ortalama kaydırma olasılık resimleri.....	57
Şekil 2.15.	Poligonları kullanarak yol seçmek için geliştirilen uygulamanın ekran görüntüsü.....	59
Şekil 2.16.	Örnek araç verisi Kalman filtresinin tahmini ile nesne algılama yönteminin belirlediği konum arasındaki hata değerleri.....	62

Şekil 2.17.	Örnek araç verisi için Kalman filtresinin tahmini ile nesne algılama yönteminin belirlediği konumlar ve kesin referans bilgisi arasındaki hata değerleri.....	63
Şekil 2.18.	Temel bileşenler ile güzergâh aralığını belirlemek için geliştirilen uygulamanın ekran görüntüsü.....	64
Şekil 2.19.	Örnek ilişkilendirme problemi için oluşturulan hipotezlerin ağaç yapısında gösterimi.....	65

SEMBOLLER DİZİNİ

ITS	Akıllı Ulaşım Sistemleri (Intelligent Transportation Systems)
Adaboost	Adaptif Hızlandırma (Adaptive Boosting)
SVM	Destek Vektör Makineleri (Support Vector Machines)
PCA	Temel Bileşen Analizi (Principal Component Analysis)
GMM	Gaussian Karışım Modeli (Gaussian Mixture Model)
ETO	En Çok Tekrarlanma Oranı
GNN	Genel En Yakın Komşuluk (Global Nearest Neighbor)
JPDA	Birleşme Yeri Olasılıksal Veri İlişkilendirmesi (Joint Probabilistic Data Association)
MHT	Çoklu Hipotez Takibi (Multiple Hypothesis Tracking)
LR	Olabilirlik Oranı (Likelihood Ratio)
LLR	Logaritmik Olabilirlik Oranı (Log-Likelihood Ratio)
λ	Özdeğerler
v	Özvektörler
$I(x,y)$	Kameradan alınan görüntü çerçevesinin (x,y) konumundaki renk değeri
$B(x,y)$	Arkaplan görüntüsü
$D(x,y)$	Arkaplan ile kamera görüntüsünün farkı
μ	Ortalama değer
C	Kovaryans matrisi
$N(\mu, \Sigma)$	Ortalaması μ , varyansı Σ olan normal dağılım
σ	Standart sapma
x_k	k anındaki sistem durumu
\hat{x}_k	k anındaki sistem durum kestirimi
A	Sistem durumları arasındaki geçiş matrisi
H	Sistem durumları ile ölçümler arasındaki geçiş matrisi
z_k	k anındaki ölçüm değeri
w	Sistem gürültüsü
v	Ölçüm gürültüsü
u_k	k anındaki kontrol girişleri
B	Kontrol girişleri ile sistem durumu arasındaki ilişki matrisi

Q	Sistem gürültüsü kovaryansı
R	Ölçüm gürültüsü kovaryansı
K	Kalman kazancı
\hat{x}_k^-	k anında sistem durumunun önsel kestirimi
$L(k)$	k anındaki güzergâh puanı
\hat{P}_D	Beklenen nesne algılama oranı

1. GENEL BİLGİLER

Günümüzde karayollarındaki trafiğin giderek artması yaşanan kazaların artmasını da beraberinde getirmektedir. Bu kazalar can ve mal kaybına, insanların sakatlanmasına ve trafik sıkışıklıklarına neden olabilmektedir.

Yapılan araştırmalar göstermiştir ki, karayollarında yaşanan olayların %20 ile %50 arası daha önce gerçekleşen ve birincil olay adı verilen olayların etkilerinden dolayı yaşanmaktadır [1]. Birincil olayları takip eden ve birincil olayların etkileri sonucu meydana gelen bu olaylara ikincil olay adı verilir. Yaşanan ikincil olaylar, birincil olayların gerçekleşmesinden sonraki 10 dakikalık süre içerisinde gerçekleşmektedir. Birincil olayların anında algılanıp, olay yerine doğru seyretmekte olan sürücülerin bilgilendirilmesi muhtemelen ikincil olayların yaşanmasını engelleyecektir. Japonya'daki Metropolitan Express yolunda yapılan bir araştırmaya göre bu yoldaki trafik sıkışıklıklarının %20'si yaşanan trafik kazaları ve devre dışı kalan araçlar nedeniyle oluşmaktadır [2].

Günümüzde, yukarıda anlatılan nedenlerden dolayı, otomatik trafik denetleme sistemleri bir gereklilik haline gelmiştir. Otomatik trafik denetleme sistemlerinin uygulamaya koyulmasıyla birlikte kazalar sonucu oluşan trafik sıkışıklıkları, can ve mal kayıplarının kısmen önüne geçilebilir.

1.1. Mevcut Olay Algılama Sistemleri ve Trafik Parametreleri Çıkaran Sistemler

Versavel, Akıllı Ulaşım Sistemleri(ITS)'ni test etmek amacıyla Belçika'da Gent il Antwerp arasındaki karayolunu pilot bölge olarak seçmiştir [1]. Bu yol üzerindeki kameralardan alınan görüntüleri inceleyerek günlük trafik sıkışıklığı ile kazalardan kaynaklanan trafik sıkışıklığını ayırt etmiştir. Versavel, video sinyallerine dayalı trafik verilerinin alınmasının güvenilir ve uygun maliyetli bir çözüm olduğu kanısına varmıştır.

McLauchlan, trafikteki araçları izleyerek aşağıdaki trafik parametrelerini çıkarmaktadır [3]:

- Akış: Saat başına düşen araç sayısı
- Hız: Ortalama araç hızı
- Yoğunluk: Birim mesafedeki araç sayısı
- Aralık: Araçlar arasındaki ortalama boşluk

Araçları izlemek için özellik tabanlı bir izleme yöntemi kullanmışlardır. Resimler içerisinde bulunan köşe noktaları izleme özelliklerini oluşturmaktadır. Köşe noktalarının bulunmasına sonraki bölümlerde değinilecektir.

Yu, ardışık görüntü dizilerini kullanarak çok şeritli yollarda araç sayısı ve hız kestirimi gibi parametreleri çıkarmıştır [4]. Görüntü dizilerindeki araçları algılamak için adaptif bir arkaplan modeli kullanmış ve arkaplan çıkarma yapmıştır. Ayrıca kenar algılama teknikleri kullanarak görüntülerdeki gölgeli kısımları elimine etmiştir.

Ikeda, Metropolitan Express yolunda uyguladığı, görüntü işlemeyle dayalı otomatik olay algılama sisteminde aşağıdaki tipteki olayları algılayabildiğini söylemektedir [2]:

- Duran araçlar
- Normalden yavaş seyreden araçlar
- Araçlar üzerinden düşen nesnelere
- Başarılı şerit değiştirme teşebbüsleri

Ikeda, oluşturulan bir arkaplan resminden mevcut resmi çıkartarak hareketli araçları algılamıştır. Araçların zaman içerisindeki konum değişimlerinden de hız kestirimi yapmaktadır. Fakat arkaplan resminin oluşturulması ve hız kestiriminin nasıl yapıldığı yeterince açıklanmamıştır.

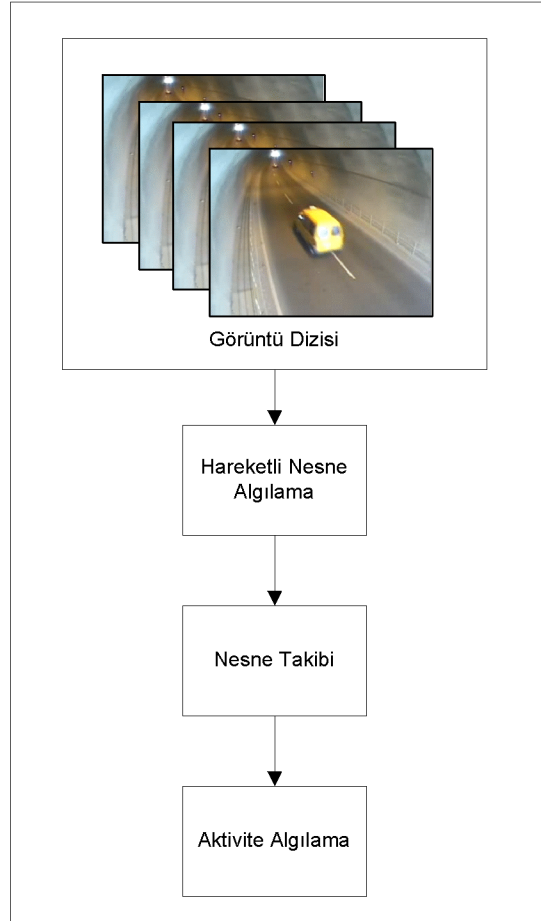
Kimanchi, olaylara sebebiyet verebilecek olan anormal araç davranışlarını incelemiştir [5]. Olayları gerçekleştikten sonra algılamak yerine, olay gerçekleşmeden önce görüntü işleme ve bulanık mantık teorisi kullanarak olayın gerçekleşeceğini tahmin etme yoluna gitmişlerdir.

Veerarghavan, Gaussian Karışım Modellerini kullanarak araç izlemeye dayalı bir otomatik olay algılama sistemi geliştirmiştir [6]. Araç veya nesnelere sahne içerisindeki konumunu kestirmek için genişletilmiş Kalman Filtresi kullanmıştır. Sahne içerisindeki belirli noktaları kullanarak kamera kalibrasyonunu gerçekleştirmiş ve 2 boyutlu resim düzlemindeki nesne konumlarını gerçek dünya koordinatlarına dönüştürmüştür. Bu sayede sahne içerisindeki nesnelere gerçek konum, hız gibi bilgilerini elde etmiştir. Olay algılama kısmında ise sadece araçların çarpışmasının algılanması ile ilgilenmiştir.

Yüksek performanslı bir olay algılama sisteminin kısa ortalama algılama zamanı ve yüksek algılama oranı gibi 2 parametresinin olması beklenir. Bu özellikler sayesinde olayın doğrulanması ve gelen sürücülerin uyarılması gibi sonraki işlemler daha hızlı ve kararlı bir şekilde yapılabilir.

1.2. Otomatik Olay Algılama

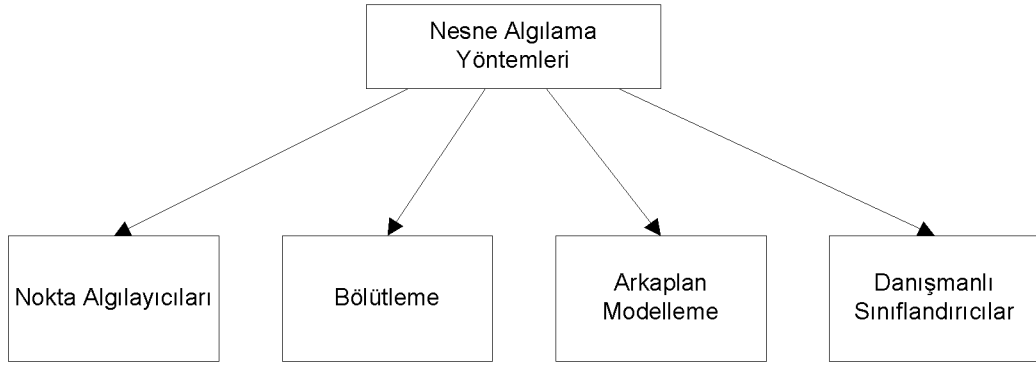
Otomatik olay algılama sistemleri genellikle hareketli nesne algılama, algılanan hareketli nesnelerin takibi ve aktivite anlama adımlarından oluşur. Bu sistemin akışı Şekil 1.1. 'de verilmiştir.



Şekil 1.1. Otomatik olay algılama sistemlerinin genel akış şeması

1.2.1. Görüntü Dizisindeki Nesnelere Algılama

Nesne algılama, otomatik olay algılamanın ikinci adımı olan nesne takibini gerçekleştirmek için gerekli bir işlemdir. Nesne algılama yaklaşımları her bir tekil görüntü çerçevesindeki nesnelere algılamak şeklinde olabileceği gibi görüntü dizisinin zamansal bilgisini kullanarak istatistiksel olarak da yapılabilir. Nesne algılama yöntemlerini sınıflandıırırsak Şekil 1.2. 'deki gibi 4 ana gruba ayırabiliriz.



Şekil 1.2. Nesne algılama yöntemlerinin sınıflandırılması

1.2.1.1. Nokta Algılayıcıları

Nokta algılayıcıları görüntünün dokusundaki ayırt edici kısımlar olan önemli noktaları algırlar. Bu noktalar genellikle köşe noktaları olarak adlandırılırlar. Köşe noktaları iki kenarın kesiştiği noktadır. Başka bir deyişle bir noktanın yerel bir komşuluğunda farklı yönde ve baskın iki kenar varsa, o noktaya köşe noktasıdır denir.

Nokta algılayıcılarının kalitesi, ışık seviyesi ve kamera görüş açısı değişimindeki değişmezliği ile ölçülür. Nokta algılayıcıları genellikle sağlıklı veriler üretmezler. Bir uzmanın denetiminden geçerek veya çok sayıda noktanın elimine edilmesiyle sağlıklı duruma getirilebilirler.

1.2.1.1.1. Moravec Köşe Noktası Algılama Algoritması

İlk köşe noktası algılama algoritmalarındandır [7]. Algoritma görüntü içerisindeki bütün piksellerin köşe noktası olup olmayacağına bakar. Merkezi ilgili piksel olan görüntü parçasının, yakınındaki ve iç içe geçen diğer parçalarıyla olan benzerliğini dikkate

olarak köşe noktası tespiti yapar. Köşe noktası tespiti için yatay, dikey, köşegen ve ters-köşegen yönlerindeki komşu parçaları dikkate alır. Benzerlik ölçütü olarak parçalar içerisindeki piksel değerlerinin farklarının kareleri toplamını alır. Toplam değeri küçük olan bölgeler birbirine daha benzerdir.

İlgili piksel, tekbiçimli parlaklık değerine sahip bir bölge içerisindeyse yakındaki parçalar benzer olacaktır. Eğer piksel bir kenar bölgesindeyse kenara paralel yönde olan parçalar değerlendirildiğinde benzer olurken, kenara dik yönde olan parçalar oldukça farklı olacaktır.

Köşe noktasının gücü o parça ile yatay, dikey, köşegen ve ters-köşegen yönündeki yakın parçalarla olan farkların kareleri toplamı ile tanımlanır. Eğer bu değer yerel olarak yeterli büyüklükte ise aranan özellik burada bulunmuş demektir.

Moravec 'in de belirttiği gibi bu yöntemdeki esas problem yöntemin yönbağımsız olmamasıdır. Eğer komşu yönlerin haricinde bir kenar mevcutsa bu kısım bir köşe noktası olarak algılanmayacaktır.

1.2.1.1.2. Harris ve Stephens Köşe Noktası Algılama Algoritması

Harris ve Stephens, Moravec 'in yöntemini geliştirerek; kaydırılmış parçalar kullanmak yerine direkt olarak yöne dayalı olan köşe noktasının diferansiyelini dikkate almışlardır [8]. Bu algorithmada tutulan skor değeri literatürde genellikle otokorelasyon değeri olarak belirtilse de aslında tuttukları skor değeri farkların karelerinin toplamıdır.

Algoritmayı daha kolay açıklamak için 2 boyutlu gri seviye bir resim üzerinde çalıştığımızı varsayalım. Bu resme I diyelim. (u,v) alanlı bir resim parçası aldığımızı ve bunu (x,y) kadar kaydırığımızı düşünelim. İki parça arasındaki farkların karelerinin ağırlıklı toplamı, S , (1.1) bağıntısıyla hesaplanır.

$$S(x, y) = \sum_u \sum_v w(u, v) (I(u, v) - I(u + x, v + y))^2 \quad (1.1)$$

$I(u + x, v + y)$ 'yi Taylor açılımını kullanarak yaklaşım yaparsak aşağıdaki gibi yazabiliriz.

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y \quad (1.2)$$

Sonuçta (1.3) bağıntısı ile S 'yi elde ederiz.

$$S(x, y) \approx \sum_u \sum_v w(u, v) (I_x(u, v)x + I_y(u, v)y)^2 \quad (1.3)$$

I_x ve I_y , I 'nin kısmi türevleridir. Başka bir ifade ile S 'yi (1.4) bağıntıyla ifade edebiliriz.

$$S(x, y) \approx (x \ y) A \begin{pmatrix} x \\ y \end{pmatrix} \quad (1.4)$$

A yapı tensörü olarak adlandırılır ve aşağıdaki bağıntı ile ifade edilir.

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \quad (1.5)$$

$\langle \text{ ve } \rangle$ parantezleri ortalamayı ifade eder((u, v) üzerinde toplama). Eğer Gaussian gibi çembersel ağırlıklandırılmış bir pencere kullanılırsa, sonuç yönbağımsız olacaktır.

A matrisi yönlü parlaklık değişimlerini ifade ettiği için A matrisinin özdeğerleri bize noktanın köşe noktası olup olmadığı bilgisini verir. A matrisinin özdeğerleri λ_1 ve λ_2 olsun,

- Eğer $\lambda_1 \approx 0$ ve $\lambda_2 \approx 0$ ise (x, y) noktası ile ilgilenilmez,
- Eğer $\lambda_1 \approx 0$ ve λ_2 pozitif büyük bir değere sahipse bir kenar noktası bulunmuştur,
- Eğer λ_1 ve λ_2 büyük pozitif değerlere sahipse incelenen nokta köşe noktasıdır.

Harris ve Stephens, özdeğer ve özvektör hesabının zaman alan bir işlem olduğunu belirterek M_c değerinin hesaplanmasını önermişlerdir. Bu işlemle birlikte sadece matrisin determinant ve izini hesaplamak yeterli olmaktadır. κ deneysel olarak belirlenen bir sabittir.

$$M_c = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(A) - \kappa \text{iz}^2(A) \quad (1.6)$$

1.2.1.1.3. Shi ve Tomasi Köşe Noktası Algılama Algoritması

Bu algoritma Harris köşe noktası algılama algoritmasına dayanır. Shi ve Tomasi afin dönüşümlere uğramış resim parçalarında $\min(\lambda_1, \lambda_2)$ değerini seçmenin M_c değerini seçmenin daha iyi bir ölçüt olacağı kararına varmışlardır [9].

1.2.1.2. Bölütleme

Bölütlemenin amacı resmi algısal olarak benzer bölgelere ayırmaktır. Bölütlenen bölgeler görüntünün diğer kısımlarından soyutlanır.

1.2.1.2.1. Ortalama Kaydırma Algoritması ile Kümeleme

Comaniciu ve Meer, görüntü bölütlemeyi gerçekleştirmek için görüntü ve renk uzayının birleşiminden oluşan uzayda $([l, u, v, x, y])$ ortalama kaydırma yöntemini kullanarak bir kümeleme yöntemi önermişlerdir [10]. $[l, u, v]$ rengi, $[x, y]$ ise uzaysal konumu ifade eder. Algoritma, verilen bir resim için, veri içerisinde rastgele seçilen yüksek sayıdaki varsayılan kümelenme merkezleri ile başlatılır. Daha sonra her bir kümelenme merkezi, verinin içerisinde bulunduğu ve kümelenme merkezinde olan çok boyutlu elipsoidin ortalamasına doğru kaydırılır. Eski ve yeni kümelenme merkezleri arasındaki vektöre ortalama kaydırma vektörü denir. Ortalama kaydırma vektörü kümelenme merkezleri sabit kalana kadar iteratif olarak hesaplanır.

Ortalama kaydırmaya dayalı bölütleme, çeşitli parametrelerin iyi ayarlandığı durumda iyi bir bölütleme gerçekleştirir. Örneğin renk ve uzaysal çekirdek bant genişliklerinin ve bölütleme için seçilen minimum alan eşiğinin seçimi sonuç bölütlemeleri üzerinde büyük etki göstermektedir.

1.2.1.3. Danışmanlı Sınıflandırıcılar

Nesne algılama, nesnelerin farklı bakış açılarından alınan görüntülerinden oluşan veri kümesini danışmanlı öğrenme mekanizmaları kullanılarak gerçekleştirilebilir. Danışmanlı öğrenme yöntemleri, verilen eğitim setini kullanarak, verilen girişlere göre istenilen çıkışları üreten bir fonksiyon oluştururlar. Danışmanlı öğrenme yöntemleri bir

fonksiyonun davranışını kestirerek ya sürekli bir değer üretirler (regresyon) ya da bir sınıf etiketi üretirler (sınıflandırma). Nesne algılama için eğitim örnekleri bir nesne sınıfının hesaplanan birtakım özellikleri olarak seçilebilir.

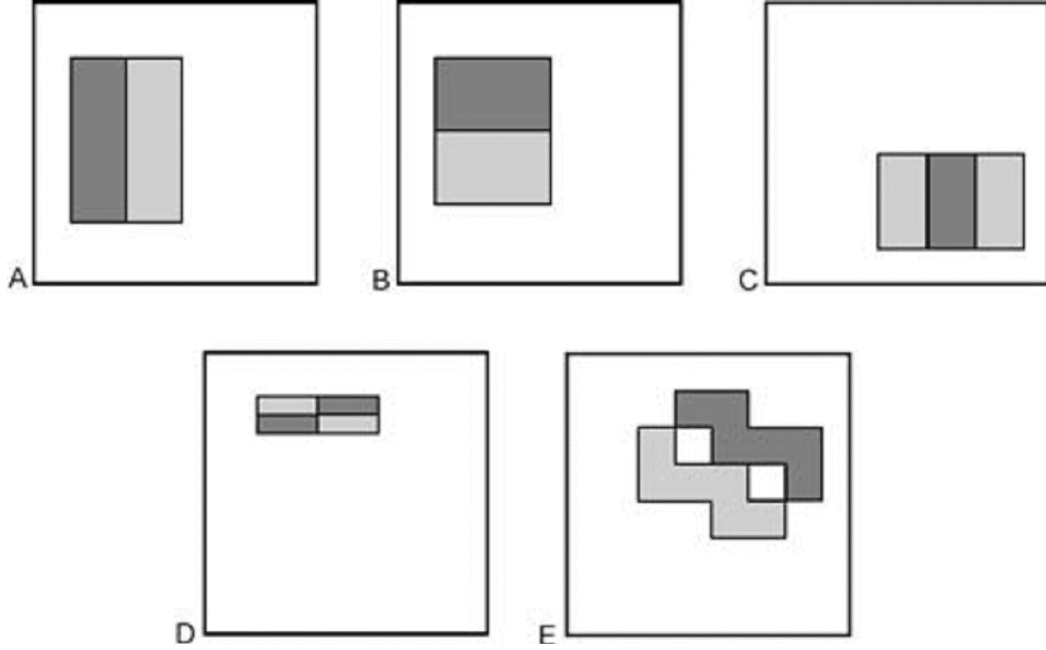
Nesne algılama için seçilen nesne özelliklerinin sınıflandırıcının performansında büyük etkisi vardır. Özellik seçimi yapıldıktan sonra, nesnelerin farklı bakış açılarından çıkartılan bu özellikler danışmanlı sınıflandırıcılar tarafından öğrenilebilir. Nesne algılama için kullanılan danışmanlı öğrenme yöntemleri, bunlarla sınırlı olmamakla birlikte, yapay sinir ağları [11], adaptif hızlandırma [12], karar ağaçları [13] ve destek vektör makinelerini [14] içerir. Bu öğrenme yöntemleri yüksek boyutlu uzaylarda nesne sınıflarını birbirinden ayıran bir aşırıyüzey hesaplarlar.

Danışmanlı öğrenme yöntemleri genellikle her bir nesne sınıftan çok sayıda örneğin olmasına ihtiyaç duyarlar. Ayrıca eğitim seti olarak adlandırılan bu örnek kümesinin el ile etiketlenmesi gerekir.

1.2.1.3.1. Adaptif Hızlandırma

Adaptif hızlandırma(Adaboost), her biri kısmen hatasız olan temel sınıflandırıcıları birleştirerek hatasız bir sınıflandırıcı bulma prensibine dayanır [15]. Adaboost algoritmasının eğitim aşamasındaki ilk adımı eğitim seti üzerinde başlangıç ağırlık dağılımının oluşturulmasıdır. Adaboost mekanizması daha sonra eğitim seti üzerinde en az hatayı veren temel sınıflandırıcılardan birisini seçer. Buradaki hata yanlış sınıflandırılan veri ağırlıklarıyla orantılıdır. Daha sonra, temel sınıflandırıcı tarafından yanlış sınıflandırılan verilerle ilişkili olan ağırlıklar artırılır. Bu şekilde algoritmanın bir sonraki adımında yanlış sınıflandırılan verileri daha iyi sınıflandırabilecek bir sınıflandırıcı seçmesi sağlanır [16].

