T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DİJİTAL VİDEOLARDA ARKA PLAN MODELLEME ve HAREKETLİ NESNE ÇIKARIMI

K. Hakan Kutluay

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd.Doç.Dr. Nurşen SUÇSUZ

Edirne - 2008

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DİJİTAL VİDEOLARDA ARKA PLAN MODELLEME ve HAREKETLİ NESNE ÇIKARIMI

K. Hakan Kutluay
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı

Bu Tez 19/10/2008 Tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir:

Yrd.Doç.Dr. Nurşen SUÇSUZ
Danışman

Yrd.Doç.Dr. Aydın CARUS
Üye

Yrd.Doç.Dr. Nilgün TOSUN
Üye

ÖZET

Yüksek Lisans Tezi, Dijital Videolarda Arka Plan Modelleme ve Hareketli Nesne Çıkarımı, T.C. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu çalışmanın amacı, dijital videolarda arka plan modelleme yöntemlerinin araştırılması, bu modelleme yöntemlerinin hareketli nesne çıkarım yöntemlerinde kullanımının karşılaştırmalı olarak incelenmesidir. Bilgi teknolojilerinin gelişimi ile video dosyalarından içerik çıkarımında bilgisayarların kullanımı oldukça popüler bir konu haline gelmiştir.

Bilgisayarların dijital video dosyalarının saklanmasında kullanımı oldukça yaygın olmasına rağmen, dijital videolardan içerik çıkarımı konusunda kullanımı oldukça yenidir. Dijital video dosyalarından elde edilebilecek en temel veri, hareketli nesnelerdir.

Bu çalışma dijital video dosyalarındaki arka plan modelleme ve hareketli nesne çıkartım yöntemlerinin karşılaştırılması üzerine yapılmış özgün bir çalışmadır.

Bu tez 2008 yılında yapılmıştır ve 58 sayfadan oluşmaktadır.

ANAHTAR KELİMELER: Arka plan modelleme, Hareketli nesne çıkartımı, HRR, PFinder

ABSTRACT

Master of Science Thesis, Background Modelling and Motion Detection on Digital Videos, T.C. Trakya University, Institue of Natural Sciences, Department of Computer Engineering .

The purpose of this research is exploring the methods of background modelling, analyzing and comparing the use of modelling methods in the methods of background modelling on digital videos. Due to the development of Information Technologies, using computers in the extraction of contents from video files has become a quite popular topic.

Although computers are widely used for storing digital video files, the subject of content extracting is quite new. The most primary data that can be extracted from digital video files are moving objects.

This work is an orginal research based on comparing background modelling and motion detection on digital videos.

This thesis is done in 2008 and consist of 58 pages.

KEY WORDS: Background modelling, Motion detection, HRR, PFinder

İÇİNDEKİLER

ÖZET i
ABSTRACT ii
İÇİNDEKİLER iii
ŞEKİLLER TABLOSU v
1 GİRİŞ 1
2 DİJİTAL VİDEO 3
2.1 Video Dosya Formatları 3
2.1.1 AVI 4
2.1.2 MPEG 4
2.1.3 RealVideo 5
2.1.4 H.261 5
3 ARKA PLAN MODELLEME 6
3.1 Arka Plan Modelleme Yöntemleri 7
3.1.1 PFinder 9
3.1.2 HRR (En çok tekrarlanma oranı) 14
3.2 PFinder ve HRR Yöntemlerinin Karşılaştırması 19
3.3 Performans Değerlendirme 25
3.3.1 HRR performans ölçümleri 25
3.3.2 PFinder performans ölçümleri 26
4 HAREKETLİ NESNE ÇIKARIMI 28
4.1 Ardışık Kare Farkı Yöntemi ile Hareketli Nesne Çıkarımı 29
4.2 Arka Plan Modelleme Yöntemleri ile Hareketli Nesne Çıkarımı 31
4.2.1 HRR yöntemi kullanarak hareketli nesne çıkarımı 32
4.2.2 PFinder yöntemi kullanarak hareketli nesne çıkarımı 33
4.3 Hareketli Nesne Çıkarımı Yöntemlerinin Karşılaştırmaları 34
4.3.1 Ardışık kare farkı yöntemi ile hareketli nesne çıkarımının artı eksileri ve performans değerlendirme 37
4.3.2 PFinder arka plan modeli kullanılarak hareketli nesne çıkarımının artı eksileri ve performans değerlendirme 38
4.3.3 HRR arka plan modeli kullanılarak hareketli nesne çıkarımının artı eksileri ve performans değerlendirme 40
4.4 Hareketli Nesne Çıkarımında Kullanılan İyileştirme Teknikleri 43
4.4.1 Eşik filtresi 43
4.4.2. Morfolojik filtreler 44
4.4.1.1 Erozyon 46
4.4.1.2 Genişleme 49
4.4.1.3 Açma 51
4.4.1.4 Kapama 52