Nesne algılamada kullanılan zayıf sınıflandırıcılar, görüntülerden elde edilen nesne özelliklerine uygulanan basit operatörler olabilir. Viola vd., Adaboost mekanizmasını görüntüdeki yayaları algılamak için kullanmışlardır [12]. Bu yaklaşımda zamansal ve uzaysal operatörlerin kombinasyonu görüntülerden çıkarılan özellikleri öğrenen zayıf sınıflandırıcılar vardır. Uzaysal operatörler Şekil 1.3. 'te bulunan dikdörtgen filtrelerden oluşur.



Şekil 1.3. Özellikleri çıkartmak kullanılan dikdörtgen filtreler [12].

Her filtre 3 bölgeden oluşur(beyaz, açık gri, koyu gri). Özellik çıkarımı yapmak için bu filtreler ile resim arasında katlama işlemi gerçekleştirilir. Beyaz rengin katsayısı 0, açık grinininki -1 ve koyu grinininki 1'dir.

Zamansal operatörlerde ise çerçeve farkı alma uygulanır. Çerçeve farkı alınarak sadece hareket algılanan bölgelerde inceleme yapılır. Bu hem arama uzayını daraltır hem de hatalı algılamaların sayısını azaltır.

1.2.1.3.2. Destek Vektör Makineleri

Destek Vektör Makineleri(SVM), bir nesne sınıfını diğerinden ayırt etmek için maksimum bir aşırıyüzey bulur. Aşırı yüzeyin sınırı, aşırıyüzey ile ona en yakın olan noktaların arasındaki bir uzaklık değeri ile tanımlanır. Aşırıyüzeyin sınırları boyunca uzanan veri noktalarına destek vektörleri denir. Nesne algılama açısından bu sınıflar nesne sınıfı(pozitif örnekler) ve nesne olmayan sınıfa(negatif örnekler) karşılık gelir. Sınıfları ayıran bu aşırıyüzey, el ile etiketlenen eğitim verilerini kullanarak, sonsuz sayıdaki mümkün aşırıyüzey arasından ikinci derece (kuadratik) programlama icra edilerek ortaya çıkarılır.

Doğrusal bir sınıflandırıcı olan SVM, giriş vektörüne çekirdek hilesini uygulayarak doğrusal olmayan bir sınıflandırıcı olarak kullanılabilir. Doğrusal olarak ayrılmayan veri kümesine çekirdek hilesini uygulamak, veriyi daha büyük boyutlu bir uzaya taşıyarak neredeyse doğrusal olarak ayrılabilen bir küme haline getirir. Çekirdek hilesi için kullanılan çekirdeklere radyal tabanlı fonksiyon(Örn: Gaussian çekirdeği, iki katmanlı algılayıcı) ve çokterimli fonksiyonlar örnek verilebilir.

Mevcut problem için çekirdek fonksiyonunun seçimi kolay değildir. Çekirdek seçildikten sonra sınıflandırmanın performansı çeşitli parametreleri ayarlayarak test edilerek artırılabilir. Fakat eğitim setine eklenen yeni verilerle birlikte sınıflandırmanın performansı eskisi gibi olmayabilir.

Papageorgiou görüntüdeki yayaları ve yüzleri algılamak için SVM kullanmıştır [14]. Sınıfları birbirinden ayırt etmek için kullanılan özellikler pozitif ve negatif veri kümesine Haar dalgacıkları uygulanarak çıkartılmıştır. Resim içerisindeki arama zamanını azaltmak için optik akış hesaplanarak zamansal bilgi kullanılmıştır.

1.2.1.4. Arkaplan Modelleme

Nesne algılama, görüntüdeki sahnenin temsil edilmesi için arkaplan modeli denilen bir model oluşturarak ve her yeni gelen görüntü çerçevesinde bu modelden olan sapmalar dikkate alınarak gerçekleştirilebilir. Bir görüntü bölgesindeki önemli bir değişim hareketli bir nesnenin varlığını belirtir. Genellikle bağlı bileşen etiketleme yapılarak nesnelere temsil eden birbirine bağlı bölgeler ayırt edilir. Anlatılan bu sürece arkaplan çıkarma adı verilir.

Arkaplan çıkarma işlemi piksel tabanlı olabileceği gibi blok tabanlı da olabilir. Piksel tabanlı arkaplan çıkarma işleminde arkaplan modeli her bir tekil piksel için ayrı ayrı modellenir. Arkaplan çıkarma işlemi de piksellerin kendileri için olan mevcut modelden sapmalarıyla belirlenir. Blok tabanlı arkaplan çıkarma işleminde ise tekil pikseller yerine görüntüler bloklara bölünür. Modelleme bu bloklar üzerinde yapılır. Oral, ağırlık merkezine dayalı, blok tabanlı bir arkaplan modelleme yapmıştır [17]. Piksel tabanlı arkaplan çıkarma işlemi genel olarak aşağıdaki bağıntıyla ifade edilebilir:

$$D(x, y) = \begin{cases} 1 & |I(x, y) - B(x, y)| > \tau \\ 0 & \text{aksi halde} \end{cases} \quad (1.7)$$

Burada $B(x,y)$ görüntüdeki (x,y) konumundaki piksel için arkaplan rengini, $I(x,y)$ yeni gelen resmin (x,y) konumundaki pikselin parlaklık değerini, τ ise arkaplan ve önplan bölgelerini birbirinden ayırmak için deneysel olarak seçilen eşik değerini belirtir. $B(x,y)$ ile $I(x,y)$ farkı τ değerini aşarsa resimdeki ilgili piksel “1” (yani hareketli bir nesne) olarak, diğer durumlarda ise “0” (yani hareketsiz arkaplan bölgesi) olarak etiketlenir.

Arkaplan modelleme için birçok yöntem öne sürülmüştür. Bunlardan bazıları bölümün alt maddelerinde açıklanmıştır.

1.2.1.4.1. İlk Çerçeve Arkaplan Modeli

Alınan görüntü dizisinin ilk çerçevesi arkaplan modeli olarak belirlenir. Daha sonra gelen görüntüler ilk çerçeveden çıkarılarak görüntü içerisindeki hareketli kısımlar bulunur. Aşağıdaki formülle ifade edilir.

$$B_t(x, y) = I_0(x, y) \quad (1.8)$$

$B_t(x, y)$, t anındaki arkaplan modelinin (x,y) konumundaki parlaklık değeridir. Bu da $I_0(x, y)$, yani başlangıç anındaki görüntü dizisinin (x,y) konumundaki parlaklık değeridir.

İlk Çerçeve arkaplan modelleme yönteminde sahnenin başlangıçta boş olduğu varsayılır. Eğer sahne içerisinde hareketli herhangi bir cisim varsa bu da arkaplan olarak algılanacaktır ve hatalara yol açacaktır.

1.2.1.4.2. Ardışıl Fark Arkaplan Modeli

Ardışıl Fark Arkaplan Modeli bir önceki çerçeveyi arkaplan modeli olarak kabul eden yöntemdir.

$$B_t(x, y) = I_{t-1}(x, y) \quad (1.9)$$

$B_t(x, y)$ t anında arkaplan modelinde konumu (x, y) olan pikselin parlaklık değeridir. $I_{t-1}(x, y)$ ise $t-1$ anında gelen çerçevede konumu (x, y) olan pikselin parlaklık değeridir.

Bu yöntemde eğer nesne büyük ve ardışıl iki çerçeve arasındaki konum değişimi nesnenin boyutundan daha küçük ise, nesne farklı iki nesne gibi algılanabilir. Bu algılanan ayrık nesnelere, aslında bir bütünün parçaları olabilir.

1.2.1.4.3. Ortalama Arkaplan Modeli

Ortalama yönteminde belirli bir öğrenme süresince ardışık çerçevelerin piksel değerlerinin ortalaması alınarak arkaplan modeli oluşturulur. Nesne oluşturulan bu modelle, gelen çerçevenin farkı alınarak bulunabilmektedir. Ayrıca öğrenme dizisi boyunca her bir pikselin standart sapması hesaplanarak arkaplan modeli ile karşılaştırmada eşik değeri olarak kullanılabilir.

$$B_t(x, y) = \frac{1}{N} \sum_{k=1}^N I_k(x, y) \quad (1.10)$$

1.2.1.4.4. Adaptif Arkaplan Modeli

Adaptif arkaplan yönteminde arkaplan modeli geribeslemeli olarak hesaplanır. Her yeni gelen çerçevede arkaplan modeli güncellenir ve güncellenen bu modelde kendinden önceki tüm çerçevelerin etkisi vardır.

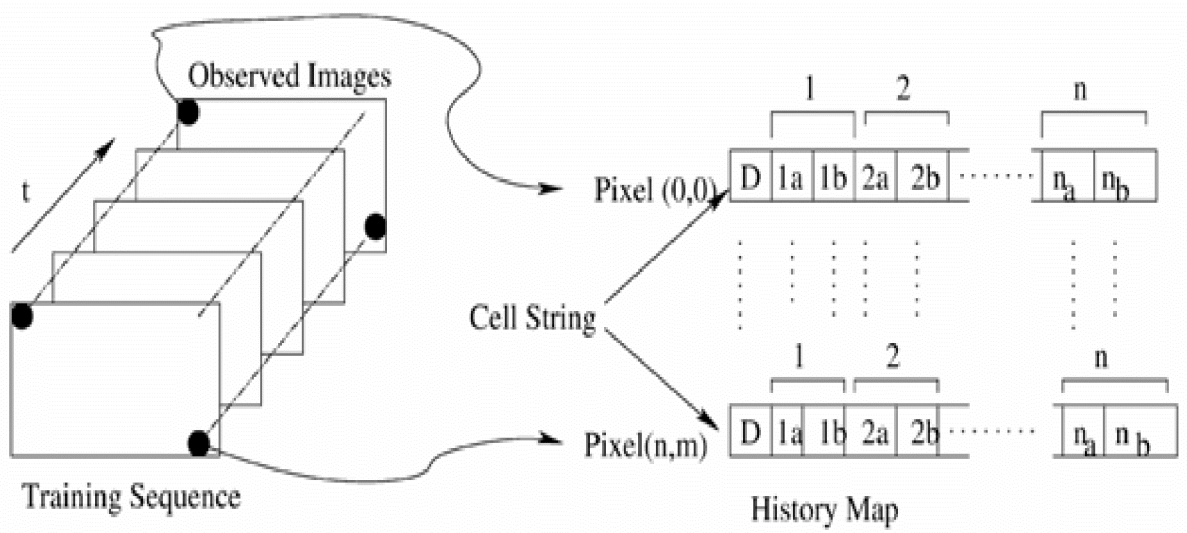
$$B_{t+1}(x, y) = \alpha \cdot B_t(x, y) + (1 - \alpha) \cdot I_t(x, y), 0 < \alpha < 1 \quad (1.11)$$

α , öğrenme katsayıdır. Model; değişen aydınlatma koşullarına, sahneye yeni gelen ve sahnedeki çıkan sabit nesnelere uyum sağlayabilmektedir. α değeri birine yaklaştıkça öğrenme yavaşlar, sıfıra yaklaştıkça ise öğrenme hızlanır. α değerinin sıfır alınması ardışıl çerçeve farkı ile aynı işlevi görür. Bu yöntemde nesnelere arkalarında izler bırakması beklenen bir sonuçtur.

1.2.1.4.5. En Çok Tekrarlılık Oranı

En Çok Tekrarlılık Oranı algoritması(ETO), arkaplan modelleme sırasında bir pikseldeki en çok beliren parlaklık değerini arkaplan modeli olarak seçer [18]. Bu yöntemde nesne öğrenme süresince yeteri kadar hızlı ise belirgin bir arkaplan modeli oluşturulur. Arkaplan öğrenme süresi $2t$ ise, nesnenin t den daha az bir zaman aynı yerde durması arkaplan öğrenmeye problem teşkil etmez.

Bir eğitim süreci boyunca gelen görüntülerdeki pikseller Şekil 1.4. 'teki gibi modellenir.



Şekil 1.4. ETO algoritmasında piksel değerlerinin modellenmesi [18].

1.2.1.4.6. Temel Bileşen Analizi Yardımıyla Arkaplan Modelinin Belirlenmesi

1.2.1.4.6.1. Temel Bileşen Analizi

Temel Bileşen Analizi (PCA), asal bileşen analizi ya da Karhunen-Loeve dönüşümü olarak da isimlendirilen, örüntü tanıma çalışmalarında yaygın kullanım alanı bulunan bir altuzay izdüşüm yöntemidir. PCA, değişkenlerden oluşan bir veri kümesinin varyans-kovaryans yapısını, bu değişkenlerin doğrusal birleşimleri yardımıyla açıklayarak, boyut indirgemesi ve yorumlamasını sağlayan çok değişkenli bir istatistik yöntemidir [29].

1901 yılında Karl Pearson ‘un başlattığı temel bileşenler analizi çalışmaları, 1993 yılında Hotelling tarafından geliştirilmiştir [30]. PCA çok sayıda birbiri ile ilişkili değişkenler içeren veri setinin boyutlarını veri içerisinde var olan değişimlerin mümkün olduğunca korunarak daha az boyuta indirgenmesini sağlayan bir dönüşüm tekniğidir [30]. Analiz eldeki veriyi daha sayıda değişkenle ifade edebilecek en iyi dönüşümü belirlemeyi amaçlar. Dönüşüm sonrasında elde edilen değişkenler ilk değişkenlerin temel bileşenleri olarak adlandırılır. İlk temel bileşen varyans değeri en büyük olandır ve diğer temel bileşenler varyans değerleri azalacak şekilde sıralanır. Temel bileşenler birbirine diktir. Gürültüye karşı düşük hassasiyet, bellek ve kapasite ihtiyaçlarının azalması, az boyutlu uzaylarda etkin çalışması PCA ‘nın avantajları arasında sıralanabilir [31].

1.2.1.4.6.2. Temel Bileşen Analizi Metodolojisi

Eğitim kümesi veri matrisinin (X) M tane Nx1 ‘lik gözlem vektöründen oluşan bir eğitim matrisi olduğu varsayalım. Burada her bir gözlem vektörü, görüntü matrisinin sütun vektörüne dönüştürülmesiyle oluşturulur. PCA yöntemi aşağıdaki şekilde uygulanır:

Eğitim veri matrisi (1.2) deki gibi gösterilir.

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^M \\ x_2^1 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^M \end{bmatrix} \quad (1.12)$$

Çok değişkenli analizde, çoğu zaman birimleri farklı olan değişkenlerle ilgilenilir. Ancak bazı durumlarda verilerin aynı birimde olması daha iyi sonuç verir. Bu amaçla, değişken değerleri merkezileştirilerek aynı birime dönüştürülür. Bu standartlaştırma veri ortalaması 0 ‘a çekilerek yapılır. Eğitim vektörlerinin ortalaması m, bağıntı (1.13) ‘deki gibi hesaplanır. Hesaplanan ortalama vektörü her bir gözlem vektöründen çıkartıldığında değişkenler 0 ortalamalı olur ve ortalaması çıkarılmış veri matrisi \tilde{X} ile gösterilir.

$$m = \frac{1}{M} \sum_{i=1}^M x^i = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{bmatrix} \quad (1.13)$$

Ortalaması çıkarılmış gözlem vektörleri (1.14) ve (1.15) bağıntılarında gösterilmiştir.

$$\tilde{x}^i = x^i - m, \forall i \quad (1.14)$$

$$\tilde{X} = [\tilde{x}^1 \quad \tilde{x}^2 \quad \cdots \quad \tilde{x}^M] \quad (N \times M) \quad (1.15)$$

0 ortalamalı eğitim matrisi bağıntı (1.16) ile bulunur.

$$\tilde{X} = \begin{bmatrix} x_1^1 - m_1 & x_1^2 - m_1 & \cdots & x_1^M - m_1 \\ x_2^1 - m_2 & x_2^2 - m_2 & \cdots & x_2^M - m_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 - m_N & x_N^2 - m_N & \cdots & x_N^M - m_N \end{bmatrix} \quad (1.16)$$

Daha sonra sıfır ortalamalı eğitim vektörünün kovaryans matrisi (1.17) kullanılarak hesaplanır ve (1.18) elde edilir.

$$C = \tilde{X} \tilde{X}^T = \frac{1}{M-1} \sum_{i=1}^M (x^i - m)(x^i - m)^T \quad (1.17)$$

$$\begin{bmatrix} (x_1^1 - m_1)^2 & (x_1^2 - m_1)(x_2^1 - m_2) & \cdots & (x_1^M - m_1)(x_N^1 - m_N) \\ (x_2^1 - m_2)(x_1^2 - m_1) & (x_2^2 - m_2)^2 & \cdots & (x_2^M - m_2)(x_N^2 - m_N) \\ \vdots & \vdots & \vdots & \vdots \\ (x_N^1 - m_N)(x_1^M - m_1) & (x_N^2 - m_N)(x_2^M - m_2) & \cdots & (x_N^M - m_N)^2 \end{bmatrix} \quad (1.18)$$

Burada köşegen üzerinde bulunan katsayılar varyans, diğerleri kovaryans katsayılarıdır. Varyans diğer boyutlardan bağımsız olarak yalnızca bir boyuttaki verinin ortalama değere göre dağılımı hakkında bilgi vermektedir. Kovaryans bir boyutun diğer boyutlara bağlı olarak nasıl bir değişim gösterdiği hakkında bilgi verir. Daima 2 boyut arasında hesaplanır [32]. Değişkenlerden birinin değeri artarken diğerinin değeri artıyor ya

da biri azalırken diğeri de azalıyorsa iki deęişken arasındaki kovaryans deęeri pozitif olacaktır. Deęişkenlerden birinin deęeri artarken diğerin azalıyorsa kovaryans deęeri negatif çıkacaktır. Eđer deęişkenler arasında belirgin bir ilişki yoksa kovaryans sifıra yakın bir deęer olur [33].

(1.19) eşitliğini kullanarak kovaryans matrisinin öz deęer ve öz vektörleri hesaplanır. C , $N \times N$ boyutlu bir matris, λ herhangi bir skaler ve v sıfırdan farklı bir sütun vektör olmak üzere,

$$Cv = \lambda v \quad (1.19)$$

Eşitliğini sağlayan λ sayısı C 'nin öz deęeri ve v 'de λ ile ilişkili öz vektörüdür. Öz deęerler büyükten küçüğe doğru sıralanır. Bu sıralı öz deęerlere karşılık gelen sıralı vektörlerden ilk P tanesi kullanılarak G izdüşüm matrisi oluşturulur. Seçilen öz vektörler matrisin satırlarını oluşturacak şekilde (1.20) 'daki gibi en büyük öz deęerden başlayarak sıralanır.

$$G^T = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_p \end{bmatrix} \quad (1.20)$$

(1.21) 'deki gibi öznitelik vektörü belirlenmek istenen vektörlerin G matrisi üzerinde izdüşümü alınır.

$$y^i = G^T x^i, i = 1, 2, \dots, M \quad (1.21)$$

1.2.1.4.6.3. Temel Bileşen Analizinin Arkaplan Modellemeye Uygulanması

Arkaplan modelinin Temel Bileşen Analizi yardımıyla belirlenmesi ilk olarak Oliver ortaya koymuştur [19]. Oliver tekil piksellerin deęişimlerini modellemek yerine, bütünsel bir yaklaşım olarak özuzay ayrışımı yapmıştır [20].

Arkaplan modeli belirlenirken, kameradan elde edilen görüntü çerçeveleri(X) boyutu $w \times h$ olan 2 boyutlu bir matristir. Arkaplan modelinin öğrenilmesi için geçmiş N adet çerçevenin de hesaba katılması elimizde 3 boyutlu bir veri kümesi oluşturur. Özüay ayırımı yapabilmek için bu veri kümesini 2 boyutlu arkaplan matrisine (X) aşağıdaki şekle uygun olarak dönüştürülmesi gerekir. Oluşturulan 2 boyutlu arkaplan matrisinin(X) herbir satırı bir resme karşılık gelir.

Kovaryans matrisi X ve X^T matrislerinden (1.17) bağıntısındaki gibi hesaplanır. Öğrenme için kullanılan geçmiş çerçeve sayısı genellikle resmin boyutundan küçük olduğu için kovaryans matrisi (1.17) bağıntısıyla oluşturulur. Bu şekilde yapılan hesaplama ile kovaryans matrisinin boyutu $N \times N$ olmaktadır. Eğer kovaryans matrisi $C = \tilde{X}^T \tilde{X}$ şeklinde hesaplanırsa, kovaryans matrisinin boyutu $(w \times h) \times (w \times h)$ olacaktı. Bu da, 320×240 boyutunda 24 bitlik bit bitleşim resmi için 395 GB boyutunda kovaryans matrisinin tutulmasını gerektirir. Kovaryans matrisinin $C = \tilde{X}^T \tilde{X}$ şeklinde hesaplanması durumunda geçmiş 50 çerçevelik bir arkaplan matrisi yaklaşık 20 KB'lık bir alan kaplar. Temel bileşen analizini bu şekilde hesaplamaya enstantane (snapshot) yoluyla hesaplama denir [34]. Kovaryans matrisine karşılık gelen özdeğerler (λ_i) ve özvektörler (V_i) (1.19) bağıntısındaki eşitlik çözümlerle elde edilebilir [20].

Hesaplanan özvektörler özdeğerlerin büyüklüklerine göre sıralanıp içlerinden anlamlı olan m adet özvektör seçildiğinde boyutu $m \times N$ olan V_M matrisi elde edilir. Özvektörler V_M matrisinin satırları olarak tutulur.

$$V_M = [V_1 \ \cdots \ V_m], 0 < m < N \quad (1.22)$$

Verilen örneklerin özüzaya izdüşürülmesi için izdüşüm matrisi P aşağıdaki şekilde hesaplanır.

$$P = X^T V_M^T \quad (1.23)$$

Her yeni gelen çerçevedeki (I) hareketli nesnelerin algılanması için arkaplan modeli B aşağıdaki eşitlikle belirlenir.

$$B = P(P^T(I - \mu)) + \mu \quad (1.24)$$

Hareketli cisimlerin bulunduğu resim, arkaplan modeli ve yeni gelen çerçevenin mutlak farkı alınarak hesaplanır.

$$D = |B - I| \quad (1.25)$$

1.2.1.4.7. Gaussian Karışım Modeli Kullanarak Arkaplan Modelini Belirleme

Arkaplan çıkarma yöntemi Wren'in 1997 yılında yaptığı çalışma ile daha popüler hale gelmiştir [21]. Bir pikselin renginin geçmiş çerçevelerdeki aşamalı değişimini modellemek için Wren vd. her bir pikselin rengini 3 boyutlu tekil bir Gaussian ile modellemeyi önermişlerdir. Modelin parametreleri olan ortalama $\mu_{(x,y)}$ ve kovaryans $\Sigma_{(x,y)}$ geçmişteki görüntü çerçevelerindeki değerler kullanılarak hesaplanır. (x,y) pikseli için oluşturulan model yeni gelen piksel değerlerinin olabirliği ile aşağıdaki bağıntıdaki gibi ifade edilir.

$$N(\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (1.26)$$

Bağıntıdaki D boyut sayısını, x verinin kendisini ifade eder.

Yeni gelen piksel değerinin olasılığı belirli bir eşik seviyesini aşmıyorsa bu piksel önplan yani hareketli bir nesne, aksi durumda ise arkaplan olarak etiketlenir. Gao vd.'nin çalışması göstermiştir ki bir pikseldeki renk dağılımını tek bir Gaussian ile modellemek mümkün değildir [22]. Pikselin parlaklık değeri dağılımını tek bir Gaussian ile ifade edememenin nedeni tekrarlı hareket gösteren nesnelere, gölge ve yansımalar olabilir. Stauffer ve Grimson piksel değerlerini tek bir Gaussian ile modellemek yerine birden fazla normal dağılımın birleşimi olarak modelleyen bir yöntem öne sürmüşlerdir [23]. KadewTraKuPong bu yönteme gölge elimine etmeyi de uygulayarak algoritmanın performansını artırmıştır [24].

Görüntü içerisindeki her pikselin parlaklık değeri K tane Gaussian dağılımının karışımı olarak modellenir. N anındaki parlaklık değeri x_N olan bir pikselin olasılığı aşağıdaki gibi yazılabilir.

$$p(x_N) = \sum_{j=1}^K w_j \eta(x_N; \theta_j) \quad (1.27)$$

ω_k , k. Gaussian bileşeninin ağırlık parametresidir. k. bileşenin normal dağılımı olan $\eta(x; \theta_k)$ aşağıdaki bağıntıyla ifade edilebilir.

$$\eta(x; \theta_k) = \eta(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)} \quad (1.28)$$

μ_k , k. bileşenin ortalaması, $\Sigma_k = \sigma_k^2 I$ ise k. bileşenin kovaryansdır.