SONUÇ 53
SÖZLÜK 54
KAYNAKLAR 55
TEŞEKKÜR 57
ÖZGEÇMİŞ 58

ŞEKİLLER TABLOSU

Şekil 3.1 Test Videosunun 0 - 450 arası karelerinin 25 karede bir alınmış görüntüleri ... 8
Şekil 3.2 100. karede oluşan arka plan, α = 0.025 ... 10
Şekil 3.3 200. karede oluşan arka plan, α = 0.025 ... 11
Şekil 3.4 500. karede oluşan arka plan, α = 0.025 ... 11
Şekil 3.5 200. karede oluşan arka plan, α = 0.05 ... 12
Şekil 3.6 HRR Öğrenme süreci gösterimi ... 14
Şekil 3.7 50. karede oluşan arka plan ... 15
Şekil 3.8 100. karede oluşan arka plan ... 16
Şekil 3.9 200. karede oluşan arka plan ... 16
Şekil 3.10 100. karedeki görüntü ve arka plan modeli ... 19
Şekil 3.11 125. karedeki görüntü ve arka plan modeli ... 20
Şekil 3.12 150. karedeki görüntü ve arka plan modeli ... 20
Şekil 3.13 175. karedeki görüntü ve arka plan modeli ... 21
Şekil 3.14 100. karedeki görüntü ve arka plan modeli ... 21
Şekil 3.15 125. karedeki görüntü ve arka plan modeli ... 22
Şekil 3.16 150. karedeki görüntü ve arka plan modeli ... 22
Şekil 3.17 5.Karede oluşan arka plan modeli (α = 0.05) ... 23
Şekil 3.18 PFinder yöntemi ile oluşturulan arka plan görüntüsü ... 24
Şekil 3.19 HRR PFinder yöntemi ile oluşturulan arka plan görüntüsü ... 24
Şekil 3.20 HRR Öğrenim Grafiği ... 25
Şekil 3.21 PFinder Model Güncelleme Zaman/Kare grafiği ... 26
Şekil 4.1 8 bit örnek bir kare için kare farkı hesaplaması ... 29
Şekil 4.2 *t-1* anındaki video kare örneği ... 30
Şekil 4.3 *t* anındaki video kare örneği ... 30
Şekil 4.4 Oluşan fark görüntüsü *Dt* ... 30
Şekil 4.5 HRR yöntemi ile oluşan arka plan modelleri ... 32
Şekil 4.6 Hareket anında alınan örnek kareler ... 32
Şekil 4.7 Arka plan modeli ile kare arasında oluşan fark görüntüleri ... 33
Şekil 4.8 PFinder yöntemi ile oluşan arka plan modelleri ... 33
Şekil 4.9 Hareket anında alınan örnek kareler ... 34
Şekil 4.10 Arka plan modeli ile kare arasında oluşan fark görüntüleri ... 34
Şekil 4.11 Örnek kare ve hareketli nesne çıkarımları ... 35
Şekil 4.12 Örnek kare ve hareketli nesne çıkarımları ... 36
Şekil 4.13 Ardışık kare farkı ile elde edilen hareketli nesne çıkarımı ... 37
Şekil 4.14 Ardışık kare farkı yöntemi performans ölçümü ... 37
Şekil 4.15 PFinder yöntemi ile elde edilen hareketli nesne çıkarımı ... 38
Şekil 4.16 α = 0.025 alınarak hesaplanmış PFinder arka plan çıkarımı ... 39
Şekil 4.17 α = 0.050 alınarak hesaplanmış PFinder arka plan çıkarımı ... 39
Şekil 4.18 α = 0.100 alınarak hesaplanmış PFinder arka plan çıkarımı ... 39
Şekil 4.19 PFinder yöntemi ile performans ölçümü ... 40
Şekil 4.20 HRR arka plan modeli ... 41
Şekil 4.21 Videodan alınan kare ... 41