K farklı dağılım uygunluk derecesi olan w_k/σ_K değerine göre sıralanır ve B adet dağılım arkaplan modelini belirleyen olarak seçilir. B aşağıdaki gibi hesaplanır.

$$B = \arg \min_b \left(\sum_{j=1}^b w_j > T \right) \quad (1.29)$$

T eşik değeri arkaplan modelindeki Gaussian bileşenleri için en küçük ağırlık değerini belirler. Eşik değerini aşanlar arkaplan dağılımları olarak seçilir.

Arkaplan çıkarma işlemi, yeni gelen piksel değerinin seçilen B dağılıma olan uzaklığının 2,5 standart sapması kadar uzak olmasına göre yapılır. Başka bir deyişle yeni gelen piksel değeri B dağılıma dahil olamıyorsa hareketli kısım olarak işaretlenir. Aksi durumda yeni gelen piksel değerinin dahil edildiği Gaussian bileşeni aşağıdaki bağıntılarla güncellenir.

$$\hat{w}_k^{N+1} = (1 - \alpha) \hat{w}_k^N + \alpha \hat{p}(\omega_k | x_{N+1}) \quad (1.30)$$

$$\hat{\mu}_k^{N+1} = (1 - \alpha) \hat{\mu}_k^N + \rho x_{N+1} \quad (1.31)$$

$$\hat{\Sigma}_k^{N+1} = (1 - \alpha) \hat{\Sigma}_k^N + \rho (x_{N+1} - \hat{\mu}_k^{N+1})(x_{N+1} - \hat{\mu}_k^{N+1})^T \quad (1.32)$$

$$\rho = \alpha \eta(x_{N+1}; \hat{\mu}_k^N, \hat{\Sigma}_k^N) \quad (1.33)$$

$$\hat{p}(\omega_k | x_{N+1}) = \begin{cases} 1 & \omega_k \text{ ilk karşılaşılan Gaussian bileşeni ise} \\ 0 & \text{aksi hallerde} \end{cases} \quad (1.34)$$

ω_k , k. Gaussian bileşenidir. $1/\alpha$ değeri değişimi belirleyen zaman sabitidir.

Eğer piksel değeri K adet normal dağılımla eşleştirilemezse, en düşük olasılıklı dağılım yeni bir dağılımla yer değiştirilir. Bu yeni dağılımın ortalaması eşleşmeyen piksel değeri, varyansı yüksek ve ağırlığı düşük bir değer olarak belirlenir.

Normal dağılımların sayısı olan K başlangıçta belirlenir ve süreç boyunca sabittir. Zivkovic K değerinin dinamik olarak süreç esnasında belirlenmesine yönelik bir çalışma yapmıştır [25].

1.2.1.4.8. Ağırlık Merkezi Arkaplan Modeli

Ağırlık merkezi arkaplan modelinde görüntü $m \times n$ bloğa bölünür. Arkaplan modelleme işlemi tekil pikseller yerine bu bloklar üzerinde yapılır. Aşağıdaki bağıntılarla blokların ağırlık merkezinin koordinatları bulunur.

$$x_c = \frac{\sum_x \sum_y x f(x,y)}{\sum_x \sum_y f(x,y)} \quad (1.35)$$

$$y_c = \frac{\sum_x \sum_y y f(x,y)}{\sum_x \sum_y f(x,y)} \quad (1.36)$$

$f(x,y)$, görüntünün (x,y) konumundaki pikselinin parlaklık değeridir.

Her blok için belirlenen ağırlık merkezi koordinatları K adet çerçeve için hesaplanır ve bellekte saklanır. K adet koordinatın ortalaması ve standart sapması aşağıdaki bağıntılarla hesaplanır.

$$\mu_{x_c}(m, n) = \frac{1}{K} \sum_{k=0}^{K-1} x_{c_k}(m, n) \quad (1.37)$$

$$\sigma_{x_c}(m, n) = \sqrt{\frac{1}{K} \sum_{k=0}^{K-1} [x_{c_k}(m, n) - \mu_{x_c}(m, n)]^2} \quad (1.38)$$

$\mu_{x_c}(m, n)$, görüntünün m . satır ve n . sütundaki bloğu için hesaplanan ağırlık merkezlerinin x koordinatlarındaki değerlerin ortalamasıdır. $\sigma_{x_c}(m, n)$ ise aynı koordinat değerleri için standart sapma değeridir. Ortalama ve standart sapma değerleri y koordinatlarındaki değerler için de benzer şekilde hesaplanır. Elde edilen ortalama ve standart sapma değerlerine göre yeni gelen çerçevedeki bir blokta hareketli kısımlar olup olmadığı aşağıdaki eşitlikle belirlenir.

$$D(m, n) = \begin{cases} 1 & |x_c(m, n) - \mu_{x_c}(m, n)| > \lambda \sigma_{x_c}(m, n) \\ 1 & |y_c(m, n) - \mu_{y_c}(m, n)| > \lambda \sigma_{y_c}(m, n) \\ 0 & \text{aksi halde} \end{cases} \quad (1.39)$$

λ değeri dağılım içerisinde standart sapma değerinin kaç katının kullanılacağını belirtir. Örneğin; λ değerini 6 almak normal dağılımın özelliğinden dolayı verinin %99.9996'sını seçime dahil eder [17].

1.2.2. Nesne Takibi

Nesne Takibi, bilgisayarla görme alanında önemli bir görevdir. Yüksek güçteki bilgisayarların artması, yüksek kaliteli ve ucuz görüntü yakalama aygıtları ve kameraların varlığı ve insansız olarak çalışabilen otomatik denetleme sistemlerine olan ihtiyacın artması Nesne Takibi algoritmalarına olan ilgiyi artırmıştır. Nesne Takibi işlemi genel olarak üç anahtar aşamadan oluşur: görüntüdeki hareketli nesnelere algılamak, algılanan bu hareketli nesnelere çerçeveden çerçeveye izlenmesi ve bu izleme verilerini kullanarak nesnelere davranışlarını tanımak.

Nesne takibi aşağıdakiler gibi birçok bilgisayarla görme probleminde anahtar aşamadır:

- Harekete Dayalı Tanıma: Yürüyüş şekline göre kişi tanıma gibi
- Otomatik Gözetleme: Bir görüntü sahnesini sürekli izleyerek şüpheli durumları algılama.
- Video İndisleme: Videoları izleyerek otomatik olarak bilgi çıkarma ve etiketleme
- Bilgisayar-İnsan Etkileşimi: El hareketleri veya göz bebeği hareketlerini algılayarak bilgisayara komutlar verme

- Trafik Gözetleme: Karayollarındaki araçların yol içerisindeki akışlarına göre trafik yoğunluğu, ortalama hız ve sıkışıklık olan bölgeler gibi bilgilerin çıkarılması
- Araç Seyri Belirleme: İnsansız araçlarda yol içerisindeki hareketli ve hareketsiz nesnelerin takibi ile araca bir yol çizme

Nesne takibi, bir nesnenin sahneye girdiği anda sahneden çıktığı ana kadar görüntü içindeki konumunun belirlenmesidir. Nesne takip algoritması ayrıca nesnenin görüntüde her bir çerçevede kapladığı alan bilgisini de verebilir. Görüntü içerisindeki nesnelere algılamak ve bu nesnelerin daha önceki çerçevelerdeki nesnelerle ilişkilendirilme işlemi, nesne takibi algoritmasının içinde yapılabileceği gibi ayrı olarak da yapılabilir.

1.2.2.1. Nesne Temsili

Bir takip senaryosunda, takip edilen nesne ilgilenilen herhangi bir şey olabilir. Örneğin; denizdeki bir kayık, akvaryum içindeki bir balık, yoldaki bir araç, bir meydandaki yayalar, havada uçmakta olan bir uçak, su içerisindeki hava kabarcıkları takip için ilgilenilen nesnelere olabilir. Nesnelere takip için farklı şekillerde temsil edilebilir. Nesnelere genel olarak, basit geometrik şekiller, nesne silüeti ve sınırları, eklemli şekil modelleri ve iskelet modelleri ile temsil edilir.

1.2.2.1.1. Noktalar

Nesnelere cismin merkezindeki bir nokta veya bir nokta kümesi ile ifade edilebilir. Görüntü içerisinde küçük alan kaplayan nesnelere için noktalar ile temsil daha uygun olmaktadır.

1.2.2.1.2. Basit Geometrik Şekiller

Nesnenin şekli dikdörtgen, elips, kare, üçgen, vb. geometrik şekillerle ifade edilebilir. Bu şekildeki temsilde nesnenin hareketi genellikle öteleme, afin ve izdüşümsel dönüşüm olarak modellenir. Katı cisimler (eklemleri olmayan cisimler) için bu tipteki temsil şekli daha uygundur. Katı olmayan cisimler içinde geometrik şekillerle temsil kullanılmaktadır.

1.2.2.1.3. Nesne Silueti ve Sınırları

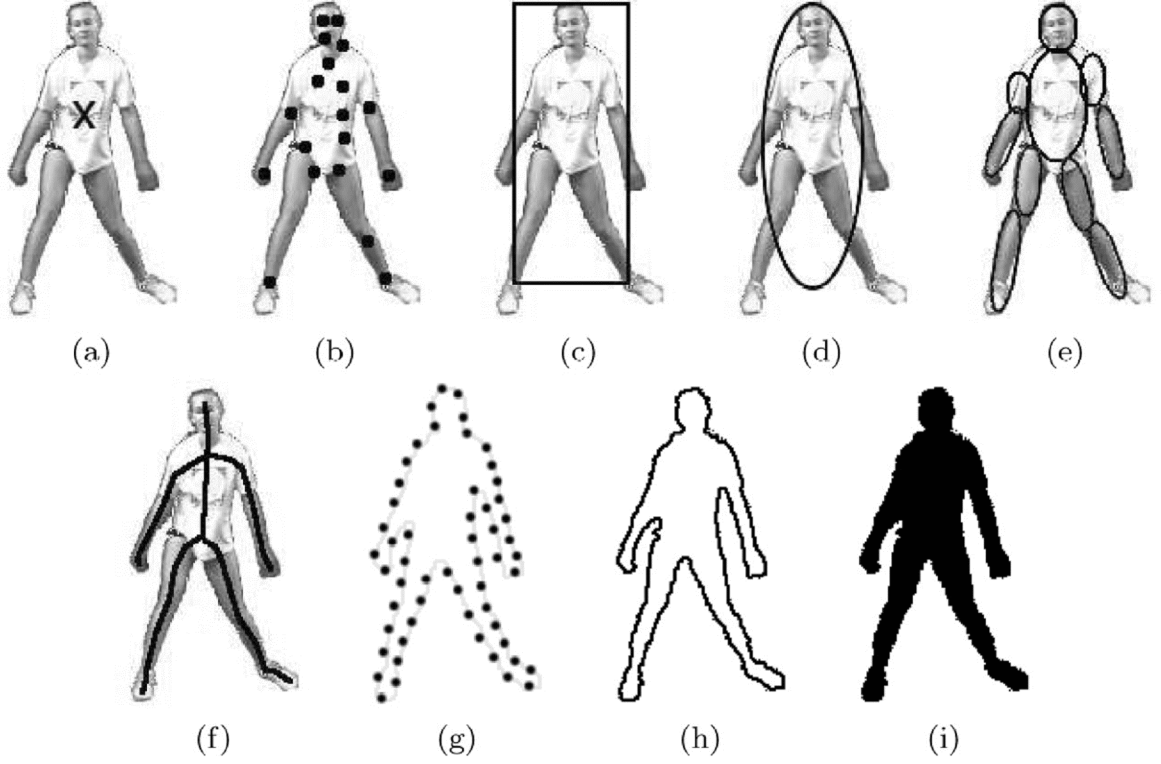
Sınır temsili bir nesnenin sınırlarını belirtir. Sınır içerisinde kalan bölge ise nesnenin silueti olarak adlandırılır. Siluet ve sınır temsili katı olmayan cisimlerin temsili için daha uygundur.

1.2.2.1.4. Eklemlı Şekil Modelleri

Eklemlı cisimler gövdeye tutunan ve eklemlerle birbirine bağlanan parçalardan oluşur. Örneğın; nesne takibi için insan gövde, bacaklar, kollar, kafa, eller ve ayaklardan oluşun eklemlı bir nesnedir. Parçalar arasındaki ilişki eklem açısı gibi kinematik hareket modelleri yönetilir. Eklemlı bir cismi temsil etmek için nesne parçaları silindir ve elips gibi geometrik şekillerle tanımlanabilir.

1.2.2.1.5. İskelet Modelleri

Nesne iskeleti, nesnenin silüetine orta eksen dönüşümü uygulanarak elde edilebilir. Bu model genellikle nesneleri tanımak için kullanılır. İskelet temsili hem eklemlı, hem de katı cisimler için kullanılabilir.



Şekil 1.5. Nesne temsilleri (a) Merkez noktası, (b) Çoklu noktalar, (c) Dikdörtgen bölge, (d) Eliptik bölge, (e) Eklem modeli, (f) İskelet Modeli, (g-h) Sınır temsili, (i) Silüet Temsili [26]

1.2.2.2. Nesne Takibi İçin Özellik Seçimi

Nesne takibinde doğru özelliklerin seçimi kritik bir rol oynar. Bir görsel özellikten beklenen en önemli nitelik, bu özelliğin nesnelere ayırt ederken eşsiz olabilmesidir. Özellik seçimi nesnenin temsili ile yakından ilişkilidir. Örneğin; nesnelere sınırlarla temsil ediliyorsa, genellikle özellik olarak nesnelere kenar bilgileri seçilir.

1.2.2.2.1. Renk Özellikleri

Görüntüdeki cisimlerin renkleri temel olarak fiziksel faktörlerden etkilenir.

- Görüntüdeki ışığın spektral güç dağılımı,
- Cisim yüzeyinin yansımaya özellikleri.

Görüntü işlemede nesneyi temsil etmek için genellikle RGB (Kırmızı – Yeşil – Mavi) uzayı kullanılır. Fakat RGB uzayı algısal olarak tek biçimli değildir. RGB uzayındaki renkler arasındaki farklar, insanların renkleri algılamadaki farklılıklara karşılık

gelmez. Bunun yanında RGB uzayının boyutları birbiri ile aşırı derecede ilişkilidir. Bunun tersi olarak $LxUxV$ ve $LxAxB$ renk uzayları algısal olarak tek biçimlidir. HSU uzayında yaklaşık olarak tek biçimlidir. Fakat bu renk uzayları gürültüye karşı çok hassastır.

Özet olarak özelliklerin hangi renk uzayından seçileceği konusunda kesin bir yargı yoktur. Nesne takibi için çok çeşitli renk uzayları kullanılmaktadır.

1.2.2.2.2. Kenar Özellikleri

Nesnelerin sınır bölgeleri görüntüdeki parlaklık değerleri arasında güçlü değişimler gösterir. Kenar algılama teknikleri bu değişimleri algırlar. Kenarların önemli bir özelliği renk özellikleriyle karşılaştırıldıklarında ışıklandırma değişimlerine daha az duyarlı olmasıdır.

Nesnenin sınırlarını takip eden algoritmalar temsil edici olarak genellikle kenar özelliklerini kullanırlar. Basitliği ve etkinliği nedeniyle en popüler olan kenar algılama algoritması Canny kenar algılayıcısıdır [28].

1.2.2.2.3. Optik Akış Özellikleri

Optik akış, bir bölgedeki piksellerin ötelemelerinin oluşturduğu yer değişim vektörleridir. Optik akış parlaklık sabitini kullanarak hesaplanır. Hareketin olmadığı bölgelerde parlaklık değerinin değişmez olduğu varsayılır. Optik akış genellikle hareket tabanlı nesne algılama ve takip yöntemlerinde özellik olarak kullanılır.

1.2.2.2.4. Doku Özellikleri

Resimdeki doku, bir yüzeyin yumuşaklık ve düzenlilik gibi parametrelerini dikkate alan bir parlaklık değişimi ölçütüdür. Renk ile karşılaştırıldığında doku tanımlayıcı özelliklerin elde edilmesi için ek bir işlem gerektirir. Gri seviye birliktelik matrisi ve dalgacıklar doku özelliklerine örnek verilebilir. Kenar özelliklerinde olduğu gibi, doku özelliklerinde de renkle karşılaştırıldığında ışıklandırma değişimlerine daha az duyarlıdır.

1.2.2.3. Ortalama Kayma Algoritması

Bu algoritma, takip edilecek nesnenin elde edilen özelliklerden (renk, doku vs.) yararlanarak hedefin kısmi olarak kaybolması durumunda bile iyi bir sonuç üretmektedir. Bu algoritmanın amacı, takip edilecek olan nesnenin özelliklerine bakarak, nesnenin yerini bir sonraki görüntü çerçevesinde bulmaktır [10].

Takip edilecek nesnenin m seviyeden (0-255) oluşan normalize edilmiş renk histogramı, bağıntı (1.40) 'teki gibi \hat{q} değişkeni ile gösterilmiştir.

$$\hat{q} = \{\hat{q}_u\}, u = 1, \dots, m \quad (1.40)$$

$$\sum_{u=1}^m \hat{q}_u = 1 \quad (1.41)$$

Hedef görüntüye ait m seviyeden oluşan y noktasındaki renk histogramı, bağıntı (1.42) 'daki gibi gösterilebilir.

$$\hat{p}(y) = \{\hat{p}_u(y)\}, u = 1, \dots, m \quad (1.42)$$

$$\sum_{u=1}^m \hat{p}_u = 1 \quad (1.43)$$

Bu iki histogram bağıntısı arasındaki benzerliği ölçmek için kullanılacak ölçütlerden biri Bhattacharyya katsayısı olabilir. Bu katsayı (1.44) 'teki gibi ifade edilir.

$$\hat{p}(y) \equiv p[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y)\hat{q}_u} \quad (1.44)$$

Normalize edilmiş histogramlar için Bhattacharyya katsayısı 1 değerine ne kadar yakınsa iki görüntü arasındaki benzerlik o kadar fazla olmaktadır.

Ortalama kayma algoritmasının adımları şu şekildedir;

1. Şimdiki görüntüde, y_0 noktasında bulunan nesne için $\hat{p}(y_0)$ değeri hesaplanarak Bhattacharyya katsayısı $\hat{p}(y_0)$ bulunur.

2. Her bir nokta için histograma dayalı piksel ağırlıklarını ifade eden ω_i değerleri (1.45) 'daki bağıntıya göre hesaplanır.

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}}{\hat{p}_u(\hat{y}_0)}} \quad (1.45)$$

3. Bir sonraki görüntüde nesnenin yeni konumu olan y_1 noktası (1.46) 'taki bağıntıyla belirlenmeye çalışılır. X_i mevcut görüntüdeki normalize edilmiş piksellerin konumlarını ifade etmektedir.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i \omega_i}{\sum_{i=1}^{n_h} \omega_i} \quad (1.46)$$

4. Eğer, $|y_1 - y_0| < \text{eşik değeri}$ koşulu sağlanmış ise yeni konum y_1 olarak kabul edilir. Fark değerinin eşik değerinden büyük olması durumunda $y_0 = y_1$ yapılarak 2. adımdan itibaren işlemler tekrarlanır. Burada Bhattacharyya katsayısı kontrol değeri olarak kullanılabilir.

1.2.2.4. Rastlantısal Kestirim

Ölçüm işlemlerinin bir kümesinden oluşan bilinmeyen bir durumun kestirimi için yaklaşımlar olmasına rağmen, bu yöntemlerin çoğu ölçümlerin gürültülü doğasını göz önünde bulundurmazlar. Örneğin; radar sistemlerinde sensörlerden gelen ölçüm bilgisi gürültü içermektedir. Bu gürültü, istatistiksel olarak doğada bulunmakta ve problemin belirlenmesi için bizi rastlantısal kestirim yöntemini kullanmaya yöneltmektedir.

1.2.2.4.1. Durum-Uzayı Modelleri

Durum-uzaylı modelleri, aslında kestirim ve kontrol problemlerinin çözümlenebilir şekle getirilmesi için uygun gösterim şeklidir. Farz edelim ki dinamik bir işlem n . dereceden diferansiyel denklem (1.47) bağıntısındaki gibi tanımlansın:

$$y_{i+1} = a_{0,i}y_i + \cdots + a_{n-1,i}y_{i-n} + 1 + u_i, i \geq 0 \quad (1.47)$$

u_i istatistiksel sıfır ortalamalı, spektral olarak beyaz rastgele gürültü değişkeni, aşağıda verilen öz ilişki formülü ile belirtmektedir.

$$E(u_i, u_j) = R_u = Q_i \delta_{ij} \quad (1.48)$$

$y_0, y_{-1}, \dots, y_{-n+1}$ başlangıç değerleri bilinen $n \times n$ 'lik kovaryans matrisine sahip sıfır ortalamalı rastgele değişkenlerdir.

Gürültü istatistiksel olarak kestirimi yapılacak olan işlemde bağımsızdır. Bazı diğer temel durumların olması halinde [35] bu diferansiyel eşitlik (1.49) bağıntısındaki gibi yeniden yazılabilir:

$$\hat{x}_{i+1} \equiv \begin{bmatrix} y_{i+1} \\ y_i \\ y_{i-1} \\ \vdots \\ y_{i-n+2} \end{bmatrix} = \underbrace{\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} y_i \\ y_{i-1} \\ y_{i-2} \\ \vdots \\ y_{i-n+1} \end{bmatrix}}_{\hat{x}_i} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_G \quad (1.49)$$

bu ifade durum-uzay modelinin (1.50) ve (1.51) bağıntılarındaki gibi ifade edilmesini sağlar.

$$\hat{x}_{i+1} = A\hat{x}_i + Gu_i \quad (1.50)$$

$$\hat{y}_i = [1 \ 0 \ \cdots \ 0] \hat{x}_i \quad (1.51)$$

ya da daha genel biçimiyle;

$$\hat{x}_{i+1} = A\hat{x}_i + Gu_i \quad (1.52)$$

$$\hat{y}_i = H_i \hat{x}_i \quad (1.53)$$

Eşitlik (1.52), bir önceki durum x_i ve bazı işlem gürültüsü u_i değişkenlerinin doğrusal kombinasyonu şeklinde modellenerek, yeni durum x_{i+1} 'in gösteriminin yoludur. Eşitlik (1.53), x_i dahili durumlarından, süreç ölçümlerinin ya da y_i gözlemlerinin türetildiğinin tanımlamasının yoludur. Bu iki eşitlik genelde sırasıyla işlem modeli ve ölçüm modeline karşılık gelir ve bütün doğrusal kestirim yöntemleri için temel teşkil eder.

1.2.2.4.2. Gözlemci Tasarım Problemi

Doğrusal sistem teorisi alanında rastlantısal kestirim ile ilgili genel bir problem bulunmaktadır. Bu problem literatürde Gözlemci Tasarım Problemi olarak geçmektedir. Burada temel problem, sadece sistem çıktılarına erişimi verilmiş doğrusal bir sistemin dahili durumlarının kestirilmesidir (Sistem kontrol girdilerine erişim olduğu farz edilir ama burada bu göz ardı edilir). Bu problem insanların, genelde kara kutu probleminde olduğu gibi, kara kutudan gelen bazı sinyallere erişildiğine fakat doğrudan içinde ne olduğunun gözlemlenememesini düşünmeleri ile benzerdir.

Bu temel probleme getirilen yaklaşımlar benzer olarak durum-uzay modeline dayanmaktadır. Burada sistem durumunun dönüşümünü biçimlendiren bir sistem modeli vardır. Bu model (1.54) bağıntısında olduğu gibi doğrusal rastlantısal diferansiyel şeklinde ifade edilmektedir:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (1.54)$$

Buna ek olarak sistem durumu ve ölçümler arasındaki ilişkiyi tanımlayan ölçüm modeli vardır. Bu model genellikle (1.55) 'de bulunan doğrusal ifade ile temsil edilmektedir.

$$z_k = Hx_k + v_k \quad (1.55)$$

w_k ve v_k terimleri sırasıyla rastgele sistem ve ölçüm gürültü değişkenlerini temsil etmektedir. Dikkat edilirse (1.53) bağıntısında bağımlı değişken olan y_k (1.55) bağıntısında z_k ile değiştirilmiştir. Bunun mantığı, ölçümlerin belirli bir şekilde durumun elemanları olması gerekmediğini ancak durum elemanlarının herhangi bir doğrusal kombinasyonu şeklinde olabileceğini göstermektir.

1.2.2.4.2.1. Ölçüm ve Sistem Gürültüsü

Birçok ölçümlerde, birçok farklı gürültü kaynağı bulunmaktadır. Burada gürültülü sensör ölçümlerinin genel durumu düşünülebilir. Örneğin, her bir sensör türünün, bulunduğu fiziksel ortamla ilişkili olarak, kendilerine özgü kısıtlamaları bulunmaktadır. Ve bu sınırlamaları göz ardı ettiğiniz anda genellikle sinyaller çözümlenebilir. Buna ek olarak, sensörler ve elektrik devreleri aracılığıyla bir miktar rastgele elektriksel gürültü işarete eklenmektedir. Saf işarete sürekli zamanla değişen oranda etkiyen elektriksel gürültü bilginin miktarını ve kalitesini etkileyecektir. Sonuç olarak, herhangi bir sensörden elde edilen bilgi, tüm kestirim dizisinin bir parçası gibi yorumlanacak şekilde nitelendirilmelidir. Ve çözümlenmeli ölçüm modelleri genel olarak bazı rastgele ölçüm gürültüsünü ya da belirsizliği birleştirir.

Gerçek durum dönüşüm modelinin, tamamen bilinmemesi ek bir problem olarak ortaya çıkmaktadır. Şimdiki durum dönüşümlerine dayandırılmış modelleri kullanarak birbirleriyle ilişkili ve kısa aralıklar üzerinden tahminler yaparken, bazı tahminler dönüşümlerin tahmin edilebilir olduğunu fakat bunun her zaman mümkün böyle olmadığını kabul eder. Sonuç olarak sensör bilgilerinde olduğu gibi, durumun devam eden kestirimleri, kestirimler dizisinin bütününde ölçümlerle birleştirilerek nitelendirilmelidir. Buna ek olarak sistem modelleri genel olarak bazı rastgele hareketleri ya da belirsizliği birleştirir.

1.2.2.5. Kalman Filtresi

Gürültülü sensör ölçümlerinden gelen verilerin rastlantısal kestirimi için kullanılan matematiksel modeller içerisinde en iyi bilinen ve en sık kullanılanıdır. Kalman filtresi, Rudolf E. Kalman tarafından 1960 yılında yayınlanan ve ayrık verilerin doğrusal filtrelenmesi problemine ışık tutan ünlü yayınından sonra bu ismi almıştır.

Kalman filtresi, bir dizi matematiksel eşitlikten oluşan bir kestiricidir ve bazı ön varsayımlar karşılandığında kestirilen hata kovaryansını en aza indirdiği için en uygun filtreleme yöntemidir. Kalman filtresi ortaya çıktıktan bu zamana kadar otomatik veya desteklenmiş seyir uygulama ve araştırmalarına konu olmuştur. Bilgisayarlı hesaplamaların zamanla gelişmesiyle ve Kalman filtresini çok kullanışlı hale getirmiştir. Özellikle takip işlemlerinde (radar, kamera, ... vs.) geniş bir kullanım alanı bulmuştur.

1.2.2.5.1. Ayrık Kalman Filtresi

Bu kısım ölçümlerin ve kestirim yapılacak durumun zamanda ayrık noktalar olarak ifade edildiği ayrık sistemler için anlatılır.

1.2.2.5.1.1. Kestirim Yapılacak Sistem

Kalman filtresi, ayrık zamanlı bir sistemin durumu olan $x \in \mathfrak{R}^n$ 'i kestirmeye çalışır. Sistemin k adımındaki durumu (1.56) bağıntısında verilmiştir:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (1.56)$$

Sistemin ölçüm değeri olan $z \in \mathfrak{R}^m$ (1.57) bağıntısıyla tanımlanır.

$$z_k = Hx_k + v_k \quad (1.57)$$

Rastgele değişkenler olan w_k ve v_k sırasıyla sistem ve ölçüm gürültüleridir. Bu iki gürültünün normal dağılıma sahip, beyaz ve birbirinden bağımsız olduğu varsayılır.

$$p(w) \sim N(0, Q) \quad (1.58)$$

$$p(v) \sim N(0, R) \quad (1.59)$$

Pratikte sistem gürültüsü kovaryansı Q ve ölçüm gürültüsü kovaryansı R zamanla değişebilir fakat burada sabit oldukları varsayılmıştır.

(1.56) diferansiyel denklemindeki $n \times n$ boyuta sahip A matrisi gürültünün yokluğunda $k-1$ adımındaki durumu k adımındaki durumuyla ilişkilendirilir. Pratikte A matrisi her adımda değişebilir fakat burada sabit olduğu varsayılır. $n \times 1$ boyutlu B matrisi kontrol girişi $u \in \mathfrak{R}^l$; x durumu ile ilişkilendirilir. (1.57) bağıntısındaki $m \times n$ boyutlu H matrisi ölçüm değeri z_k 'yı durum ile ilişkilendirir. Pratikte H matrisi de zamanla değişebilir fakat burada sabit olduğu varsayılmıştır.

1.2.2.5.1.2. Kalman Filtresinin Hesapsal Tabanı

$\hat{x}_k^- \in \mathfrak{R}^n$, k adımından önce verilen bilgiler ışığında k adımındaki önsel durum kestirimi olarak tanımlanır. $\hat{x}_k \in \mathfrak{R}^n$ ise k adımında yapılan ölçümler ışığında sonsal durum kestirimi olarak tanımlanır. Önsel ve sonsal kestirim hataları (1.60) ve (1.61) bağıntılarındaki gibi tanımlanabilir.

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (1.60)$$

$$e_k \equiv x_k - \hat{x}_k \quad (1.61)$$

Böylelikle önsel kestirim hatasının kovaryansı (1.62) bağıntısındaki gibidir:

$$P_k^- = E[e_k^- e_k^{-T}] \quad (1.62)$$

Sonsal kestirim hatasının kovaryansı da (1.63) bağıntısındaki gibidir:

$$P_k = E[e_k e_k^T] \quad (1.63)$$

Bu aşamada; ölçüm tahmini $H\hat{x}_k^-$ ve mevcut ölçüm z_k 'nın ağırlıklandırılmış farkı ve önsel kestirim olan \hat{x}_k^- arasında doğrusal bir kombinasyonla sonsal durum kestirimi \hat{x}_k 'yı hesaplamak gerekir. Bu da (1.64) bağıntısındaki gibi hesaplanır:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (1.64)$$

(1.64) bağıntısındaki $(z_k - H\hat{x}_k^-)$ farkı artık veya kalıntı olarak adlandırılır. Artık değer tahmin edilen $H\hat{x}_k^-$ ölçüm değeri ise gerçek ölçüm değeri z_k arasındaki farklılığı yansıtır. Artık değerın sıfır olması tahmin edilen değer ile gerçek ölçümün uyduğunu gösterir.

Bağıntı (1.64) 'deki $n \times m$ boyutlu K matrisi, eşitlik (1.63) 'daki sonsal hata kovaryansını en aza indirecek şekilde seçilir. En aza indirme işlemi [36,37] 'da ayrıntılı

şekilde görülebilir. Eşitlik (1.63) 'u en aza indirilen K 'nın bir biçimi (1.65) bağıntısındaki gibidir:

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (1.65)$$

Eşitlik (1.65) 'e bakıldığında, ölçüm hatası kovaryansı R sifıra yaklaştıkça K kazancının artık değer üzerindeki etkisi artar.

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (1.66)$$

Diğer yönden önsel hata kestiriminin kovaryansı P_k^- sifıra yaklaştıkça K kazancının artık değer üzerindeki etkisi azalır.

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (1.67)$$

Başka bir deyişle ölçüm hata kovaryansı R sifıra yaklaştıkça gerçek ölçüm değeri z_k , tahmin edilen ölçüm değeri $H \hat{x}_k^-$ 'ya göre çok daha güvenilirdir. Bunun tersi olarak önsel kestirim hata kovaryansı P_k^- sifıra yaklaştıkça tahmin edilen ölçüm $H \hat{x}_k^-$, ölçüm değeri z_k 'ya göre çok daha güvenilirdir.

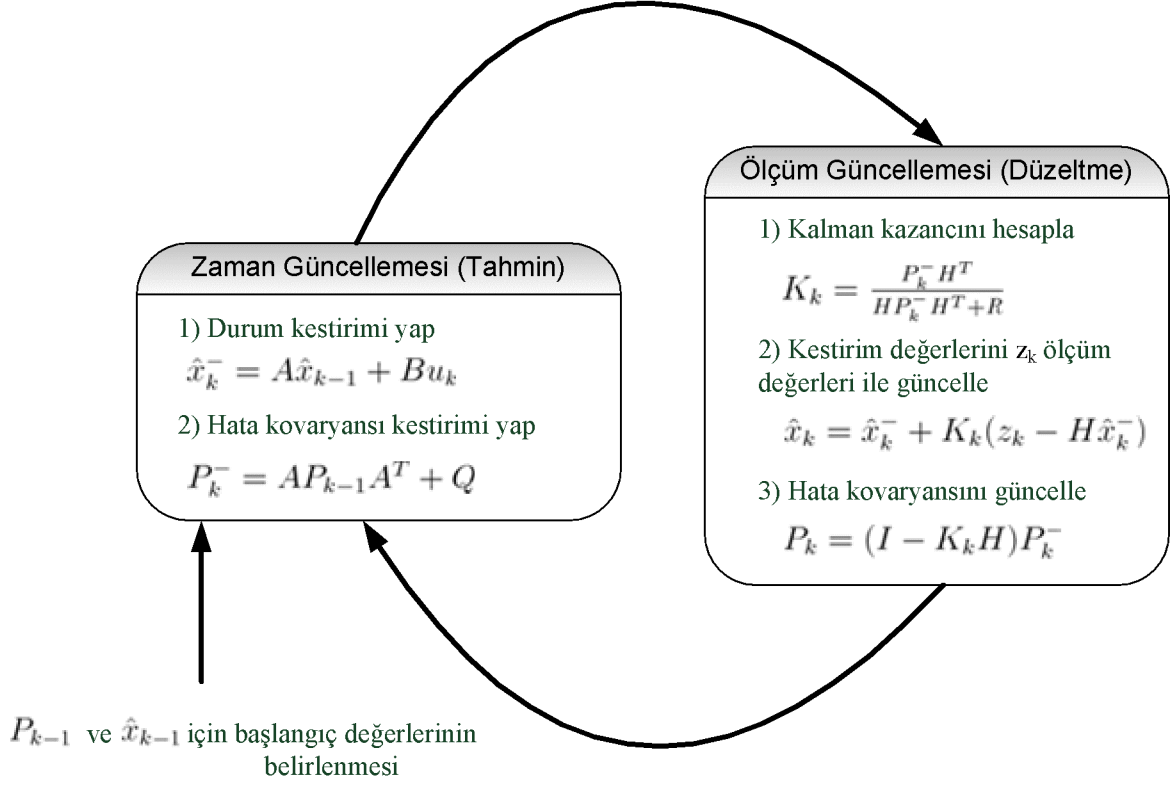
1.2.2.5.1.3. Ayrık Kalman Filtresi Algoritması

Kalman filtresi geribesleme kontrollerini kullanarak bir sistemin durumunu kestirir. Filtre sistem durumunu verilen herhangi bir anda kestirir, daha sonra geri besleme olarak ölçüm değerlerini alır. Kalman filtresi eşitlikleri iki gruptan oluşmaktadır:

- Zaman güncellemesi eşitlikleri,
- Ölçüm güncellemesi eşitlikleri.

Zaman güncellemesi eşitlikleri, bir sonraki adımda kullanılacak önsel kestirimleri elde edebilmek için mevcut sistem durumu ve hata kovaryansı kestirimini ileri doğru izdüşürmekle sorumludur. Ölçüm güncellemesi eşitlikleri ise yeni ölçüm değerlerini önsel kestirim değerlerine dahil ederek sonsal kestirimi elde etmekle yükümlüdür.

Zaman güncellemesi eşitlikleri, tahmin edici eşitlikler; ölçüm güncellemesi eşitlikleri ise düzeltici eşitlikler olarak düşünülebilir. Şekil 1.6. 'da görüldüğü üzere Kalman filtresi algoritması tahmin edici – düzeltici tipte bir algoritmadır.



Şekil 1.6. Kalman filtresinin bir döngüsünde yapılan işlem adımları

Kalman filtresi, özyinelemeli yapısı sayesinde birçok probleme uygulanabilir. Kalman filtresi modellenecek sistemin durumları arasındaki ilişkinin doğrusal olmasını gerektirir. Bağlantı (1.41) 'deki A geçiş matrisi sistemin bir önceki durumu ile bir sonraki durumu arasındaki doğrusal ilişkiyi tanımlar. Sistemin durumları arasındaki ilişkinin doğrusal olmaması durumunda ise Genişletilmiş Kalman Filtresi kullanılır [38].

Kalman filtresi yapılan varsayımlar dahilinde doğrusal bir sistemin kestirimi için uygundur. Nesne takibi işleminde durum biraz daha farklı olabilir. Kalman filtresi ile tek bir sistemin durumunu kestirirken, izlediğimiz sahne içerisinde birden fazla nesne bulunabilir. Bunun yanı sıra sahnenin belirli bölgelerinde hatlı ölçümden veya nesne algılama algoritmalarındaki hatalardan dolayı yalancı nesnelere algılanabilir. Kalman filtresinin tek başına bu durumlarla baş etmesi mümkün değildir. Birden fazla nesnenin

takibi ve yalancı nesnelere göz ardı edilmesi için veri ilişkilendirmesi yapmak şarttır. Ancak veri ilişkilendirilmesi yapıldıktan sonra tekil nesnelere için kalman filtresi ile durum kestirimi yapılabilir.

1.2.2.6. Mahalanobis Uzaklığı

İstatistikte, Mahalanobis uzaklığı 1936 yılında P.C.Mahalanobis tarafından uzaklık ölçüsü olarak ileri sürülmüştür [39]. Bu uzaklık, hangi farklı örüntülerin belirlendiğine ve analiz edildiğine bağlı olarak, değişkenler arasındaki korelasyonlara dayanmaktadır. Bilinmeyen bir örnek setinin, bilinen bir setle arasındaki benzerliklerine karar verilmesinde kullanılan etkin bir yoldur. Veri kümesinin korelasyonlarının hesaplanması ve ölçek değişmez olması (ölçümlerin ölçeklerine bağımlı olmadan) bu uzaklığın Öklit uzaklığından farklı olduğu noktalarıdır.

Çok değişkenli $x = (x_1, x_2, x_3, \dots, x_N)^T$ vektörünün, ortalamalardan oluşan değer grubu $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$ ve kovaryans matrisi S ile tanımlanması aşağıdaki gibidir [40]:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (1.68)$$

Mahalanobis uzaklığı, S kovaryans matrisi ile birlikte aynı dağılıma sahip \vec{x} ve \vec{y} rastgele vektörlerinin arasındaki benzeşmezlik ölçümü olarak da tanımlanabilir.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (1.69)$$

Eğer kovaryans matrisi birim matris ise, Mahalanobis uzaklığı Öklit uzaklığına indirgenir. Eğer kovaryans matrisi köşegen matris ise ortaya çıkan uzaklık ölçüsüne normalize edilmiş Öklit uzaklığı denir.

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^N \frac{(x_i - y_i)^2}{\sigma_i^2}} \quad (1.70)$$

Yukarıdaki bağıntıda σ_i örnek setindeki x_i değişkeninin standart sapmasıdır.

Mahalanobis, 1927 yılında yapılan, ölçümlere dayanan kafataslarının benzerliklerinin belirlenmesi problemiyle ortaya çıkmıştır [41].

Mahalanobis uzaklığı cluster analizi ve diğer sınıflandırma tekniklerinde geniş bir kullanım alanına sahiptir. Bu yöntem çok değişkenli istatistik testi için kullanılan Hotelling's T-square dağılımı ve güdümlü sınıflandırma için kullanılan Fisher'in Doğrusal Diskriminant Analizi ile yakından ilgilidir [43].

Mahalanobis uzaklığını, bir test noktasının N adet sınıftan hangi birine ait olduğunu belirlemede kullanabilmek için ilk olarak genellikle her bir sınıfa ait bilinen örneklere dayandırılarak her bir sınıfın kovaryans matrisi kestirilir. Ardından verilmiş test örneğinde, her bir sınıf için Mahalanobis uzaklığı hesaplanır ve bu uzaklığın minimum değerine bağlı olarak test noktasının hangi sınıfa ait olduğu belirlenir. Yukarıda verilmiş olasılıksal yorumlamayı kullanmak, maksimum benzerliğe dayanarak sınıfı seçme ile eşdeğerdir.

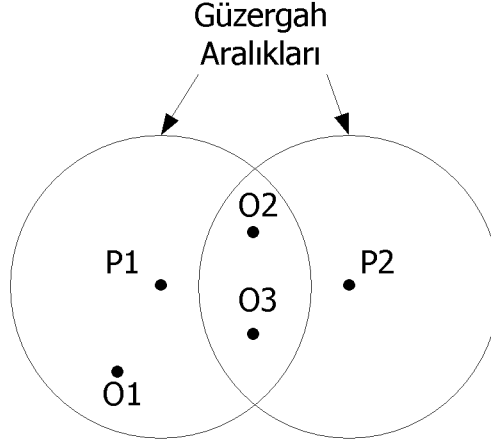
1.2.2.7. Veri İlişkilendirmesi

Birçok gözetleme sistemi birden fazla hedefi takip etmeye gereksinim duyduğundan çoklu hedef takibi, takip uygulamalarında önemli bir yer tutar. Sahne içerisindeki hareketli nesnelere hakkında sensörlerden gelen ve çeşitli hesaplamalar sonucu elde edilen bilgiler gözlem sonucu olarak adlandırılır. Örneğin; bir radar takip uygulamasındaki cismin konumu gözlem sonucu olabileceği gibi, bir kameralı takip sistemindeki cismin renk dağılımı bilgisi de gözlem sonucu olarak değerlendirilebilir.

Bir cismin sahneye girdiği anda seçilen herhangi bir ana kadar olan gözlem sonucu bilgileri nesnenin güzergâhı olarak adlandırılır. Örneğin; bir gözetleme uygulamasındaki belirli bir cismin sahne içindeki konumlarının bütünü bu cismin güzergâhını temsil eder.

Sahne içerisinde birden fazla nesne bulunduğu veya hatalı ölçüm sonuçlarından kaynaklanan sahte gözlem sonuçları mevcut olduğunda hangi gözlem sonucunun hangi güzergâh ile ilişkilendirileceğini belirlemek bir problem haline dönüşebilir. İzlenebilecek en basit yol güzergâh için yapılan kestirim noktasını en yakın gözlem sonucu ile ilişkilendirmektir. Yakınlık değerini belirlerken uzaklık ölçüsü olarak öklit uzaklığı kullanılabilir. İzlenen bu en yakın komşuluk yaklaşımında bir gözlem sonucu birden fazla güzergâh için en yakın olabilir. Dolayısıyla tek bir gözlem sonucu birden fazla güzergâh ile ilişkilendirilebilir. Bunu engellemek için izlenecek bir yol güzergâh aralığının kullanıldığı genel en yakın komşuluk (GNN) yaklaşımını uygulamaktır. Güzergâh aralığı, söz konusu

olan güzergâh için bir sonraki adımda yapılan ölçümlerin içinde bulunabileceği bölge olarak tanımlanır. Şekil 1.7. 'de farklı iki güzergâh için güzergâh aralıkları gösterilmiştir. Eğer herhangi bir adımda bir güzergâhın, güzergâh aralığında gözlem sonucu bulunmuyorsa bir ilişkilendirme yapılmaz.



Şekil 1.7. İki farklı güzergâh için güzergâh aralıkları ve elde edilen gözlem sonuçları

Şekil 1.7. 'deki O_1 , O_2 , O_3 noktaları gözlem sonuçlarını, P_1 ve P_2 noktaları farklı iki güzergâh için tahmin edilen nesne konumlarını, çemberler ise güzergâhlar için güzergâh aralıklarını ifade eder.

GNN yaklaşımında sadece en iyi ilişkilendirmeler tutulur. İlişkilendirilmemiş ölçüm sonuçları (Şekil 1.7. için O_3) yeni güzergâhlar oluşturur. Güzergâhın onaylanması (uygunluğunun kabulü) ve silinmesi belirli kurallara göre yapılır. Örneğin; ardışık 4 çerçevenin üçünde güzergâh ile ilişkilendirme yapılmışsa güzergâhın onaylanması ve ardışık N çerçevede güzergâh ilişkilendirilmemişse güzergâhın silinmesi gerçekleşir.

GNN ilişkilendirme şeklinde bir gözlem sonucunun sadece bir hedef nesneden elde edildiği varsayılır. Ortak gözlem sonucu bulandırmayan güzergâhlar uygun olarak tanımlanır.

Gözlem sonuçları güzergâhlarla ilişkilendirildiğinden filtreleme işlemi ile güncellenir. Filtreleme işlemi için Kalman filtresi sıkça kullanılır. Son olarak Kalman filtresi ile güzergâhların sonraki konumları belli olacak şekilde durum kestirim yapılır. Kalman filtresindeki hata kovaryansı takip işlemindeki belirsizliği verir, güzergâh aralığı belirleme ve ilişkilendirmede kullanılır.

GNN yaklaşımı sadece hedef nesnelere arasında geniş boşluklar, kesin ölçümler ve çok az yanlış alarm olduğunda iyi sonuç verir [43].

Güzergâhların yanlış alarm değerleriyle güncellenmesi sonraki adımlarda doğru ilişkilendirme şansını düşürür. Birbirine yakın konumlardaki güzergâhlarda ilerleyen cisimler için hatalı ilişkilendirme daha kötü sonuçlar doğurur [44].

Hatalı ilişkilendirme Kalman filtresi için ek bir hata kaynağını temsil eder. GNN 'nin performansını artırmak için önerilen bir yaklaşım Kalman filtresinin kovaryans matrisini belirsizliği yansıtacak şekilde artırmaktır [45,46]. Benzer bir yaklaşımda belirsiz ilişkilendirme durumları için kazancı düşürmektir [44].

Belirsiz ilişkilendirme durumlarında kullanılabilecek bir diğer yaklaşım Birleşme Yeri Olasılıksal Veri İlişkilendirmesi (JPDA) kullanmaktır. JPDA 'da belirsiz durumlarda güzergâh, güzergâh aralığı içindeki gözlem sonuçlarının ağırlıklı toplamı ile güncellenir. Bu şekilde bir gözlem sonucu birden fazla güzergâhın güncellenmesinde kullanılabilir. Örnek olarak Şekil 1.7. için; O_1 , O_2 , O_3 gözlem sonuçları 1 nolu güzergâhın, O_2 ve O_3 ise 2 nolu güzergâhın güncellenmesinde kullanılacaktır.

JPDA ve GNN yaklaşımları Kalman filtresinin kovaryans matrisini artırarak ilişkilendirme belirsizliğinin üstesinden gelmeye çalışırlar. Fakat Fleskes vd. 'in belirttiği gibi Kalman filtresinin kovaryans matrisini arttırmak güzergâhla ilgili olmayan gözlem sonuçlarının güzergâh aralığı içine alınmasına yol açar. Ayrıca JPDA yaklaşımında birbirine yakın nesnelere güzergâhlarının takibinde kaynaşma problemi oluşur. Örneğin; Şekil 1.7. 'de O_2 ve O_3 her iki güzergâhın güncellenmesinde de kullanılacağından bu iki güzergâh birleşir.

GNN ve JPDA 'daki bu problemler ve bilgisayarların hesaplama kapasitelerinin artması modern sistemlerde veri ilişkilendirme tekniği olarak Çoklu Hipotez Takibi (MHT) algoritmasını tercih edilir hale getirmiştir.

1.2.2.8. Çoklu Hipotez Takibi Algoritması

MHT, Şekil 1.7. 'deki gibi bir belirsizlik durumu olduğunda bunun üstesinden gelmek için birden fazla hipotez oluşturur. Hipotezler içerisinde güzergâhları barındıran bir çeşit gruplamadır. Şekil 1.7. için T_1 ve T_2 güzergâhlarının P_1 ve P_2 tahmin edilen noktalarıyla H_1 hipotezini oluşturduğu varsayalım. O_1 , O_2 ve O_3 gözlem sonuçları elde edildikten sonra mevcut olan H_1 hipotezinden 10 adet hipotez oluşturulur. Örneğin

oluşacak bir hipotez T_1 'in O_1 ile güncellenmesi, T_2 'nin O_2 veya O_3 ile güncellenmesi ve kalan gözlem sonucunun da yeni bir güzergâh oluşturmasıdır. Bir başka beklenilmedik fakat mümkün olan hipotez ise O_1 , O_2 ve O_3 gözlem sonuçlarının yeni kaynaklardan gelmesidir.

Reid, çoklu hipotez takibine ilk algoritmik yaklaşımı getirmiştir [47]. Şekil 1.8. 'de verildiği gibi T_3 , T_1 ile O_1 gözlem sonucunun ilişkilendirildiği güzergâhtır. NT_1 , O_1 gözlem sonucunun T_1 veya T_2 güzergâhları ile ilişkilendirilmediği durumda oluşan yeni güzergâhtır.