Şekil 4.22 HRR Hareketli nesne çıkarımı 41
Şekil 4.23 HRR Performans ölçümleri 42
Şekil 4.24 PFinder yöntemiyle oluşan hareketli nesne çıkarımındaki hatalar 43
Şekil 4.25 Hareketli nesne çıkarım sonuçlarının filtre uygulanmış halleri 44
Şekil 4.26 Sık kullanılan yapılandırma elemanları 45
Şekil 4.27 Örnek ikili görüntü 46
Şekil 4.28 Erozyon filtresi uygulanmış örnek görüntü 47
Şekil 4.29 HRR ile elde edilen hareketli nesne çıkarımı 48
Şekil 4.30 HRR ile elde edilen hareketli nesne çıkarımına erozyon filtresi uygulanması 48
Şekil 4.31 Genişleme filtresi uygulanmış örnek görüntü 49
Şekil 4.32 HRR ile elde edilen hareketli nesne çıkarımı 50
Şekil 4.33 HRR ile elde edilen hareketli nesne çıkarımına genişleme filtresi uygulanması 50
Şekil 4.34 Açma filtresi uygulanmış örnek görüntü 51
Şekil 4.35 Kapama filtresi uygulanmış örnek görüntü 52

1 GİRİŞ

Dijital video sistemleri, donanım maliyetlerinin ucuzlaması ile birlikte özellikle videoların saklanması aşamasında oldukça yaygın olarak kullanılmaktadır. Geçmişte analog ortamlarda kullanılan pek çok sistem artık dijital ortama geçmektedir. Bu geçişte dijital ortamın sağladığı olanaklar kullanılarak içerik bilgisi de elde edilme çalışmalarına başlanmıştır.

Günümüzde multimedya dosyalarından içerik elde etme çalışmaları, oldukça popüler bir konu haline gelmiştir. Dijital video dosyalarından elde edilecek en temel içerik, hareketli nesnelerin çıkarılmasıdır. Hareketli nesneler düzgün bir şekilde tespit edildikten sonra ileriki aşamalarda hareketli nesneler sınıflandırılarak hareketin içeriği de çıkarılabilecektir.

Bu çalışmada; hareketli nesneleri çıkarımda arka planın modellenmesi ve modellenen arka plan görüntüsü yardımı ile hareketli nesnelerin çıkarım yöntemleri incelenmiştir.

Bölüm 2'de belli başlı video dosya formatları incelenmiş, bu formatların dosya yapısı ve sıkıştırma teknikleri açısından genel bilgiler verilmiştir.

Bölüm 3'de piksel tabanlı arka plan modelleme yöntemlerinden PFinder ve HRR incelenmiştir. Bu yöntemler incelenirken, kullanılan kameraların sabit olduğu öngörülmüştür. PFinder ve HRR modelleme yöntemleri karşılaştırılmış, karşılaştırmada arka planın ortamın değişikliklerine karşı dinamik güncellenmesi, oluşan arka plan modelinin aslına yakınlık derecesi ve kaynak tüketimi gibi kriterler göz önüne alınmıştır.

Arka plan modellerini kullanarak hareketli nesne çıkarımı yapan yöntemler Bölüm 4'de incelenmiş, genel olarak hareketli nesne çıkarımının artı ve eksileri ile kullanılan arka plan modellerine göre avantajları incelenmiştir.

Hareketli nesne çıkarım yöntemlerinden elde edilen çıktıların genel sorunlarını düzeltme amaçlı kullanılmak üzere, eşik filtresi ve morfolojik filtreler Bölüm 5'de araştırılmış, bu filtrelerin hareketli nesne çıkarım yöntemlerinin sonuçlarına uygulanmasının sonuçları incelenmiştir.

Bu çalışma boyunca yapılan uygulamalarda, Microsoft.Net Framework 1.1 platformunu ve açık kaynak kodlu AForge.Net görüntü işleme kütüphanesini kullanılmıştır. Uygulama geliştirme dili olarak C# seçilmiştir.