<u>Güzergâh İlişkilendirmeleri</u>
• $T_3(T_1, O_1)$
• $T_4(T_2, O_2)$
• $T_5(T_2, O_3)$
• $T_6(T_1, O_2)$
• $T_7(T_1, O_3)$
<u>Oluşan Hipotezler</u>
$H_1: T_1, T_2, NT_1, NT_2, NT_3$
$H_2: T_3, T_4, NT_3$
$H_3: T_3, T_5, NT_2$
$H_4: T_2, T_3, NT_2, NT_3$
$H_5: T_5, T_6, NT_1$
$H_6: T_2, T_6, NT_1, NT_3$
$H_7: T_4, T_7, NT_1$
$H_8: T_2, T_7, NT_1, NT_2$
$H_9: T_1, T_4, NT_1, NT_3$
$H_{10}: T_1, T_5, NT_1, NT_2$

Şekil 1.8. Örnek veri ilişkilendirmesi için oluşturulan hipotezler

Ortak gözlem sonucu bulunmayan güzergâhlar uyumlu olarak tanımlanır. Üstteki örnekte yeni hipotezler oluşturulmadan önce T_1 ve T_2 ortak bir gözlem sonucu barındırmazlar. Oluşturulan yeni hipotezlerde de her bir hipotez içerisindeki güzergâhlar en fazla bir gözlem sonucu ile ilişkilendirilir. Buradaki amaç her bir güzergâhı en fazla bir gözlem sonucu ile ilişkilendirerek mümkün olan bütün ihtimalleri değerlendirmektir.

Reid 'in algoritmasını kullanarak hipotezler oluşturularak bir sonraki adıma taşınır. Yeni gözlem sonuçları geldiğinde güzergâh – gözlem sonucu ilişkilendirmeleri yapılarak hipotezler yeni hipotez kümesine genişletilir. Bu şekilde sürekli olarak MHT algoritması işler.

1.2.2.8.1. Güzergâh ve Hipotez Değerlendirmesi

MHT algoritmasının herhangi bir anındaki hipotezlerin hangilerinin mümkün olabileceğini belirlemek için bir değerlendirme ölçütüne ihtiyaç vardır. Bu değerlendirme ölçütünün yanlış alarm yoğunluğu, algılama ardışıklığı ve güzergâhlardaki gözlem sonuçlarının hareketinin tutarlılığı gibi veri ilişkilendirme probleminin bütün yönlerini ele alması gerekir. Reid bu tipte olasılıksal bir yaklaşım önermiştir [47]. Slitter tarafından önerilen logaritmik olabilirlik oranı (LLR) Reid 'in yaklaşımına matematiksel olarak eşdeğer fakat hesaplama bakımından tercih edilir niteliktedir [48].

Bir güzergâhın içinde bulunan veri kombinasyonu için olabilirlik oranı (LR) özyinelemeli bir ilişki kullanarak bağıntı (1.71) 'deki gibi tanımlanır [44].

$$LR = \frac{p(D|H_1)P_0(H_1)}{p(D|H_0)P_0(H_0)} \triangleq \frac{P_T}{P_F} \quad (1.71)$$

H_0 ve H_1 hipotezleri sırasıyla P_F ve P_T olasılıklarına sahip doğru hedef ve yanlış alarm hipotezleridir. D burada veriyi temsil eder. $p(D|H_i)$, H_i 'nin doğru olduğu varsayımına göre alınan verinin olasılık yoğunluk fonksiyonunu temsil eder. $P_0(H_i)$ ise H_i 'nin önsel olasılığını temsil eder.

Doğru hedef genellikle ardışık birkaç adım boyunca güzergâh aralığı içerisinde yer alan nesnelere olarak tanımlanır. LLR değeri bağıntı (1.72) 'deki gibi hesaplanır.

$$LLR = \frac{P_T}{P_F} \quad (1.72)$$

LLR, bağıntı (1.73) 'deki gibi direkt olarak doğru hedef olasılığına çevrilebilir.

$$\frac{P_T}{P_F} = \frac{P_T}{1-P_T} = e^{LLR}$$

$$P_T = \frac{e^{LLR}}{1+e^{LLR}} \quad (1.73)$$

Böylelikle güzergâhın geçerliliğini belirlemek için sadece LLR ‘nin hesaplanması yeterli olacaktır. LLR ‘yi aynı zamanda güzergâhın puanı olarak adlandırabiliriz. Güzergâhı onaylamak için güzergâh puanı kullanılabilir [44].

k anındaki güzergâh puanı $L(k)$ uygun bir özyinelemeli şekle, bağıntı (1.74) ‘deki gibi dönüştürülebilir [43]:

$$L(k) = L(k-1) + \Delta L(k)$$

$$\Delta L(k) = \begin{cases} \ln(1 - \hat{P}_D) & \text{k adımında güncelleme yapılmamışsa} \\ \Delta L_u(k) & \text{k adımında güzergâh güncellenmişse} \end{cases} \quad (1.74)$$

Güzergâhta bir güncelleme gerçekleşmediği zaman güzergâh puanındaki kayıp beklenen nesne algılama oranına \hat{P}_D bağlı bir değerdir. Güncelleştirme gerçekleştirildiğindeki ΔL_u kazancı ise artık değer (tahmin edilen konum ile ölçüm yapılan konum arasındaki fark), hata kovaryans matrisi ve hatalı algılama olasılıklarının bir fonksiyonudur. Cox güncelleme değeri olarak Mahalonobis uzaklığının kullanılabileceğini söylemektedir [49].

Hesaplanan güzergâh puanları ile birlikte bir hipotezin puanı barındırdığı güzergâhların toplamına eşittir. Eldeki hipotez puanları ile hipotezlerin olasılıkları hesaplanabilir [43]. Son olarak güzergâhların olasılıkları içinde bulunduğu tüm hipotezlerin olasılıkları toplamına eşittir. Şekil 1.8. ‘deki örnek için T_3 güzergâhının olasılığı H_2 , H_3 ve H_4 hipotezlerinin olasılıkları toplamına eşittir.

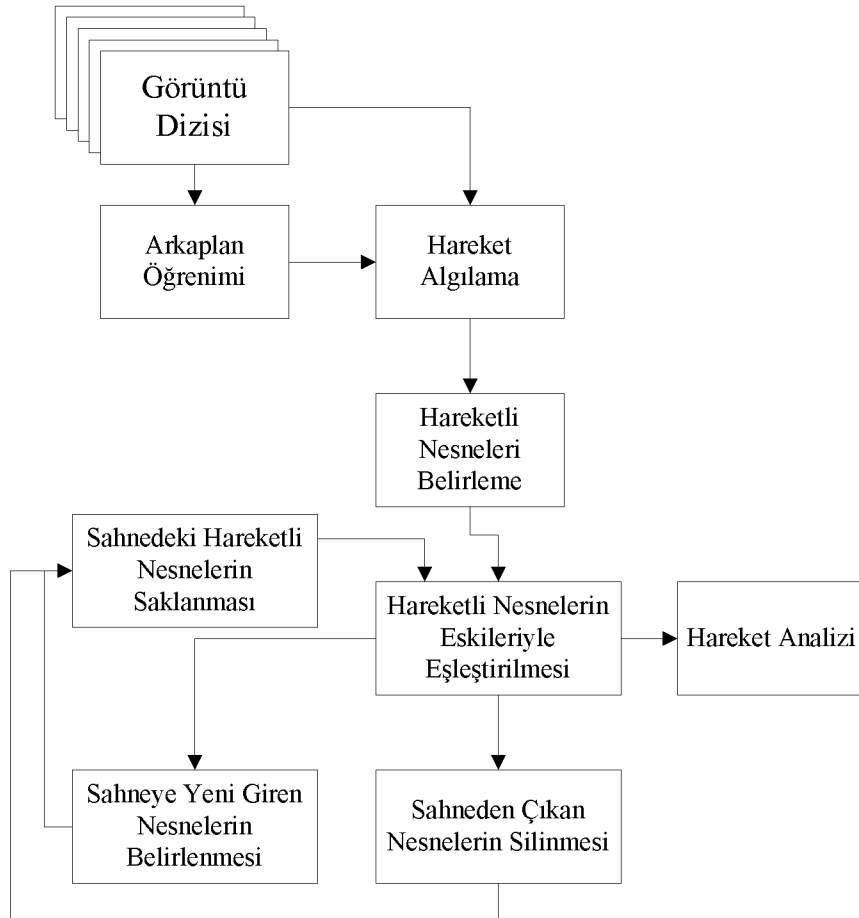
Güzergâh ve hipotez olasılıklarını hesaplamak için basit hesaplamalar kullanılabilir. Olasılık değerleri hesaplanırken hedef nesnelerin hareketi ve ölçüm hataları için normal dağılım, yanlış alarm (parazit ve gürültüler) ve yeni hedefler için tek biçimli dağılım ve sembolik bir P_0 değeri seçilebilir [44]. Pratikte MHT sistemlerini geliştirenler hep bu varsayımları kullanmıştır. Bu varsayımlar ışığında MHT, geliştirilen diğer yaklaşımlara göre oldukça iyi sonuçlar vermiştir.

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Giriş

Yapılan deneysel çalışma, gerçek tünel görüntüleri üzerinde araç algılama ve takibi üzerinedir. Bilgisayarla görme alanında popüler olan nesne algılama algoritmaları ile tünel içerisindeki araçları algılama ve bu araçların kamera görüntüsü içerisinde yer aldıkları süre boyunca takibinin gerçekleştirilmesi amaçlanmıştır. Bilgisayarla görme işlemlerinin gerçek zamanda icra edilmesi beklendiğinden algoritmalar arasında hız değerlendirmesi de yapılmıştır. Tüm bu işlevleri gerçekleştirecek bir uygulama geliştirilmiştir.

Şekil 2.1’de tünel içerisine yerleştirilen sabit bir kameradan alınan görüntüleri yorumlayarak olay algılamayı gerçekleştirmesi düşünülen sistemin genel yapısı verilmiştir.



Şekil 2.1. Olay algılama sistemi için düşünülen sistemin genel yapısı

Düşünülen sistemde arkaplan modellemeye dayalı bir nesne algılama vardır. Arkaplan modelleme için mevcut olan algoritmalar denenmiş ve sonuçları irdelenmiştir. Arkaplan modeli belirlendikten sonra yeni gelen görüntü çerçevelerindeki hareketli kısımlar arkaplan çıkarma işlemi yapılarak bulunur. Bulunan bu hareketli kısımlar bağlı bileşen etiketlemesi yapılarak birbirinden ayırt edilir. Birbirinden ayırt edilen bu bölgeler aradığımız özelliklerde bir nesne olmayabilir. Örneğin bu kısımlar bir nesnenin parçalarından bir tanesi ya da görüntülerdeki gürültülerden kaynaklanan hatalı bir algılama olabilir. Gürültüden kaynaklanan hatalı bölgeler genellikle küçük alana sahip olduğundan morfolojik işlemler kullanılarak elimine edilebilir. Benzer şekilde bir nesnenin parçaları arasında kopukluk varsa ve bu parçaların sınırları birbirlerine yakınsa morfolojik işlemler kullanılarak bu parçalar birleştirilebilir.

Sistemin başlangıç anında sahnede bir nesne varsa bile hareketli bir cisim olarak algılanmaz. Bunun nedeni arkaplan modeli ve hareket verisi gibi bazı parametrelerin başlangıç anında bulunmamasıdır. Sistemin çalışması esnasında her çerçevede algılanan nesnelere daha önceki çerçevelerde algılanan nesnelere ile ilişkilendirilmeye çalışılır. Bu nesne takip işleminin ana gereksinimidir. Eğer algılanan hareketli nesnelere daha önceki nesnelere ile ilişkilendirilemiyorsa, sistemin doğru çalışması durumunda, sahneye yeni bir cisim girmiş demektir. Bu yeni giren nesnenin bilgileri bellekte saklanır. Daha sonraki çerçevelerde algılanan hareketli nesnelere bu yeni nesne ve bellekte saklanan eski nesnelere oluşturduğu kümedeki nesnelere ile ilişkilendirilmeye çalışılacaktır. Bu işlemin tersi olarak eğer bellekte tutulan bir nesne belirli bir süre boyunca yeni çerçevelerdeki nesnelere ile ilişkilendirilemiyorsa nesne ya sahneden çıkmıştır ya da arkaplana dahil olmuştur. Dolayısıyla bu nesne ile daha fazla ilgilenmeye gerek kalmaz ve bellekteki nesne kümesinden silinebilir.

Sahnedeki bir nesnenin davranışını anlamak için bu nesnenin takip verileri değerlendirilebilir. Takip işlemi sonucunda cismin görüntü uzayındaki hızı, zamansal hız değişimi ve yörüngesi gibi veriler elde edilebilir.

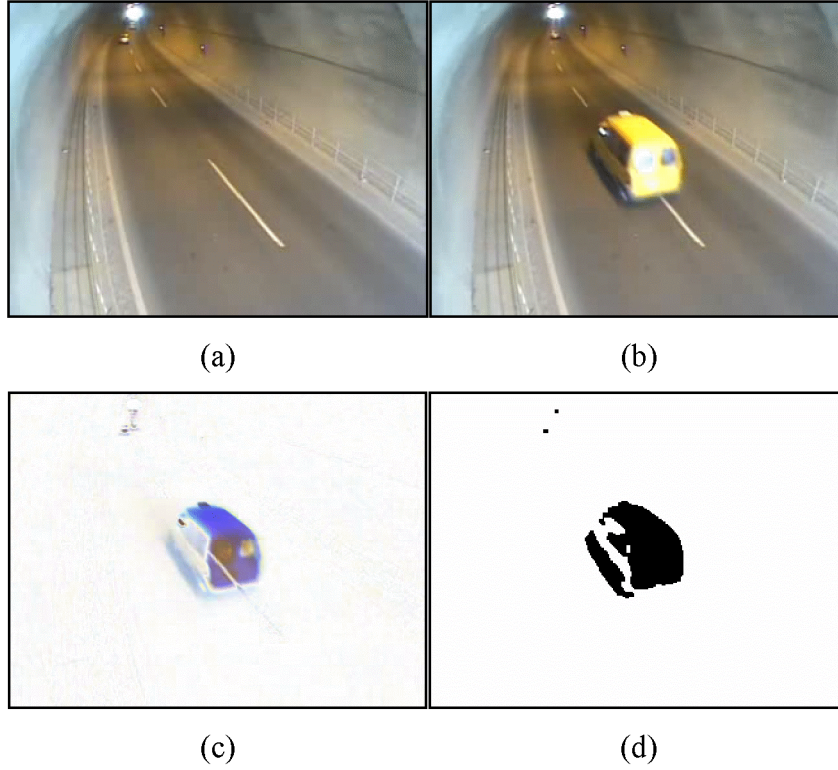
Bundan sonraki aşamalarda tünel içerisinden alınan görüntüler üzerinde çeşitli nesne algılama yöntemlerinin uygulanışı ve elde edilen sonuçlar değerlendirilecektir.

2.2. Nesne Algılama Yöntemlerinin Tünel Görüntülerine Uygulanması

Çalışmada kullanılan algoritmalar Intel Centrino 1.6 MHz işlemcili 512 MB belleğe sahip bir dizüstü bilgisayarda denenmiştir. İşlemleri gerçek zamanda yapabilmek için tünellerden elde edilen görüntülerin boyutu 320x240'a ölçeklenmiştir. Algoritmalar için verilen hesaplama süreleri bu koşullar altında elde edilmiştir.

2.2.1. İlk Çerçeve Yöntemi

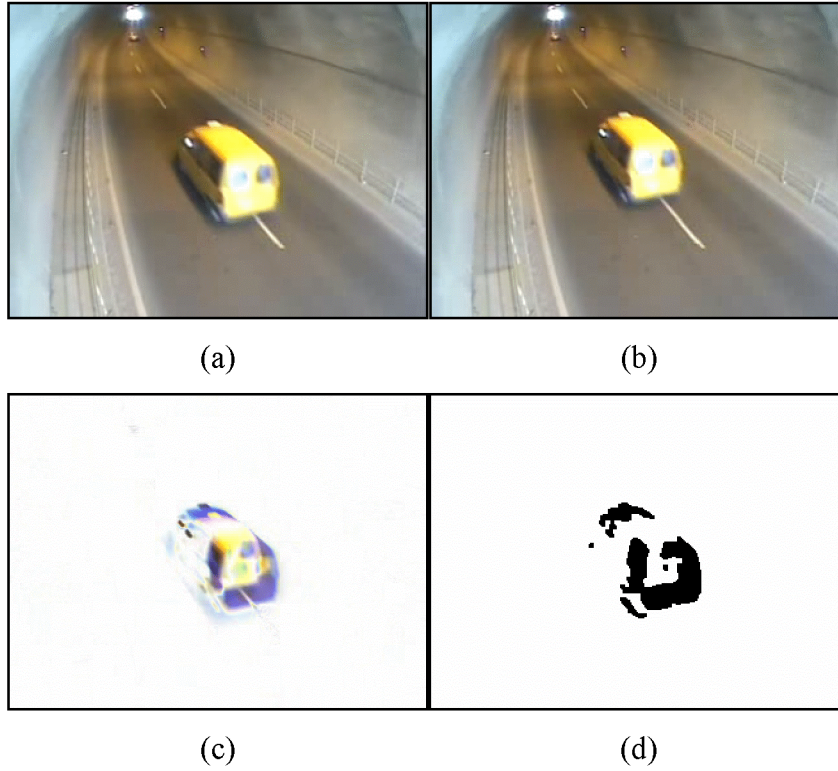
Alınan ilk çerçeve arkaplan modeli olduğu için, ilk çerçevedeki hareketli cisimler yöntemin doğruluğunu etkiler. Şekil 2.2.d 'deki küçük iki nokta ilk çerçevede bulunan araçtan kaynaklanan hatalı bir algılamadır.



Şekil 2.2. İlk çerçeve yöntemine göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

2.2.2. Ardışıl Fark Yöntemi

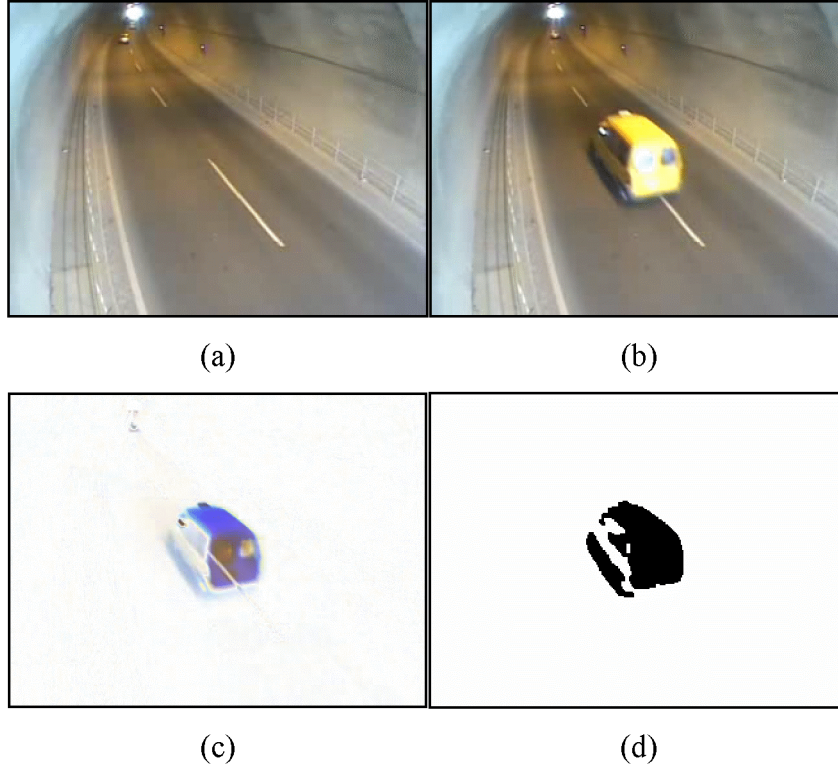
Ardışıl fark yönteminde bir cismin iki çerçeve arasındaki konum değişimi cismin boyutundan küçük ise, nesnenin parçalı olarak algılanması beklenen bir sonuçtur. Şekil 2.3.d 'de hareket eden minibüsün 5 parça halinde algılandığı görülmektedir. Ardışıl fark ve ilk çerçeve yöntemlerinde hesaplama yükü olmadığı için bir çerçeveyi işleme süresi 1 ms 'den azdır.



Şekil 2.3. Ardışıl fark yöntemine göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

2.2.3. Ortalama Yöntemi

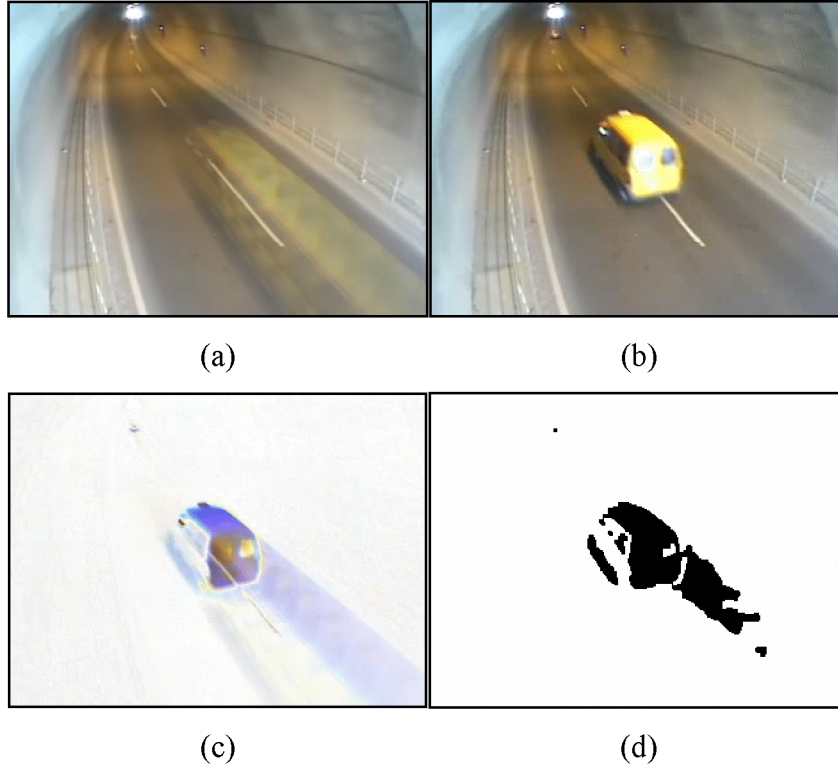
Ortalama yönteminde ardışıl fark ve ilk çerçeve yöntemlerine göre algoritmanın performansı açısından nispeten daha iyi bir sonuç elde edilmiştir. Yöntem, değişen aydınlatma koşullarına daha iyi uyum sağlayabilmesi için belirli aralıklarla arkaplan güncellemesi yapacak şekilde geliştirilebilir.



Şekil 2.4. Ortalama yöntemine göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

2.2.4. Adaptif Arkaplan Yöntemi

Adaptif arkaplan modelleme yönteminde geribeslemeli hesaplama nedeniyle geçmiş çerçevelerin etkisi görülmektedir. Şekil 2.5.a. 'daki araç izleri bundan kaynaklanmaktadır. Bunu engellemek için öğrenme katsayısının değeri azaltılabilir. Fakat öğrenme katsayısının değerini azaltmak sistemin arkaplanı daha geç öğrenmesine ve oluşan izlerin daha kalıcı olmasına neden olacaktır.

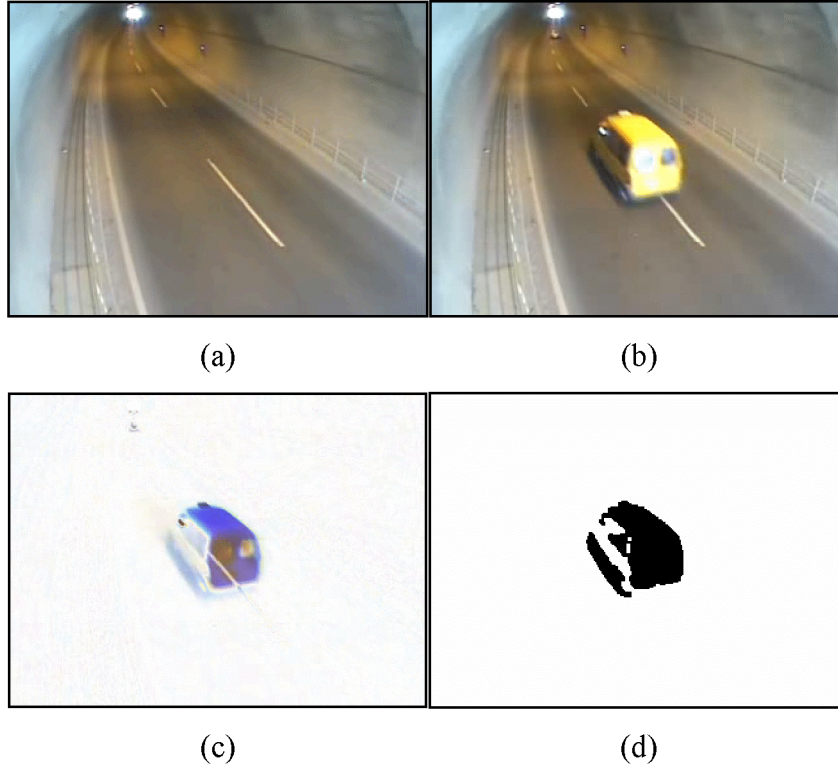


Şekil 2.5. Adaptif arkaplan yöntemine göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

2.2.5. ETO Algoritması

ETO algoritması, adaptif arkaplan ilk çerçeve ve ardışıl fark yöntemlerine göre daha yavaştır. Fakat sadece piksellerin parlaklık değerinin dağılımını bellekte tuttuğu için daha az bellek gerektirecektir. Öğrenme süresini artırmak ETO algoritmasının bellek gereksinimini ve hızını değiştirmeyecektir. Bu nedenle çok uzun sürelerde bile aynı performansta çalışacaktır. Bunu ortalama ve PCA yöntemleri için söylemek mümkün değildir. Ayrıca belirli aralıklarla arkaplan güncellenerek algoritmanın aydınlatma değişimlerine uyum sağlaması mümkündür. Belirli aralıklarla sürekli bir arkaplan güncellemesi yapmak yöntemin hızını değiştirmeyecektir.