2 DİJİTAL VİDEO

Video; hareketi oluşturan sabit karelerin kaydedilmesi, işlenmesi, saklanması, bir yerden bir yere iletilmesi ve bu karelerin sıralanmasıyla görüntünün oluşturulması teknolojisine verilen isimdir [Wikipedia]. Video teknolojileri ilk başlarda televizyon için geliştirilmiş olsa da ilerleyen zamanda pek çok aygıt / ortam için geliştirilmeye devam edilmiştir.

Başlangıçta analog video sistemleri yaygın olarak kullanılmaktayken, ucuzlayan ve gelişen donanım maliyetleri, dijital videonun kullanılmasını olanaklı hale getirmiştir. Her ne kadar donanım maliyetleri düşse ve donanım kapasiteleri çoğalsa da, video verisinin sıkıştırmasız olarak bilgisayarlar ve dijital depolama aygıtları üzerinde saklanması / iletilmesi olanaksızdır. Örneğin 360*288 çözünürlükte renkli, saniyede 25 kare içeren 1 saniyelik veri içeren bir sıkıştırmasız video dosyası ortalama 7MB yer kaplamaktadır (360*288*3*25) [Taşkın, 2007]. Bu sebepten dolayı, kayıplı sıkıştırma yöntemleri kullanılarak video verisinin dijital ortamlarda efektif olarak kullanılması sağlanmıştır.

2.1 Video Dosya Formatları

Temel olarak tüm video dosya formatları, karelerden oluşan video verisini kayıplı olarak saklayıp, daha sonra bir yazılım sayesinde videoyu tekrar oluşturan karelere dönüşüm yapılabilen dosyalardır.

2.1.1 AVI

Avi, Microsoft tarafından geliştirilen ve tanıtımı Kasım 1992'de yapılan çoklu ortam formatıdır. AVI dosyaları, eş zamanlı olarak video ve ses içeriğini saklamaktadır. Avi sanılanın aksine bir sıkıştırma formatı değil, bir dosya yapısı formatıdır [Microsoft corp., 1997].

Avi, Microsoft'un **RIFF** dosya yapısının özel bir uygulamasıdır. Bu yapı, Electronic Arts'ın 1980'lerin ortasında geliştirdiği **IFF** dosya yapısının bir kopyasıdır. RIFF ve IFF dosyanın içeriğini **chunk** adı verilen veri blokları şeklinde tutar. Her veri bloğu 4 byte'lık başlık bilgisiyle ayırt edilir. İlk veri bloğu **hdlr** başlık bilgisine sahiptir ve bu blok videonun yapısı hakkında bilgileri içerir. İkinci veri bloğunun başlığı **movi'**dir ve videoyu oluşturan görüntü ve ses verisini içerir. **idx1** başlık bilgisine sahip veri bloğu ise tüm veri bloklarının adreslerinin indekslerini tutmaktadır.

Movi bloğunun içindeki data; sıkıştırılmamış veri blokları veya MPEG, Real Video, MPEG4 gibi sıkıştırma formatlarıyla sıkıştırılmış olabilir.

2.1.2 MPEG

Mpeg; ilk çalışmalarına 1988 yılında başlayan bir araştırma grubudur. Bu grup çeşitli üniversite ve araştırma enstitülerinden üyelere sahiptir. Mpeg grubunun amacı video ve ses dosyalarının sıkıştırılması için standartlar oluşturmaktır ve günümüze dek MPEG-1, MPEG-2, MPEG-3, MPEG-4, MPEG-7 gibi standartları oluşturmuşlardır [Asai ve Fuminobu,1998].

MPEG video standartlarının temel yapısında kareler I,B,P olarak sınıflandırılarak B, P sınıfındaki kareler diğer karelerden oluşturulmaktadır. Bu her karenin tüm bilgisini saklamak yerine, I karelerinin tüm bilgisini saklayıp, B ve P karelerinin ise sadece farklarını saklayarak büyük ölçüde kayıplı sıkıştırma sağlamaktadır. Ayrıca saklanan karelerin içeriği **DCT** ile sıkıştırıldığından I karelerinin de boyutları düşürülmüştür.

2.1.3 RealVideo

RealVideo; 1997 yılında RealNetworks şirketi tarafından geliştirilen ve temellerini H.263'ün oluşturduğu kayıplı sıkıştırma formatıdır. Bu format pek çok işletim sisteminde ve taşınabilir cihaz üzerinde çalışabilmektedir. Günümüzde internet üzerinden yapılan video yayınlarında popüler olarak kullanılmaktadır.

RealVideo 8. sürümüne kadar H.263 tabanlı sıkıştırma formatını kullanmıştır. 8. sürümden sonra kendi sıkıştırma formatlarını geliştirmeye başlamışlardır. RV10 ve RV20 H.263 tabanlı, RV30 ve RV40 RealNetworks'ün patentli sıkıştırma formatlarıdır. İlk RealVideo sürümü RV10'u kullanırken RealVideo 10, RV40'ı kullanmaktadır.

RealVideo'nun ilk sürümünde Iterated Systems'in ClearVideo teknolojisi kullanılmış fakat sonraki sürümlerde bu teknoloji RealVideo'dan çıkarılmıştır. RealVideo; RealMedia dosyası üzerinden veya ağ üzerinden Real Time akış protokolünü kullanarak görüntülenebilmektedir.

2.1.4 H.261

H.261, 1990 yılında **ITU-T** tarafından tasarlanmış ve pratiğe dökülebilen ilk video sıkıştırma formatıdır. Bu formatın asıl amacı video görüntüsünün **ISDN** hatları üzerinden iletilebilecek boyutlara getirilebilmesidir. Her ne kadar 1984 yılında yayınlanan H.120, ilk video sıkıştırma standardı olsa da hayata geçirilememiştir [The International Telegraph and Telephonne Consultative Comitee, 1990].

MPEG çalışma grubunun oluşturduğu Mpeg-1, Mpeg-2 vb. formatların temelini H.261 oluşturmaktadır.

3 ARKA PLAN MODELLEME

Video Gözetleme Sistemleri, hesaplama tutarlarının ve görüntü sensörlerinin ucuzlaması, insan kaynağı kullanılarak yapılan işlemlerin efektif olmaması gibi sebepler sayesinde **Bilgisayar Görüş** alanında popüler bir konu haline gelmiştir. Otomatik gözetleme sistemleri konusunda geçen yıllarda pek çok araştırma çalışmaları başlatılmıştır [Collins vd., 2001], [Haritaoglu vd., 2000], [Bobick ve Davis, 2001], [Javed ve Shah, 2002]. IBM ve Microsoft gibi yazılım şirketleri de hareket algılamaya yönelik çalışmalar yürütmektedir [Haritaoglu ve Flickner, 2001], [Perez vd., 2002].

Gözetleme amaçlı kullanılan sistemlerin çoğu sabit bir kameranın belirli alanın görüntüsünü almasından oluşur. Her ne kadar bir kamerayı gözetlenecek alana monte etmek ucuz ve kolay bir iş olsa da bu kameradan alınan görüntüyü işleyecek insan gücü oldukça maliyetli olmaktadır [Ekinci ve Gedikli, 2005]. Günümüzde kullanılan gözetleme sistemleri, sabit kamera ve bu görüntüyü kaydeden bir kayıt cihazından oluşmaktadır. Bu sistemden elde edilen veri ancak olay sonrası inceleme amaçlı kullanılmaktadır. Gözetleme Otomasyonlarından elde edilmek istenilen görüntünün sadece kaydedilmesi değil, görüntüden gerçek zamanlı içeriğin elde edilebilmesidir. Bu sistemlerden istenen en temel veri; alana giren, hareket eden nesnelerin tespitidir.

Hareket eden nesnelerin tespitinin en kolay yolu, var olan kare ile bir önceki karenin karşılaştırılmasıdır. Bu karşılaştırma sadece iki kare arasındaki farkı bulmayı sağlar. Hareket eden nesnelerin tutarlı ve tam şekilde bulunması işlemi ise ancak arka planın oluşturulması, bu modelin ani ışık değişimi, arka planda oluşan değişiklikler gibi etmenlere karşı modelin güncellenebilmesiyle mümkündür.

Arka plan modelleme konusunda uygulanabilen en temel varsayım, hareket etmeyen cisimler haricinde kalan görüntünün istatistiksel verileriyle modellenebilecek düzgün bir davranış sergilemesidir. Eğer bunu sağlayan bir model oluşturulursa, modele

uymayan parçalar; sahne alanına giren, hareket eden nesneleri belirtir. Bu işlem **Arka Plan Modelleme** ya da **Arka Plan** olarak adlandırılmaktadır.