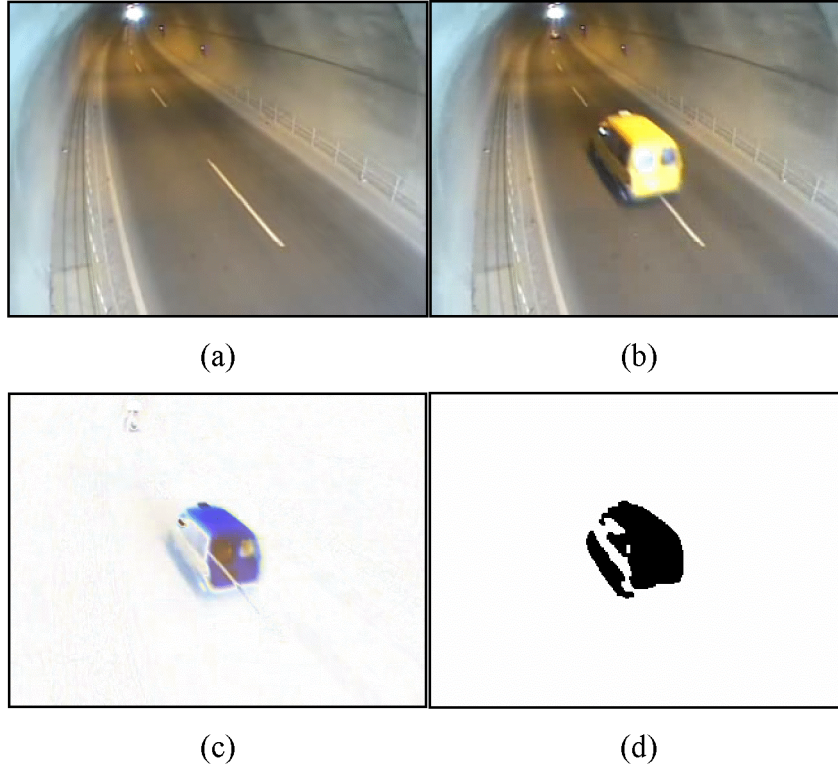
Piksellerin parlaklık değerlerinin dağılımı bellekte tutulurken, değerlere kuvantumlama uygulanabilir. Böylelikle algoritmanın bellek gereksinimi azalacak, hızı artacak ve gürültülere karşı daha az duyarlı olması sağlanacaktır.



Şekil 2.6. ETO algoritmasına göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

2.2.6. PCA ile Arkaplan Modelleme Yöntemi

PCA ile arkaplan modelleme yöntemi ortalama ve ETO yöntemlerine benzer sonuçlar üretmektedir. Fakat hesaplama süresi göz önüne alındığında performansı bu yöntemlere göre çok kötüdür. Hesaplama hızını artırmak için seçilen belirleyici özvektör sayısı azaltılabilir. Belirleyici özvektör sayısını azaltmak yöntemin hızını bir nebze artırmasına rağmen hem performansı düşecektir hem de ortalama yönteminin hızına erişemeyecektir.

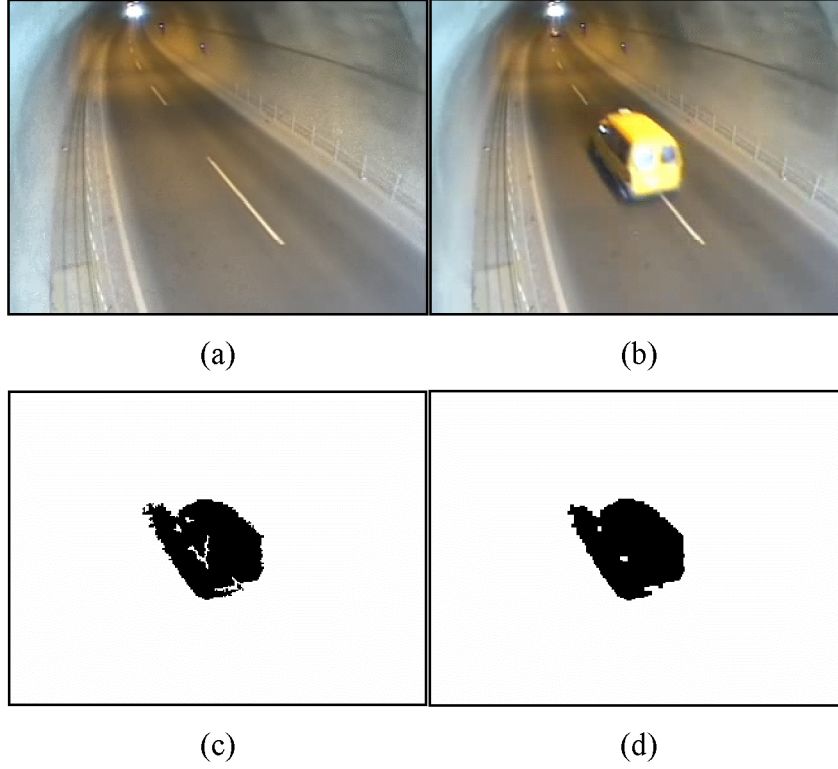


Şekil 2.7. PCA ile arkaplan modelleme yöntemine göre nesne algılama
 a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere

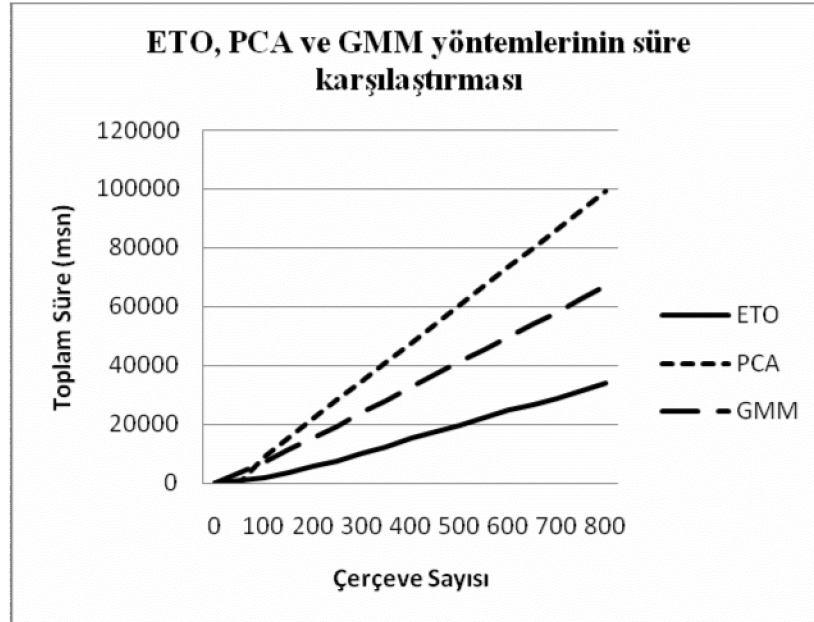
2.2.7. GMM Kullanarak Arkaplan Modelleme Yöntemi

Uygulanan yöntemler içerisinde GMM en doğru sonucu verendir. PCA ile karşılaştırıldığında daha da hızlı olduğu tespit edilmiştir. Yönteme gölge elimine etmeyi de dahil etmek daha doğru sonuç vermesini sağlayacaktır.

Algoritmaların hesaplama sürelerini karşılaştırmak için tünelden alınan 800 çerçevelik görüntü dizisi üzerinde algoritmalar denenmiştir. Değerlendirme sonucunda en hızlı algoritmanın, Şekil 2.9. 'da görüldüğü gibi, ETO olduğu gözlenmiştir. Hesaplama sürelerinin azlığı ve doğruluk oranlarının iyi olmamasından dolayı diğer yöntemler değerlendirmeye dahil edilmemiştir. PCA başlangıçta öğrenme sürecinde olduğu için ilk 50 çerçevede hızlı gözükmemektedir. Fakat öğrenme esnasında ve öğrenme gerçekleştikten sonra matris çarpım işlemlerinin getirdiği yükten dolayı büyük sürelerde hesaplama yapmaktadır.



Şekil 2.8. GMM kullanarak arkaplan modelleme yöntemine göre nesne algılama a) Öğrenilen arkaplan modeli b) Yeni gelen çerçeve c) Arkaplan çıkarma işleminin sonucu d) Algılanan hareketli nesnelere



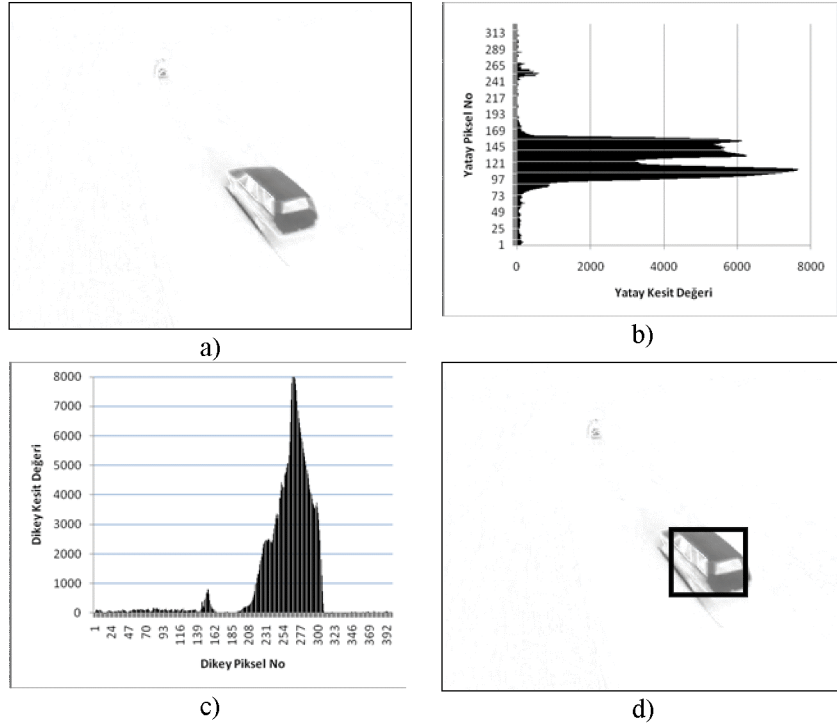
Şekil 2.9. ETO, PCA ve GMM yöntemlerinin süre karşılaştırması

2.3. Yatay ve Dikey Kesit Değerleri ile Nesne Algılama

Arkaplan modelleme yöntemleri kullanılarak arkaplan çıkarma işlemi yapıldıktan sonra elde edilen resimde hareketli nesne bilgisi bulunur. Elde edilen bu resim gri seviyeli bir resimdir. Gri seviyeli bu resim içerisindeki hareketli nesnelere ayırt etmek için nesnelerin ayrıştırılması gerekir. Bu, genellikle eşikleme işlemi ile yapılır. Çalışmada önerilen yatay ve dikey kesit yönteminde ise resmin 2.1. bağıntısına göre kesitleri alınarak eşiklenmektedir [27].

$$Y_i = \sum_{k=1}^E I_{i,k} \quad D_j = \sum_{k=1}^B I_{k,j} \quad (2.1)$$

Burada Y ve D sırasıyla yatay ve dikey kesiti, E resmin enini, B resmin boyunu, I_{ij} resmin i. satır ve j. sütundaki pikselinin parlaklık değerini ifade eder.



Şekil 2.10. Yatay ve dikey kesit değeri ile nesne algılanmasına bir örnek a) Arkaplan çıkarma yapıldıktan sonraki gri seviye resim b) Resmin yatay kesit değerleri c) Resmin dikey kesit değerleri d) Eşiklenmiş yatay ve dikey kesit işaretlerinin kesişimi

Hesaplanan yatay ve dikey kesit değerlerinde istenmeyen bölgeler de olacaktır. Arkaplan çıkarma yapılmış resim içerisinde gürültü ve aydınlatma değişimleri gibi etkenlerden dolayı bazı kısımlarda istenmeyen bilgiler olacaktır. Bunu engellemek için yatay ve dikey kesit değerleri bir eşikleme işleminden geçirilir. Daha sonra bu değerler (2.2) bağıntısındaki gibi kesiştirilerek hareketli bölge ayırt edilmiş olur(Şekil 2.10.a).

$$H_{i,j} = \begin{cases} 1, & D_j = 1 \text{ ve } Y_i = 1 \\ 0, & \text{aksi halde} \end{cases} \quad (2.2)$$

$H_{i,j}$, algılanan nesnenin (i,j) koordinatındaki değeridir.

Nesne algılama için kullanılan basit yöntemler, kullanılan tünel görüntülerindeki farklı tipteki problemler nedeniyle her zaman doğru sonuç veremez. Bu nedenle tünellerdeki araç takip işlemi zorlaşmaktadır. Tünellerde nesne algılama sırasında ortaya çıkabilecek problemler aşağıdaki şekilde sıralanabilir:

- Girişe ve çıkışa yakın kameralar için gün içinde değişen ışık seviyesi,
- Girişe yakın kameralar için tünele giren araçların görüntüdeki ışık seviyesini ani olarak değiştirmesi,
- Araç ışıklarının tünel yüzeyine yansması ve bunların da nesne gibi algılanması,
- Alınan görüntülerdeki gürültü,
- Tünel içindeki kameraların, başka amaçlarla kullanılan güvenlik kameralarına göre yeryüzüne daha küçük bir açıyla bakması,
- 3 boyutlu dünyadan 2 boyutlu resim düzlemine izdüşüm yapılırken kaybedilen bilgi.

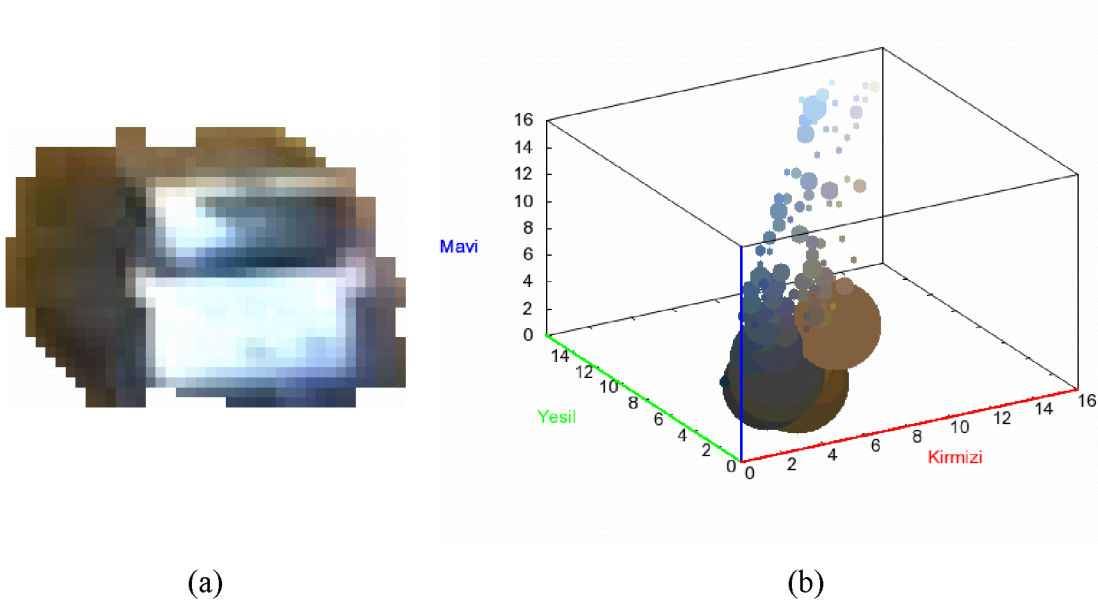
2.4. Nesne Takibi

2.4.1. Ortalama Kaydırma Algoritmasının Uygulanması

Ortalama kaydırma algoritması sadece takip işlemi için kullanılmayan, genel bir veri analizi yaklaşımıdır. Ortalama kaydırma ile takip işleminde, takip edilecek cismin renk dağılımı kullanılarak sonraki çerçevelerde bu bölge yakalanmaya çalışılır. Ortalama kaydırma algoritması takip edilecek cismin başlatılmasını gerektirir. Başlatılma işlemi

genellikle takip edilecek cismin resim içerisinde kapladığı bölgenin seçilmesi şeklindedir. Seçilen bölgenin daha sonraki çerçevelerdeki konumunu belirlemek için ilk olarak renk histogramı çıkartılır. Renk histogramı RGB, HSV ve diğer renk uzayları için hesaplanabilir. Örneğin; bilgisayarla görme alanındaki birçok işlemi kolaylaştıran OpenCV kütüphanesinde HSV uzayının H kanalı için bir histogram hesaplaması yapılmıştır [50].

Takip edilecek cismin sonraki çerçevelerdeki yeri, başlatma anındaki konum ve renk histogramı bilgisi kullanılarak bulunur. Problem için RGB renk uzayındaki histogram hesaplamasını ele alınsın. Renk uzayının eksenleri 16 eşit parçaya bölünür. Bununla birlikte $16^3 = 4096$ hücresi olan bir küp elde edilir. Elde edilen küpün her bir hücresi o renk aralığına düşen piksel değerlerinin toplamını ifade eder. Mevcut nesne algılama yöntemleri ile elde edilen nesnelere için örnek bir histogram değeri Şekil 2.11. 'deki gibidir.



Şekil 2.11. Örnek histogram dağılımı a) Histogramı oluşturulan araç görüntüsü b) RGB uzayındaki renk dağılımı(histogram)

Şekil 2.11. 'de dairelerin büyüklüğü, o aralığa düşen piksellerin toplamı ile orantılıdır. Aynı zamanda daireler bulunduğu aralığın rengini temsil edecek şekilde boyanmıştır.

Ortalama kaydırma algoritmasında nesnelere takip etmek için, takip edilecek nesnenin renk histogramı kullanılarak sonraki çerçeveler birer olasılık resmine dönüştürülür. Bu olasılık resmindeki piksellerin değeri, o pikselin renk histogramındaki

olasılık değeridir. Böylelikle olasılık resminde bulunan ve renk histogramında baskın olan renkleri barındıran pikseller daha belirgin olacaktır.

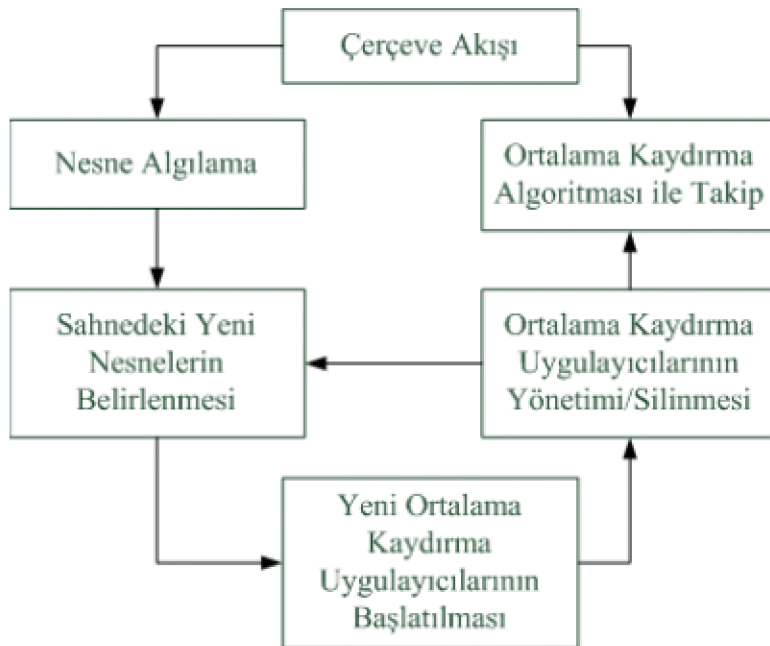
Elimizde bulunan tünel görüntülerini almakta kullanılan kameralar, tünelin içerisine yol düzlemi ile arasında $10^\circ - 15^\circ$ açı yapacak şekilde yerleştirilmiştir. Görüntülerde 3 boyutlu dünyadan 2 boyutlu kamera düzlemine izdüşüm yapılırken büyük miktarda bilgi kaybı yaşanmaktadır. Bunun yanı sıra araçlar görüntü düzlemine girerken çok büyük, kameradan uzaklaştıkça çok küçük alanlar kaplamaktadır. Dolayısıyla ortalama kaydırma algoritmasındaki başlangıç bölgesinin giderek küçülen bir alana sahip olması gerekmektedir. Aksi durumda ortalama kaydırma algoritması doğruluğunu kaybedecektir.

Ortalama kaydırma algoritması tekil nesnelere için çalışan bir algoritmadır. Aynı anda birden fazla nesnenin takibi mümkün değildir. Ele alınan problemde ise aynı anda birden fazla aracın takip edilmesi söz konusudur.

Bu bilgiler doğrultusunda geliştirilen takip sisteminin şu şartları sağlaması gerekmektedir:

- Çoklu nesne takibi,
- Ortalama kaydırma uygulayıcısının alanının giderek küçülmesi,
- Otomatik başlatma.

Bu şartları sağlaması amacıyla Şekil 2.12. 'deki gibi bir sistem önerilmiştir.



Şekil 2.12. Ortalama kaydırma algoritması ile çoklu nesne takibini gerçekleştirmesi için önerilen sistem yapısı

Önerilen sistemde, daha önceki kısımlarda anlatılan nesne algılama teknikleri ile görüntü içerisindeki hareketli nesnelerin algılanması hedeflenmiştir. Algılanan farklı nesnelerin takibi amacıyla sistemde birden fazla ortalama kaydırma uygulayıcısı bulunur. Bu ortalama kaydırma uygulayıcılarının her biri farklı bir nesne ile ilişkilendirilir ve sistemin başlangıç durumunda sayıları sıfırdır. Görüntü içerisinde nesneler algılandıkça ilk olarak bunun takip edilen mevcut bir nesne olup olmadığına bakılır. Bu işlem için mevcut takip edilen nesneler ile yeni çerçevelerde algılanan nesnelerin konumları karşılaştırılır. Yeni olduğuna karar verilen nesneler için yeni ortalama kaydırma uygulayıcıları başlatılır. Bu sayede sahneye yeni giren cisimler belirlenmiş olur. Sahneden çıkan nesnelerin belirlenmesi içinse yine konum bilgileri değerlendirilebilir. Her yeni gelen çerçevede bu işlemler yapılarak ortalama kaydırma algoritması ile çoklu nesne takibi gerçekleştirilmiş olur.

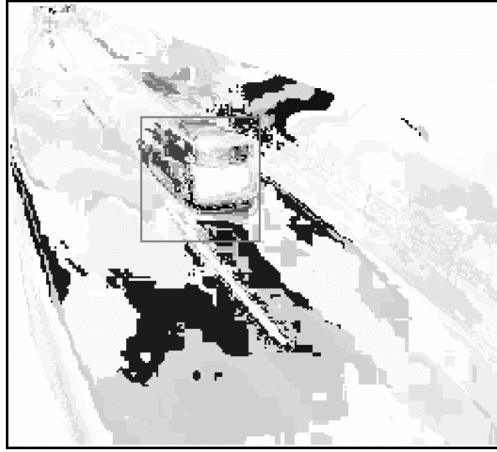
Önerilen sistem iyi gözükse de uygulamada bazı problemler ortaya çıkmaktadır. Şekil 2.13. 'e bakıldığında tek bir araç için üretilen olasılık resminin, aracın bulunduğu bölge haricindeki yerlerde de parlak pikseller olduğu gözlenmektedir. Bunun nedeni nesne algılama algoritmasının aracı tüm ana hatlarıyla algılayamamış olmasıdır. Algılanan bölge içerisinde bulunan yol renk histogramına dahil olacaktır. Geri izdüşüm yapılan olasılık resminde de yol bölgeleri parlak olacaktır.