Arka plan modelleme için kullanılan yöntemlerin en büyük bölümünü tümevarım yöntemler oluşturur. Bu yöntemler, görüntüyü oluşturan her bir karenin piksel değerlerinin dağılımını kullanarak arka planı modeller. Arka planın gerçekçi bir şekilde modellenmesi, eğitim için 10 – 30 saniye süresince video karelerinin istatistiksel değerlenin yorumlanması ile olur [Ekinci ve Gedikli, 2005].

Eğitim süreci içeren tekniklerde, arka planın gerçeğine en yakın şekilde modellenebilmesi için, eğitim süresince hareketli nesnelerin mümkün olduğunca az, hatta hiç olmaması gerekmektedir. Fakat gerçek durumlarda bu çoğu zaman mümkün değildir. Kalabalık alışveriş merkezlerinden, sürekli trafik akışı olan yollara kadar çok geniş bir aralıktaki sahnelerde arka plan modelleme çalışması yapılması gerekebilir. Bu yüzden modelin öğrenim süresinde bu tarz durumlara karşı çözüm getirmesi gerekmektedir.

3.1 Arka Plan Modelleme Yöntemleri

Arka plan modelleme yöntemlerinden, piksel değerlerini kullanan yöntemlerinden **PFinder** ve **HRR** incelenecektir. Yapılan uygulamalarda her iki yöntem için de piksel değerlerinin gri tonlamalı değerleri kullanılmaktadır. Şekil 3.1'de, bu modellerin uygulamasında kullanılan test videosunun 25 karede bir alınmış görüntüleri bulunmaktadır.



Şekil 3.1 Test Videosunun 0 - 450 arası karelerinin 25 karede bir alınmış görüntüleri

3.1.1 PFinder

PFinder yöntemi, arka plan modelini yeni karedeki piksel değerlerini α katsayısıyla çarpıp güncellemesi ile arka plan modeli oluşturan bir yöntemdir [Wren vd., 1997]. Bu model var olan arka planı ışık değişimi, arka plan yapısının farklılaşması gibi değişikliklerle arka planı dinamik olarak güncellemektedir.

PFinder yönteminde kullanılan arka plan piksel değerinin hesaplama yöntemi (3.1)'deki eşitlikte gösterilmiştir. Eşitlikteki α katsayısı, her yeni karenin arka plan modelini güncelleme katsayısıdır. Bu değerin az veya fazla seçilmesine göre model yapısı değişmektedir.

Az seçilen α değeri, modelin değişikliklere karşı tepki süresini düşürürken, hareketli nesnenin de bir kısmının arka plana dahil edilmesine yol açar.

Fazla seçilen α değeri ise modelin değişikliklere karşı tepki süresini artırır, bu da ani ışık değişimi vb değişikliklerin modele yansımasının zaman almasına yol açar.

$$B_t = (1-\alpha)B_{t-1} + \alpha I_t \tag{3.1}$$

B_t = t anındaki arka plan piksel değeri

B_{t-1} = t-1 anındaki arka plan piksel değeri

I_t = t anındaki görüntünün piksel değeri

α = güncelleme katsayısı ([0,005 – 0,100] arası değerler kullanılmaktadır.)

Artıları:

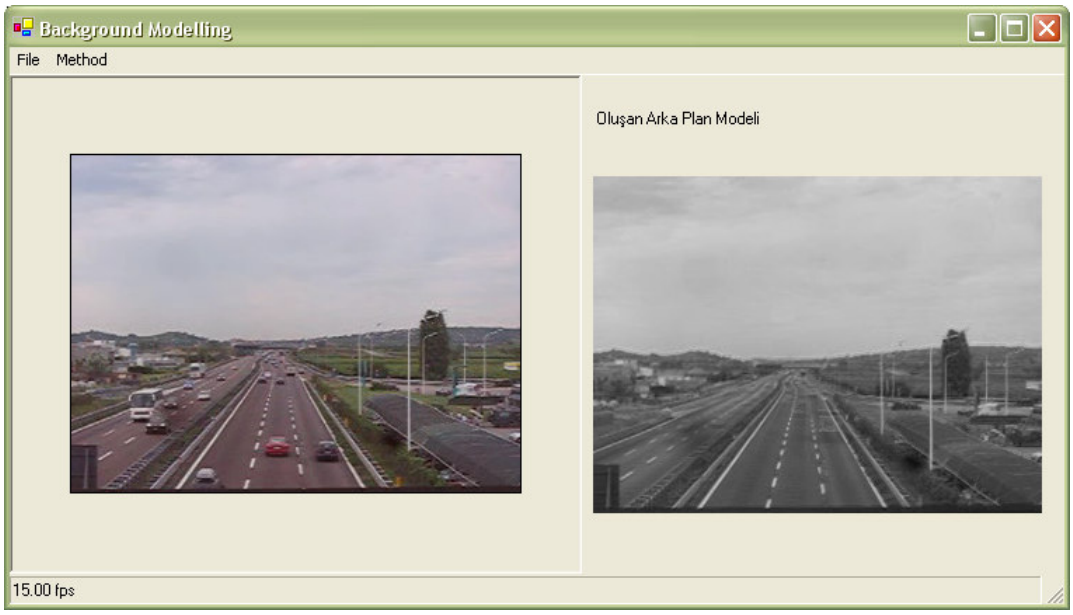
- Dinamik olarak arka planı güncellemesi, arka planın değişikliklerinin modele dinamik olarak yansıması.
- Bellek ihtiyacının az olması.