Nesne algılama ve olasılık resimlerinin baskıda daha düzgün çıkması için renkleri ters çevrilmiştir. Şekil 2.13. ve 2.14. 'deki resimler için kullanılan parlak piksel tabiri o pikselin siyah renge daha yakın olduğunu ifade eder.



(a)

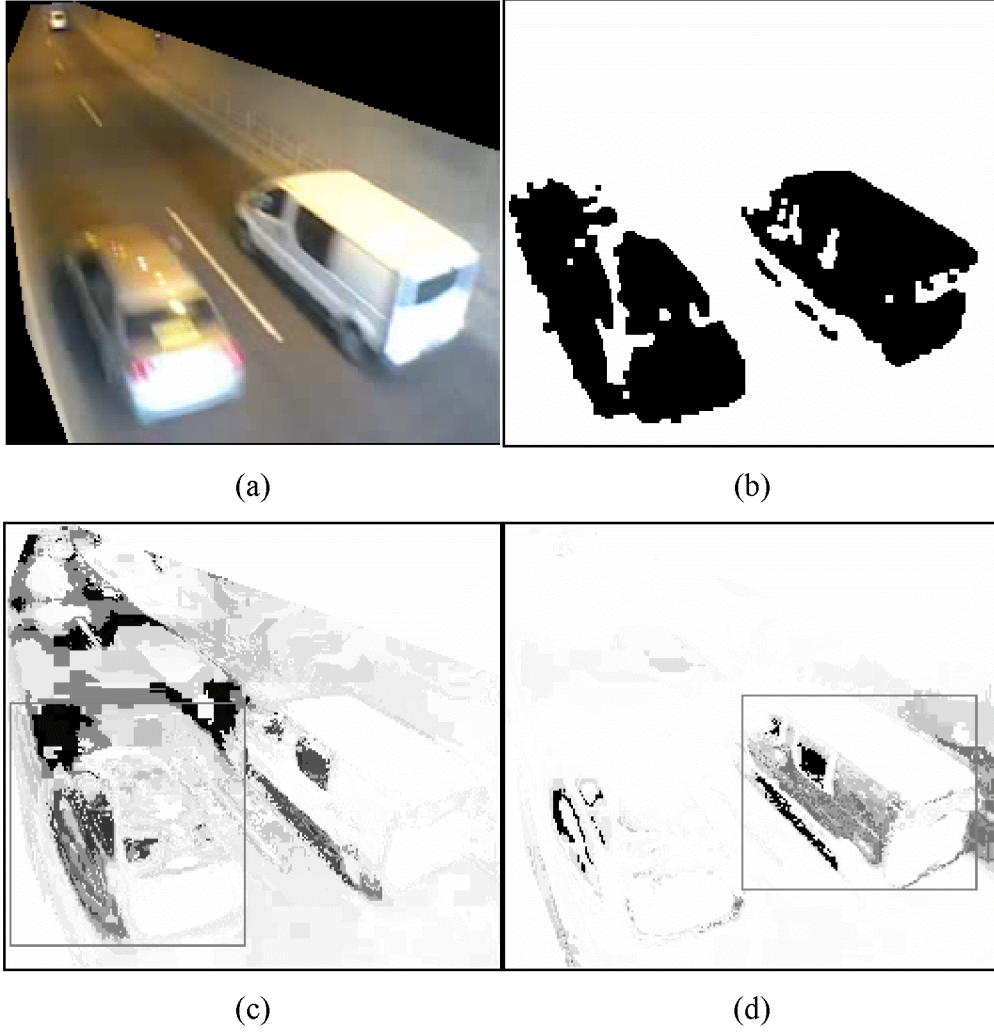
(b)



(c)

Şekil 2.13. Tek araç için oluşturulan ortalama kaydırma olasılık resmi a) Tünel görüntüsü b) Algılanan hareketli nesne c) Algılanan nesne bölgesi için oluşturulan olasılık resmi

Bir başka problem ise yan yana hareket eden araçlarda ortaya çıkmaktadır. Şekil 2.14. 'de yan yana hareket eden iki araç için oluşturulmuş olasılık resimleri verilmiştir. Görüldüğü üzere Şekil 2.14. 'de soldaki aracın olasılık resminde, sağdaki aracın bulunduğu bölgede parlak pikseller mevcuttur. Benzer şekilde sağdaki araç için oluşturulan olasılık resminde de soldaki aracın bulunduğu bölgede parlak pikseller bulunmaktadır. Bu durumda ortalama kaydırma algoritmasının gerçekleştirdiği takipte, araçların yörüngelerinden sapsması muhtemeldir. Yapılan bu değerlendirmeler ışığında ortalama kaydırma algoritmasının ele alınan problemi çözmekte yetersiz kaldığı görülmektedir.



Şekil 2.14. İki araç için oluşturulan ortalama kaydırma olasılık resimleri a) Tünel görüntüsü b) Algılanan nesnelere c) Soldaki araç için oluşturulan olasılık resmi d) Sağdaki araç için oluşturulan olasılık resmi

2.4.2. Görüntüdeki Yol Bölgesinin Poligonlar ile Seçilmesi

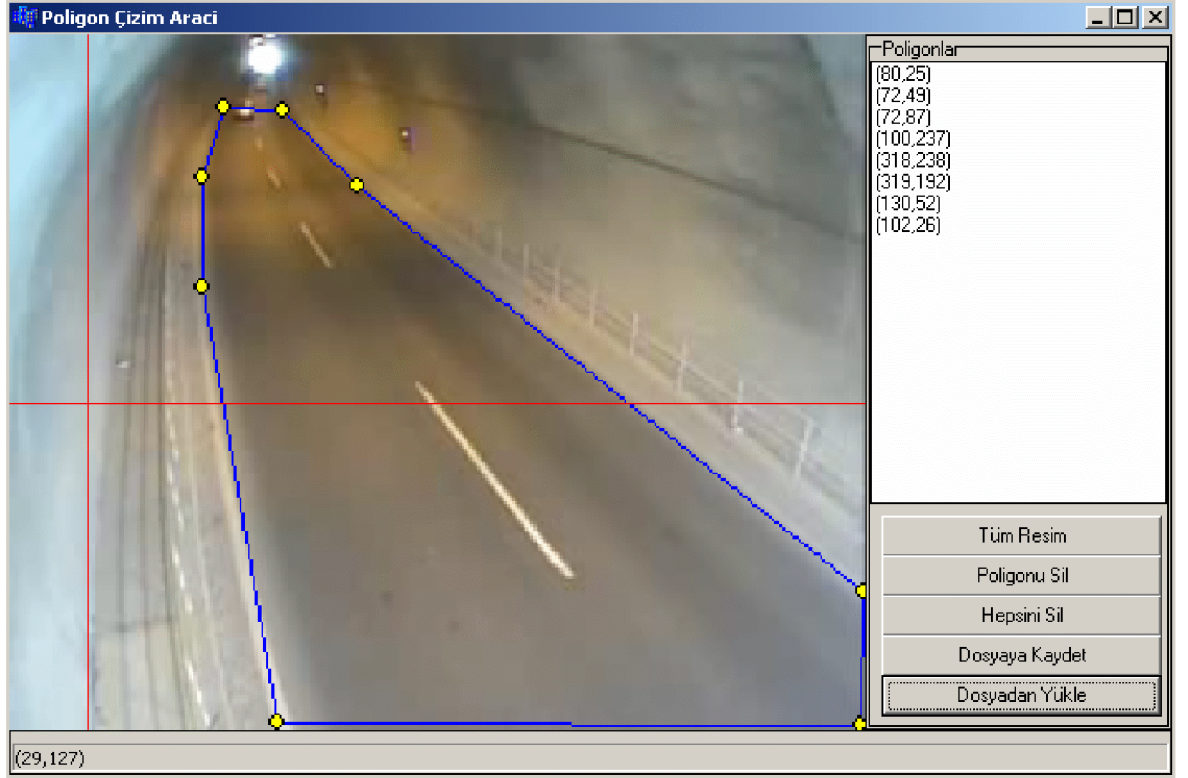
Kaydedilen tünel görüntülerinde problemi çözmekte kullanılmayacak ya da işe yaramayacak bölgeler bulunmaktadır. Örneğin; araçlar sadece yol üzerinde hareket edebileceği için araç takibi esnasında tünel duvarlarının bulunduğu bölgeler kullanılmayacaktır. Bu bölgelerin görüntüden çıkarılıp işlemlerin devam ettirilmesi programa hız kazandıracaktır. Bunun yanı sıra görüntünün gereksiz kısımlarında oluşan hatalı algılamalar araç takibini olumsuz yönde etkileyecektir. Bu nedenle görüntüdeki gereksiz bölgeler kaldırılıp yolun bulunduğu kısım üzerinde işlemler gerçekleştirilmiştir.

Seçme işlemi, seçilen bölgenin görüntünün diğer kısımlarından ayırt edilmesini sağlar. Yol bölgesi her zaman basit bir geometrik şekille ifade edilemeyeceğinden, seçme işlemi poligonlar kullanılarak yapılmıştır.

Nesne algılama ve nesne takibi işlemleri başlangıçta belirlenen bu seçim bölgesi üzerinde yapılmaktadır. Aşağıdaki kaynak kod bloğu ile bir noktanın verilen poligon noktaları içinde kalıp kalmadığı tespit edilir.

```
bool InPolygon(int xt,int yt,vector<TPoint> poly)
{
    unsigned int xnew,ynew;
    unsigned int xold,yold;
    unsigned int x1,y1;
    unsigned int x2,y2;
    int i;
    int inside=0;
    int npoints=poly.size();
    if (npoints < 3) {
        return false;
    }
    xold=poly[npoints-1].x;
    yold=poly[npoints-1].y;
    for (i=0 ; i < npoints ; i++) {
        xnew=poly[i].x;
        ynew=poly[i].y;
        if (xnew > xold) {
            x1=xold;
            x2=xnew;
            y1=yold;
            y2=ynew;
        }
        else {
            x1=xnew;
            x2=xold;
            y1=ynew;
            y2=yold;
        }
        if ((xnew < xt) == (xt <= xold)&&
            ((long)yt-(long)y1)*(long)(x2-x1) < ((long)y2-(long)y1)*(long)(xt-x1))
        ) {
            inside=!inside;
        }
        xold=xnew;
        yold=ynew;
    }
    return inside==1;
}
```

Görüntüdeki bütün pikseller için ilgili fonksiyon çağırıldığında verilen poligonların belirttiği alan seçilmiş olur. Seçme işlemi gerçekleştirilmek için Şekil 2.15. 'te ekran görüntüsü verilen uygulama gerçekleştirilmiştir.



Şekil 2.15. Poligonları kullanarak yol seçmek için geliştirilen uygulamanın ekran görüntüsü

2.4.3. Kalman Filtresinin Probleme Uygun Modellenmesi

Kalman filtresi probleme uygun modellenirken hareket eden araçların konum bilgisi için P_x ve P_y gibi iki değişken kullanılır. Araçların 3. Boyuttaki hareketini kamera bilgisinden elde edemeyeceğimiz için sadece iki boyut kullanılmıştır. Aynı zamanda bu aracın hızını belirten v_x ve v_y gibi iki değişkenimiz vardır. Bu değişkenlerle birlikte Kalman filtresinin durum vektörü aşağıdaki gibi olur.

$$x_k = \begin{bmatrix} P_x \\ P_y \\ v_x \\ v_y \end{bmatrix} \quad (2.3)$$

Sistemin belirli bir anındaki geçmiş durum ile mevcut durum arasındaki ilişkiyi belirlemek için bağıntı (1.56)'daki A geçiş matrisi aşağıdaki gibi tanımlanır.

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

dt değeri sistemde yapılan iki ölçüm arasındaki zaman farkını ifade eder. Ölçüm zaman aralıkları sabit ise dt değerini 1 almak bir problem oluşturmaz. Kontrol girişleri ve gürültüyü ihmal edersek sistemin k anındaki durumu aşağıdaki bağıntıdır.

$$x_k = Ax_{k-1} \quad (2.5)$$

Matris çarpımı gerçekleşince x_k durumu aşağıdaki gibi olur.

$$x_k = Ax_{k-1} = \begin{bmatrix} P_x^k \\ P_y^k \\ v_x^k \\ v_y^k \end{bmatrix} = \begin{bmatrix} P_x^{k-1} + v_x^{k-1} dt \\ P_y^{k-1} + v_y^{k-1} dt \\ v_x^{k-1} \\ v_y^{k-1} \end{bmatrix} \quad (2.6)$$

x_k , k anındaki tahmin işlemini belirtir. Kameradan aldığımız gerçek ölçüm değerleri(z_k) ise iki boyutlu bir vektördür ve cismin x ve y boyutundaki konum bilgisini içerir.

$$z_k = \begin{bmatrix} z_x \\ z_y \end{bmatrix}_k \quad (2.7)$$

Bağıntı (1.57)'deki sistemin durum vektörü x_k ile ölçüm vektörü z_k arasındaki ilişkiyi tanımlayan H geçiş matrisi aşağıdaki gibi tanımlanır.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.8)$$

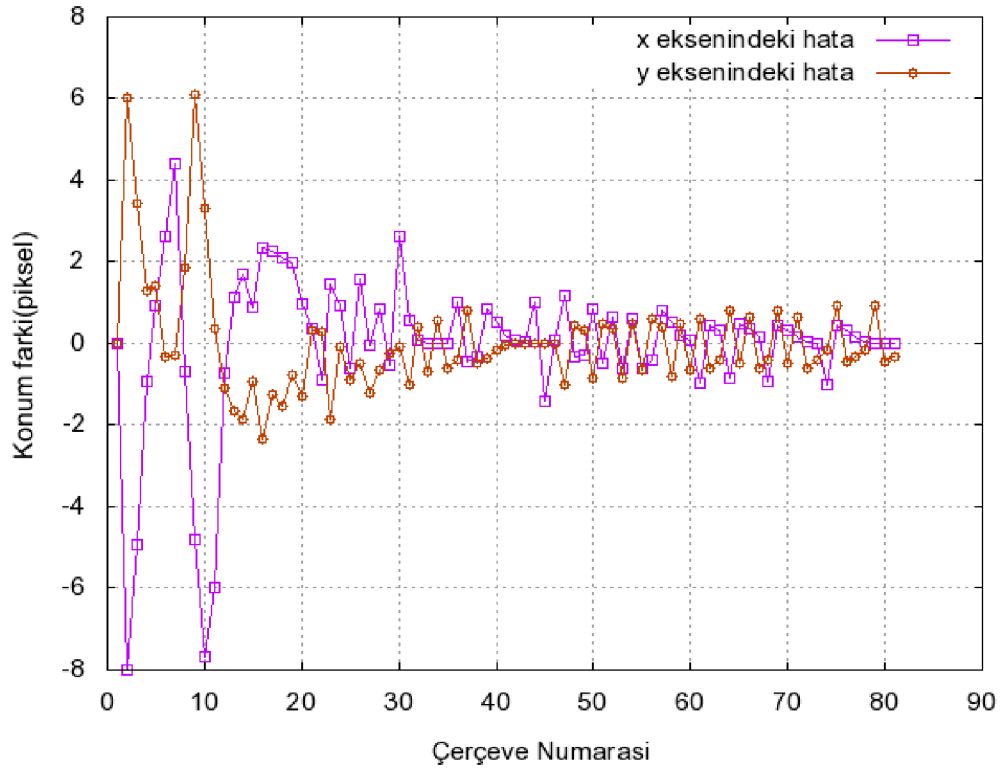
Verilen bu matrisler ışığında tahmin edilen durum ile ölçüm değerleri arasındaki hata aşağıdaki bağıntıdaki gibidir.

$$z_k - H\hat{x}_k^- = \begin{bmatrix} z_x^k - \hat{P}_x^k \\ z_y^k - \hat{P}_y^k \end{bmatrix} \quad (2.9)$$

2.4.4. Kalman Filtresinin Araç Verilerine Uygulanması

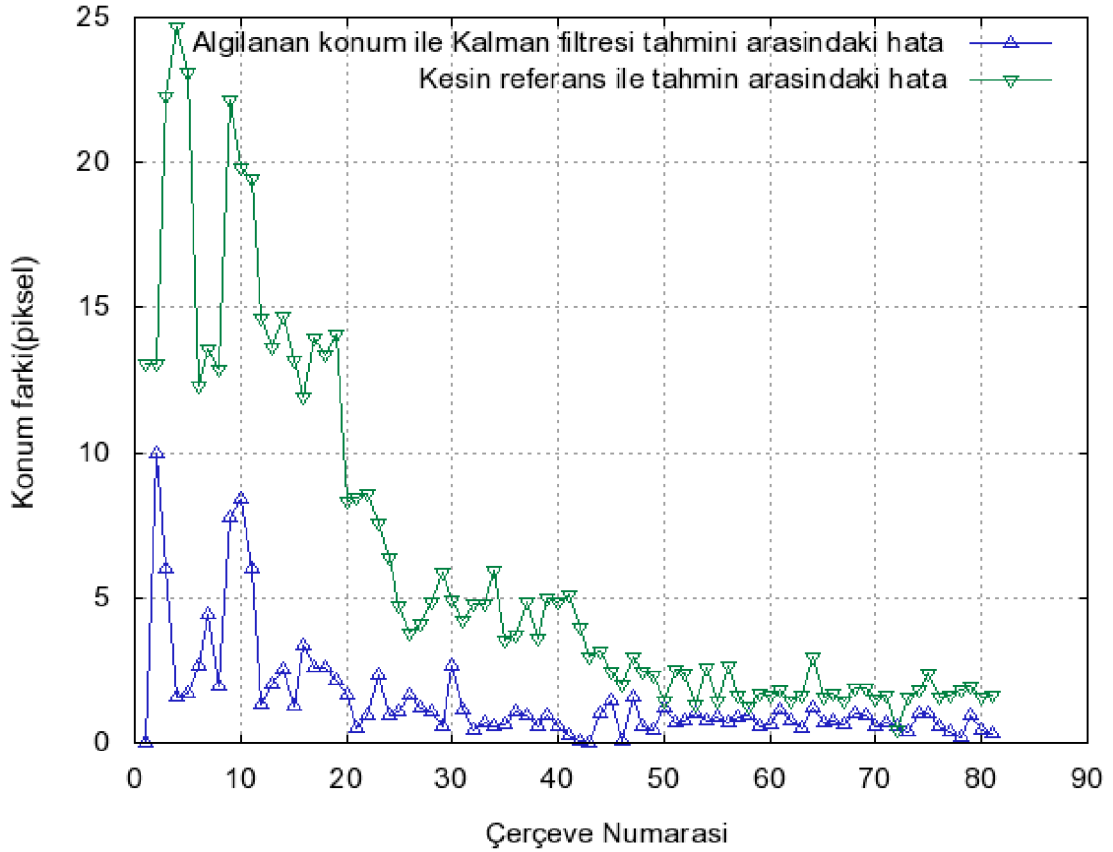
Nesne algılama yöntemleri kullanılarak tünel görüntülerindeki araçların iki boyutlu kamera düzlemindeki konum bilgileri elde edilir. Kalman filtresi kullanılarak gürültü içeren konum bilgilerinin gürültüden arındırılması gerçekleştirilir. Kalman filtresinin ürettiği sonuçların doğruluğunu değerlendirmek için araçların konum bilgilerinin kesin referansına ihtiyaç duyulur. Kesin referans bilgisi araçların gerçekte konumunun ne olduğunu belirtir ve nesne algılama yöntemlerinin bulunduğu konum bilgisinden farklı olabilmektedir. Kalman filtresinin doğruluğunu değerlendirmek amacıyla tünel görüntüsündeki bir aracın, nesne algılama yöntemleri kullanılarak konum bilgileri ve el ile işaretleme yaparak kesin referans bilgisi elde edilmiştir. Bu bilgiler tüneldeki bir aracın, tünele girişi ve çıkışı arasındaki 81 çerçevelik zaman aralığı içerisinde (Görüntüler 25 çerçeve/sn 'de kayıt edilmiştir).

Aracın konum bilgileri elde edilmeden önce Kalman filtresi ilk olarak durum kestirimi yaparak aracın iki boyutlu kamera düzlemindeki konumunu tahmin eder. Nesne algılama yöntemlerinden konum bilgisi geldiği anda Kalman filtresi bu veriyi kullanarak düzeltme yapar. Yapılan bu işlemler araç sahneden (görüntü düzlemi) çıkana kadar devam eder. Şekil 2.16. 'de bir araç için nesne algılama yöntemlerinin belirlediği konumlar ile Kalman filtresinin tahmin ettiği konumlar arasındaki hata değerleri bulunmaktadır.



Şekil 2.16. Örnek araç verisi için Kalman filtresinin tahmini ile nesne algılama yönteminin belirlediği konum arasındaki hata değerleri

Şekil 2.17. 'de ise aracın hareketi ile ilgili iki farklı hata grafiği bulunmaktadır. Bunlardan biri nesne algılama yöntemi ile Kalman filtresinin tahmini arasındaki uzaklık değerleri, diğeri ise el ile işaretlenen kesin referans ile Kalman filtresinin tahmini arasındaki uzaklık değeridir.



Şekil 2.17. Örnek araç verisi için Kalman filtresinin tahmini ile nesne algılama yönteminin belirlediği konumlar ve kesin referans bilgisi arasındaki hata değerleri

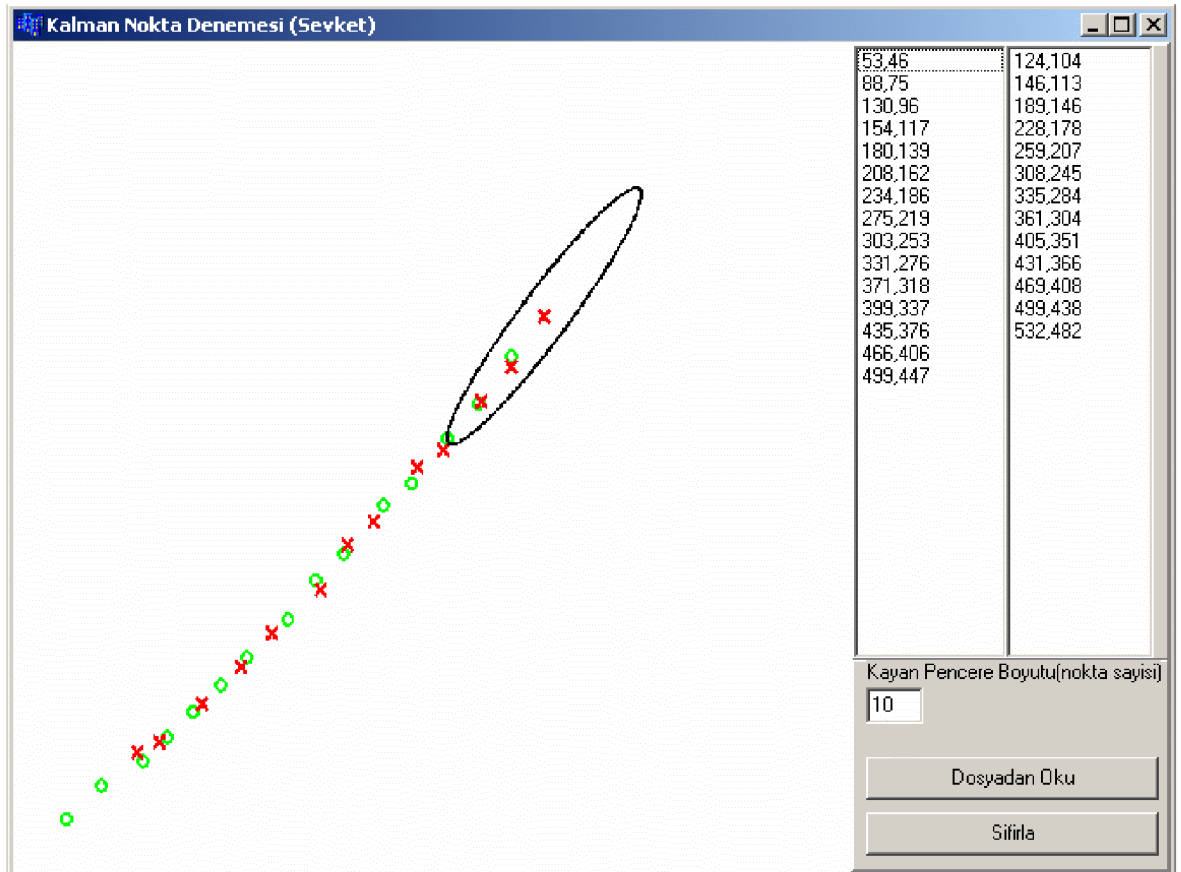
Şekil 2.16. ve Şekil 2.17. 'de görüldüğü üzere Kalman filtresinin yaptığı tahmin giderek gerçek sonuçlara yaklaşmaktadır. Hata değerinin azalmasında kamera düzlemine yapılan perspektif izdüşümün de etkisi vardır. Perspektif izdüşümün sonucu olarak araçlar kameradan uzaklaştıkça görüntü düzlemindeki konum değişimleri daha küçük olur.