Eksileri:

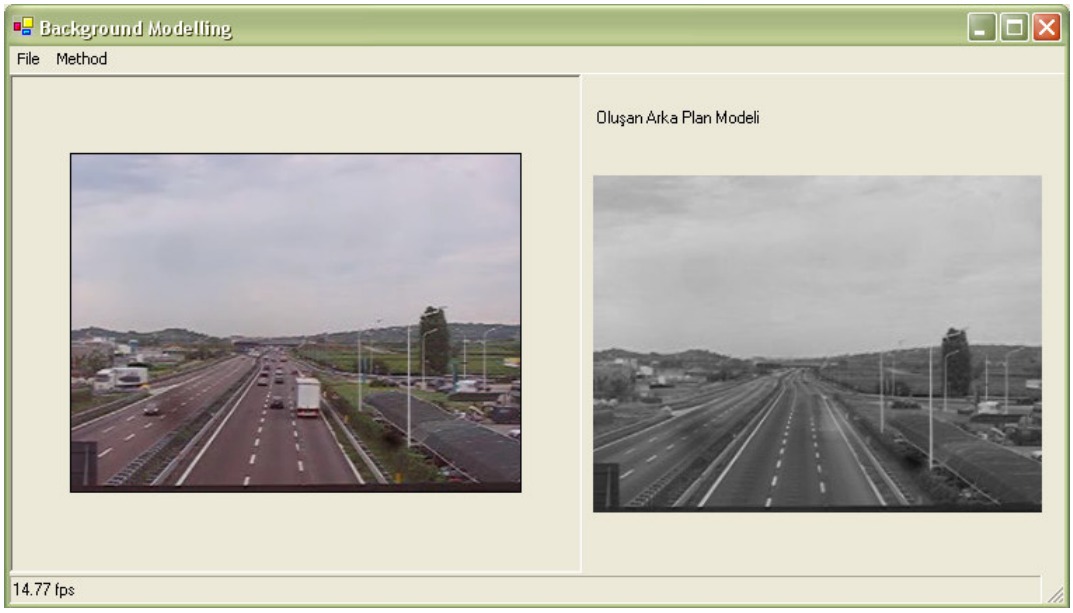
- Ani değişikliklerin arka plan modeline yansımasının zaman alması.
- Arka planda hareket çok ise (özellikle belli alanda yoğunlaşmış hareketler var ise) arka plan modelinin bozulmaya uğraması.
- (1) eşitliğindeki α değeri optimum seçilmezse modelin verimli olmaması.
- İlk alınan karenin arka plan olarak atanmasından (ilk kare hareketli nesneyi de içeriyor olabilir) dolayı arka planın düzeltilmesi zaman almaktadır.

Örnekler:

Şekil 3.2, 3.3, 3.4 ve 3.5'de örnek bir video görüntüsünün farklı α değerleri ve farklı öğrenme sürelerinde oluşturulan arka plan modelleri gösterilmiştir.