2.4.5. Temel Bileşenler ile Güzergâh Aralığının Belirlenmesi

Nesneler takip edilirken güzergâh aralığını belirlemek, nesnenin çerçevelerde nesne – gözlem sonucu ilişkilendirmesini yaparken görüntü içindeki başka nesnelere veya parazit algılamalar ile ilişkilendirilmesini büyük oranda engelleyecektir. Güzergâh aralığı hata değerinin ortalaması alınarak basit bir şekilde hesaplanabilir. Bu durumda güzergâh aralığı merkezi Kalman filtresinin tahmin ettiği nokta olan dairesel bir alandır.

Güzergâh aralığı Kalman filtresinin tahmin ettiği noktalar ile nesne algılama yöntemlerinin belirlediği nokta arasındaki fark olan hata değerlerinin dağılımını yansıtacak şekilde hesaplanabilir. Bunun için hata değerlerinin temel bileşenlerini bulmak yeterli olacaktır. Hata değerlerinin özdeğer ve özvektörleri, hatanın temel bileşenlerini temsil eder.

Hata değerleri iki boyutlu vektörler olduğundan ikişer tane özdeğer ve özvektör olacaktır. Bu özdeğer ve özvektörler kullanılarak Şekil 2.17. 'deki gibi eliptik bir güzergâh aralığı belirlenir. Büyük özdeğerin temsil ettiği özvektör elipsin uzun eksenini temsil eder. Özvektörler birbirlerine dik oldukları için küçük özdeğerin temsil ettiği özvektör elipsin kısa eksenini olarak kullanılabilir.



Şekil 2.18. Temel bileşenler ile güzergâh aralığını belirlemek için geliştirilen uygulamanın ekran görüntüsü

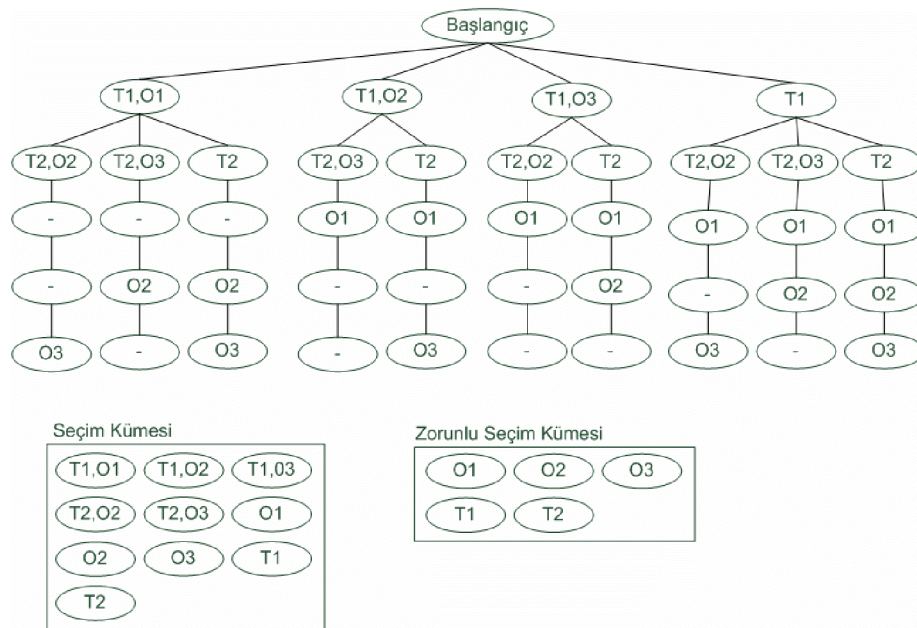
Şekil 2.18. 'de Kalman filtresi ile güzergâh tahmini yapmak ve temel bileşenler yardımıyla güzergâh aralığını belirlemek için geliştirilen uygulama görülmektedir. Görüntü içerisindeki küçük daireler ekranda tıklanan konumu, 'x' ile işaretlenmiş bölgeler ise

Kalman filtresinin sonraki adımda tahmin ettiği konumu belirtmektedir. Çizilen eliptik cisim de son adımdaki, merkezi Kalman filtresinin tahmin ettiği nokta olan güzergâh aralığını belirtir.

2.4.6. Çoklu Hipotez Takibinde Güzergâh-Gözlem Sonucu İlişkilendirmesi

Takip işleminin herhangi bir anında güzergâh aralıklarında ortak gözlem sonucu bulunduran güzergâhlar için veri ilişkilendirmesi çözülmesi gereken bir problemdir. Belirlenen hipotezler mevcut güzergâh ve gözlem sonuçlarının hepsini veya bunların ilişkilendirilmiş olduğu yeni güzergâhları içermesi gerekir. Şekil 1.7. 'deki ilişkilendirme problemi örnek alınırsa T_1 , T_2 güzergâhları ve O_1 , O_2 , O_3 gözlem sonuçları veya bunların ilişkilendirilmiş halleri bütün hipotezlerin içerisinde bulunması gerekir.

Mevcut güzergâhların, gözlem sonuçlarının ve bunlar arasındaki mümkün ilişkilendirmeleri barındıran küme, seçim kümesi olarak isimlendirilir. Şekil 1.7. 'deki problem için seçim kümesi Şekil 2.19. 'da verilmiştir. Hipotezlerin bulundurma gereken güzergâhlar ve gözlem sonuçları zorunlu seçimler kümesi olarak adlandırılır. Hipotezler, seçim kümesi ve zorunlu seçimler kümesi kullanılarak özyinelemeli bir şekilde Şekil 2.19. 'daki gibi hesaplanabilir.



Şekil 2.19. Örnek ilişkilendirme problemi için oluşturulan hipotezlerin ağaç yapısında gösterimi

Şekil 2.19. ‘daki ağacın en üst dalından tabanına kadar olan seçimlerin bütünü hipotezleri oluşturur. Ağaçta gözüken O_1 , O_2 , O_3 dalları yeni güzergâhları ifade eder. İlişkilendirme işlevini gerçekleştirmek için aşağıdaki özyinelemeli kod bloğu geliştirilmiştir.

```

void MHT::HipotezOlusturma (vector<Eslesme*> Secimler, vector<Eslesme*>
Eslesmeler, vector<void *> ZorunluSecimler, void * SecilecekEleman)
{
if (ZorunluSecimler.size() == 0)
{
for (int i=0; i<Secimler.size(); i++)
{
printf ("%s\t, ", Secimler[i]->toString());
}
printf ("\n");
return;
}
for (int i=0; i<Secimler.size(); i++)
{
if (Secimler[i]->track == SecilecekEleman ||
Secimler[i]->obs == SecilecekEleman)
{
for (int j=0; j<ZorunluSecimler.size(); j++)
{
if (ZorunluSecimler[j] == SecilecekEleman)
{
ZorunluSecimler.erase (&ZorunluSecimler[j]);
break;
}
}
HipotezOlusturma (Secimler, Eslesmeler, ZorunluSecimler, ZorunluSecimler[0]);
ZorunluSecimler.insert (ZorunluSecimler.begin(), SecilecekEleman);
break;
}
}
for (int i=0; i<Eslesmeler.size(); i++)
{
if (Eslesmeler[i]->track == SecilecekEleman ||
Eslesmeler[i]->obs == SecilecekEleman)
{
vector<Eslesme*> TempEslesme = Eslesmeler;
for (int j=0; j<TempEslesme.size(); j++)
{
if (TempEslesme[j]->track == Eslesmeler[i]->track &&
Eslesmeler[i]->track != NULL ||
TempEslesme[j]->obs == Eslesmeler[i]->obs &&
Eslesmeler[i]->obs != NULL)
{
TempEslesme.erase (&TempEslesme[j--]);
}
}
for (int j=0; j<ZorunluSecimler.size(); j++)
if (ZorunluSecimler[j] == SecilecekEleman)
ZorunluSecimler.erase (&ZorunluSecimler[j]);
Secimler.push_back (Eslesmeler[i]);
HipotezOlusturma (Secimler, TempEslesme, ZorunluSecimler, ZorunluSecimler[0]);
}
}
}

```

```
        Secimler.pop_back();  
        ZorunluSecimler.insert(ZorunluSecimler.begin(),SecilecekEleman);  
    }  
}  
}
```

3. SONUÇLAR

Günümüzde yoğunluğun giderek arttığı karayollarında güvenli ulaşım büyük önem arz etmektedir. Çalışmada karayolu tünellerindeki ulaşımın daha güvenli hale getirilmesi amacıyla bilgisayar kontrollü sistemler üzerinde durulmuştur. Tünel içerisindeki araçların hareketinin analizi için ilk olarak nesne algılama yöntemleri ile hareketli araçlar algılanmıştır.

Çeşitli nesne algılama yöntemleri tünel görüntülerine uygulanarak sonuçları değerlendirilmiştir. Basit nesne algılama yöntemleri görüntüleri hızlı bir şekilde işlemesine rağmen iyi performans sağlamamaktadır. Karmaşık nesne algılama yöntemleri daha doğru sonuçlar üretmesine rağmen gerçek zamanda işlem yapmaları zordur. Araçların karmaşık şekillerinin algılamadaki olumsuz etkisini gidermek için yatay-dikey kesit yöntemi önerilmiştir. Bu yöntemle hareketli nesnelere dikdörtgen bloklar halinde algılanabilmektedir.

Uygulanan nesne algılama yöntemleri kalitesiz tünel görüntüleri üzerinde her zaman doğru sonuç vermemektedir. Görüntülerdeki ani ışık değişimleri, araç ışıklarının tünelin duvarlarına ve yola vurması, yol rengine yakın renkteki araç kısımları, araç gölgeleri ve görüntülerde örtüşen araçlar nesne algılama yöntemlerinin hata yapmasına neden olmaktadır. Araçların hareket etmediği, yol haricindeki bölgelerde oluşan hataları gidermek için yol bölgesi seçilmiştir. Böylece sistem sadece yol bölgesi içinde oluşan hatalı algılamalar ile uğraşacak hale getirilmiştir.

Güçlü bir algoritma olduğu bilinen ortalama kaydırma algoritması görüntüler üzerinde denenmiştir. Tekil nesnelere için tasarlanan ortalama kaydırma algoritmasının tünel içinde hareket eden birden fazla aracı takip etmekte yetersiz kaldığı gözlenmiştir.

Tünel görüntülerinin alındığı kameralar tünel içerisinde 4,5 metre yükseklikte ve yol düzlemi ile arasındaki açı 10° - 15° olacak şekilde yerleştirilmiştir. Fiziksel kısıtlamalardan dolayı kameraları başka şekilde yerleştirmek mümkün değildir. Tünel içine bu şekilde yerleştirilen kameralar açık alanlara yerleştirilen gözetleme kameralarına göre daha kısıtlı bilgi üretmektedir. Tünelde hareket eden araçlar kameralardan uzaklaştıkça görüntü düzlemindeki konum değişimleri çok küçük olmaktadır. Bu durumda analiz yapmak zorlaşmaktadır.

Nesne takibi için Kalman filtresinin durum kestiriminden yararlanılmıştır. Kalman filtresinin yaptığı tahmin ile nesne algılama yöntemlerinin belirlediği konum arasındaki hata değerlerinin dağılımı ile güzergâh aralıkları belirlenmiştir. Daire şeklinde, öklit uzaklığına dayanan güzergâh aralıkları yerine hatanın dağılımını hesaba katan esnek yapıdaki eliptik güzergâh aralıkları kullanılmıştır. Kalman filtresinin tahminleri ile kesin referans bilgileri karşılaştırılarak tahminlerin gerçek araç konumlarına yaklaştığı gözlenmiştir.

Yapılan araştırmalarda literatürde nesne takibi üzerine çeşitli algoritmaların varlığı gözlemlenmiştir. Fakat tünellerde araç takibine dayalı bir güvenlik sistemine rastlanmamıştır.

Kullanılan algoritmaların kodunun yazımı esnasında görüntü işleme ve bilgisayarla görme alanında popüler olan OpenCV kütüphanesinin üst seviyeli fonksiyonlarından yararlanılmıştır.

Çalışma sırasında irdelenen nesne algılama algoritmaları ve tünellerde otomatik olay algılama ile ilgili olan 1 adet bildiri 2. Mühendislik ve Teknoloji Sempozyumu' na sunulmuş ve bildiri kitapçığında basılmıştır.

4. ÖNERİLER

1. Tüneldeki ışık yansımaları ve aydınlatmadaki ani değişiklikler gibi temel problemleri algılayan ve bu problemleri gideren algoritmalar geliştirilebilirse nesne algılama ve izleme daha başarılı olabilir.
2. Elde edilen izleme verileri çeşitli öğrenme yöntemleri ile eğitilerek olaylar hakkında daha doğru bir karar verme sistemi geliştirilebilir. Sisteme öğrenme yetisi kazandırılarak karar aşamasında doğruluk artırılabilir.
3. Nesne algılama performansını artırmak amacıyla birden fazla nesne algılama yöntemi birleştirilerek karma bir yöntem oluşturulabilir. İzleme aşamasından alınacak geri beslemelerle nesne algılama daha da iyileştirilebilir.
4. Nesneleri 2 boyutlu görüntü düzleminde takip etmek yerine 3 boyutlu gerçek dünya koordinatlarına ters izdüşüm yapılarak takip edilebilir. Bu işlem aracın hareketi konusunda daha sağlıklı bilgi verecektir. Ters izdüşüm için tek kamera kullanılabileceği gibi tünel içine yerleştirilen ve çift görüş sağlayan kameralar da kullanılabilir.
5. Nesne algılamadaki hataları gidermek amacıyla daha kaliteli görüntü sağlayan kameralar kullanılabilir.
6. Nesne takibinin performansını artırmak amacıyla Kalman filtresi yerine parçacık filtresi veya bu ikisinin birleşiminden oluşan karma bir filtre kullanılabilir.
7. Nesne algılama tekniklerinin hatalı ve eksik algılamasının nesne takibi üzerindeki olumsuz etkisini gidermek için çoklu hipotez takibi algoritmasının güçlü özellikleri kullanılabilir. Çoklu hipotez takibi kullanıldığında, nesne takibi algoritmalarında temel bir problem olan nesnelere sahneye girmesi ve çıkması otomatik olarak algılanacaktır.
8. Sistemin gerçek zamanda çalışması için günümüzde kullanımı gittikçe artan çok çekirdekli işlemci mimarilerinde paralel olarak çalışacak sistemler geliştirilebilir. Hatta birden fazla bilgisayarı paralel olarak verimli bir şekilde kullanacak sistemler geliştirilebilir.

5. KAYNAKLAR

1. Versavel, J., Road Safety Through Video Detection , International Conference on Intelligent Transportation Systems, 1999, Tokyo, Bildiriler Kitabı, 753-757.
2. Ikeda, H., Kaneko, Y., Matsuo, T. ve Tsuji, K., Abnormal Incident Detection System Employing Image Processing Technology , International Conference on Intelligent Transportation Systems, 1999, Tokyo, Bildiriler Kitabı, 748-752.
3. McLauchlan, P., Beymer, D., Coifman, B. ve Mali, J., A Real-time Computer Vision System for Measuring Traffic Parameters, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, San Juan, Bildiriler Kitabı, 495.
4. Yu, M., Jiang, G. ve Yu, B., An Integrative Method For Video Based Traffic Parameter Extraction In ITS, IEEE Asia-Pacific Conference on Circuits and Systems, Aralık 2000, Tianjin, Bildiriler Kitabı, 136-139.
5. Kimachi, M., Kanayama, K. ve Teramoto, K., Incident Prediction by Fuzzy Image Sequence Analysis, Vehicle Navigation and Information Systems Conference, Ağustos 1994, Yokohama, Bildiriler Kitabı, 51-56.
6. Veeraraghavan, H., Masoud, O. ve Papanikolopoulos, N., Computer Vision Algorithms for Intersection Monitoring , IEEE Transactions on Intelligent Transportation Systems, 4, 4 (2003) 78-89.
7. Moravec, H., Visual Mapping by a Robot Rover , International Joint Conference on Artificial Intelligence, Ağustos 1979 , Tokyo, Bildiriler Kitabı, 598-600.
8. Harris, C. ve Stephens, M., A Combined Corner and Edge Detector , Alvey vision conference, Ağustos 1988, Manchester, Bildiriler Kitabı, 50.
9. Shi, J. ve Tomasi, C., Good Features to Track , IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Haziran 1994, Wahington, Bildiriler Kitabı, 593-600.
10. Comaniciu, D., Meer, P., Mean Shift: A Robust Approach Toward Feature Space Analysis , IEEE Transactions on Pattern Analysis and Machine Intelligence , 24, 5 (2005) 603-619.
11. Rowley, H., Baluja, S. ve Kanade, T., Neural Network-based Face Detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, San Francisco, Bildiriler Kitabı, 203-208.
12. Viola, P., Jones, M. J. ve Snow, D., Detecting Pedestrians Using Patterns of Motion and Appearance , IEEE International Conference on Computer Vision, Ekim 2003, Nice, Bildiriler Kitabı, 734-741.

13. Grewe, L. ve Kak, A., Interactive Learning of a Multiple-attribute Hash Table Classifier for Fast Object Recognition , Computer Vision and Image Understanding , 61, 3 (1995) 387-416.
14. Papageorgiou, C., Oren, M. ve Poggio, T., A General Framework for Object Detection, International Conference on Computer Vision, Ocak 1998, Bombay, Bildiriler Kitabı, 555-562.
15. Freund, Y. ve Schapire, R., A Decision-theoretic Generalization of On-line Learning and an Application to Boosting , Journal of computer and system sciences, 55, 1 (1997) 119-139.
16. <http://www.boosting.org/> , Adaptif Hızlandırma, 5 Haziran 2009.
17. Oral, M. ve Deniz, U., Centre of Mass Model - A Novel Approach to Background Modelling for Segmentation of Moving Objects , Image and Vision Computing , 25 (2007) 1365-1376.
18. Ekinci, M. ve Gedikli, E., Background Estimation Based People Detection and Tracking for Video Surveillance , Lecture Notes in Computer Science, 2869 (2003) 421-429.
19. Oliver, N. M., Rosario, B. ve Pentland, A. P., A Bayesian Computer Vision System for Modeling Human Interactions , IEEE Transactions on Pattern Analysis and Machine Intelligence , 22 (2000) 831-843.
20. <http://www.sn1.salk.edu/~shlens/pub/notes/pca.pdf>, Tutorial on Principal Component Analysis, 3 Mayıs 2009.
21. Wren, C., Azarbayejani, A., Darrell, T. ve Pentland, A., Pfindex: Real-time Tracking of the Human Body , IEEE Transactions on Pattern Analysis and Machine Intelligence , 19 (1997) 780-785.
22. Gao, X., Boult, T., Coetzee, F. ve Ramesh, V., Error Analysis of Background Adaption , IEEE Conference on Computer Vision and Pattern Recognition 1, Haziran 2000, South Carolina, Bildiriler Kitabı, 503-510.
23. Stauffer, C. ve Grimson, W., Adaptive Background Mixture Models for Real-time Tracking , IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2, Haziran 1999, Colorado, Bildiriler Kitabı, 246-252.
24. KaewTraKulPong, P. ve Bowden, R., An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection , European Workshop Advanced Video Based Surveillance Systems, Eylül 2001, Kingston, Bildiriler Kitabı, 1-5.
25. Zivkovic, Z., Improved Adaptive Gaussian Mixture Model for Background Subtraction , International Conference on Pattern Recognition 2, Ağustos 2004, Cambridge, Bildiriler Kitabı, 28-31.

26. Yılmaz, A., Javed, O. ve Shah, M., Object Tracking: A Survey , ACM Computing Surveys, 38, 4 (2006) 13.
27. Çakır, Ş. U. ve Nabiyev, V. V., Tünellerde Güvenliğin Sağlanması Amacıyla Arkaplan Modelinin Belirlenmesi ve Araç Takibi , Mühendislik ve Teknoloji Sempozyumu 2, Nisan 2009, Ankara, Bildiriler Kitabı, 288-293.
28. Canny, J., A Computational Approach to Edge Detection , IEEE Transactions on Pattern Analysis and Machine Intelligence , 8, 1986 , 679-698.
29. Yayıncı, A. Ö., Temel Bileşenler Analizi İçin Robust Algoritmaları, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü Ankara, 2006.
30. Filiz Z., Güvenilirlik Çözümlemesi, Temel Bileşenler ve Faktör Çözümlemesi, Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, 4, 2 (2003) 211-222.
31. Sütçüler, E., Gerçek Zamanlı Video Görüntülerinden Yüz Bulma ve Tanıma Sistemi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü İstanbul, 2006.
32. Konak, E. S., Bilgisayar Destekli Yüz Tanıma Sistemi Tasarımı, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü İstanbul, 2006.
33. Alpar, R., Uygulamalı Çok Değişkenli İstatistiksel Yöntemlere Giriş 1, 2, Nobel Yayın Dağıtım, Ankara, 2003.
34. Aykut, M., Örüntü Tanıma Amaçlı Doğrusal Olmayan Yöntemlerin Geliştirilmesi ve Uygulanması, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Trabzon, 2007.
35. Kaliath, T., Sayed, A., H. ve Hassibi, B., Linear Estimation, 1, Prentice Hall, New Jersey, 2000.
36. Maybeck, P., S., Stochastic Models, Estimation, and Control, 3, Academic Press Inc., New York, 1982.
37. Brown, Y., G. ve Hwang, P., Y., C., Introduction to Random Signals and Applied Kalman Filtering, 3, Wiley, New York, 1996.
38. Schmidt, S., Application of State-space Methods to Navigation Problems, Advances in Control Systems, 3 (1966) 293-340.
39. Mahalanobis, P., C., On the Generalised Distance in Statistics, Proceedings of the National Institute of Sciences of India, 2(1), 1936, 49-55.
40. Maesschalck, R., D., Jouan-Rimbaud, D. ve Massart, D., L., The Mahalanobis Distance, Chemometrics and Intelligent Laboratory Systems, 50 (2000) 1-18.
41. Mahalanobis, P., C., Analysis of Race-mixture in Bengal, Journal of the Asiatic Society of Bengal, 23 (1927) 301-333.

42. McLachlan, G. J., *Discriminant Analysis and Statistical Pattern Recognition*, 1, John Wiley & Sons, New Jersey, 1992.
43. Blackman, S. ve Popoli, R., *Design and Analysis of Modern Tracking Systems*, 1, Artech House Publishers, Massachusetts, 1999.
44. Blackman, S. S., Multiple Hypothesis Tracking for Multiple Target Tracking, IEEE Aerospace and Electronic Systems Magazine, 19, 1 (2004) 5-18.
45. Nahi, N., Optimal Recursive Estimation with Uncertain Observation, IEEE Transactions on Information Theory, 15, 4 (1969) 457-462.
46. Singer, R., A. ve Stein, J., J., An Optimal Tracking Filter for Processing Sensor Data of Imprecisely Determined Origin in Surveillance Systems, *IEEE Conference on Decision and Control*, 1971 Florida, 10, 171-175.
47. Reid, D., B., An Algorithm for Tracking Multiple Targets, IEEE Transactions on Automatic Control, 24, 6 (1979) 843-854.
48. Sittler, R., W., An Optimal Data Association Problem in Surveillance Theory, IEEE Transactions on Military Electronics, 8, 2 (1964) 125-139.
49. Cox, I., J., A Review of Statistical Data Association Techniques for Motion Correspondence, IJCV, 10, 1 (1993) 53-66.
50. Bradski, G. ve Adrian, K., *Learning OpenCV*, 1, O'Reilly Media, California, 2008.

ÖZGEÇMİŞ

1982 yılında Antalya'da doğdu. İlköğrenimini Atatürk İlkokulu ve Faruk Tugayoğlu İlköğretim Okulu'nda, orta öğrenimini Aldemir-Atilla Konuk Süper Lisesi'nde tamamladı. 2001 yılında Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde lisans programına başladı ve 2005 yılında bu bölümden mezun oldu.

Kısa bir süre özel sektörde çalıştıktan sonra, 2006 yılında Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans programına başladı. 2007-2008 Eğitim-Öğretim yılında Socrates-Erasmus programı ile yüksek lisansının bir yılını Avusturya'daki, Technology University of Vienna üniversitesinde geçirdi. 2008-2009 yılları arasında TÜBİTAK destekli bir projede bursiyer öğrenci olarak çalışmıştır. Yabancı dil olarak iyi derecede İngilizce ve başlangıç seviyesinde Almanca bilmektedir.