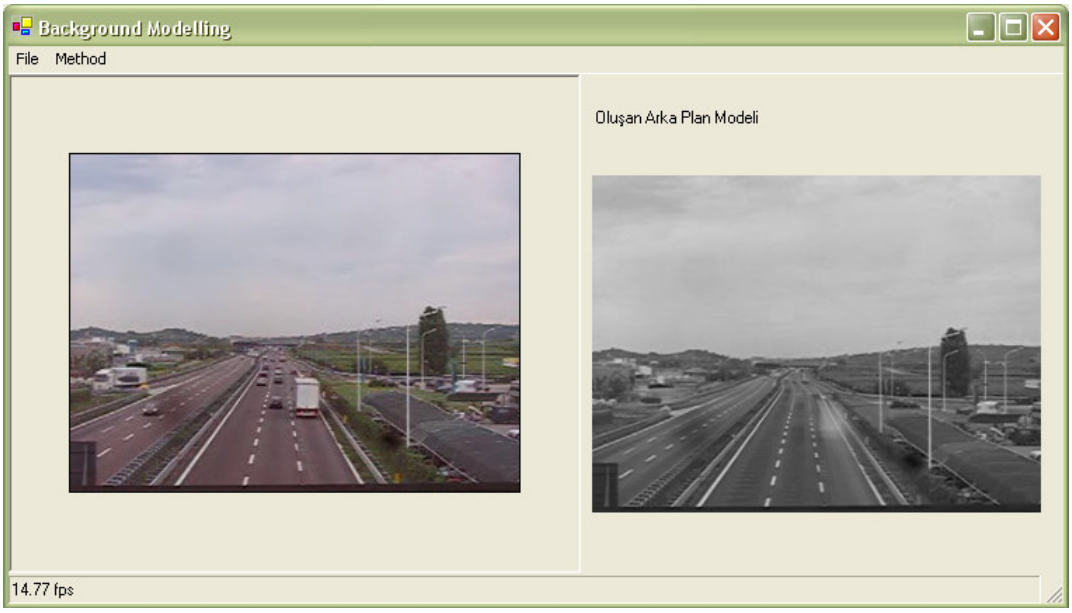
Şekil 3.2 100. karede oluşan arka plan, α = 0.025



Şekil 3.3 200. karede oluşan arka plan, α = 0.025



Şekil 3.4 500. karede oluşan arka plan, α = 0.025



Şekil 3.5 200. karede oluşan arka plan, α = 0.05

Kaynak Kod:

PFinder arka plan modelleme yönteminin C# diliyle yazılmış kodu aşağıdaki gibidir.

```
namespace motion
{
	using System;
	using System.Drawing;
	using System.Drawing.Imaging;

	public class PFinder
	{
		private Bitmap backgroundImage;

		private int width;
		private int height;
		private int stride;
		private float alpha;



		public Bitmap BackGroundImage
		{
			get{ return backgroundImage;  }
			set{ backgroundImage = value; }
		}

		public float Alpha
		{
			get{ return alpha; }
			set{ alpha = value; }
		}
```

```
            public PFinder(float alpha)
            {
                  this.alpha = alpha;
            }


            public  unsafe void UpdateBackground( Bitmap newFrame )
            {
                  if( backgroundImage == null )
                        backgroundImage = (Bitmap) newFrame.Clone();


                  int width = newFrame.Width;
                  int height = newFrame.Height;

                  BitmapData backgroundData =  this.backgroundImage.LockBits(
new Rectangle( 0, 0, width, height),ImageLockMode.ReadWrite,
PixelFormat.Format8bppIndexed );
                  BitmapData newFrameData =  newFrame.LockBits( new
Rectangle( 0, 0, width, height),ImageLockMode.ReadWrite,
PixelFormat.Format8bppIndexed );

                  int srcOffset = newFrameData.Stride - width;

                  byte * srcNew = (byte *) newFrameData.Scan0.ToPointer( );
                  byte * srcBcgr = (byte *) backgroundData.Scan0.ToPointer(
);

                  // her satır
                  for ( int y = 0; y < height; y++ )
                  {
                        // her piksel
                        for ( int x = 0; x < width; x++, srcBcgr ++,srcNew++
)
                        {
                              srcBcgr[0] = Convert.ToByte( (alpha *
srcNew[0]) + ((1-alpha) * srcBcgr[0]));
                        }
                        srcBcgr += srcOffset;
                        srcNew += srcOffset;

                  }

                  this.backgroundImage.UnlockBits( backgroundData );
                  newFrame.UnlockBits( newFrameData );



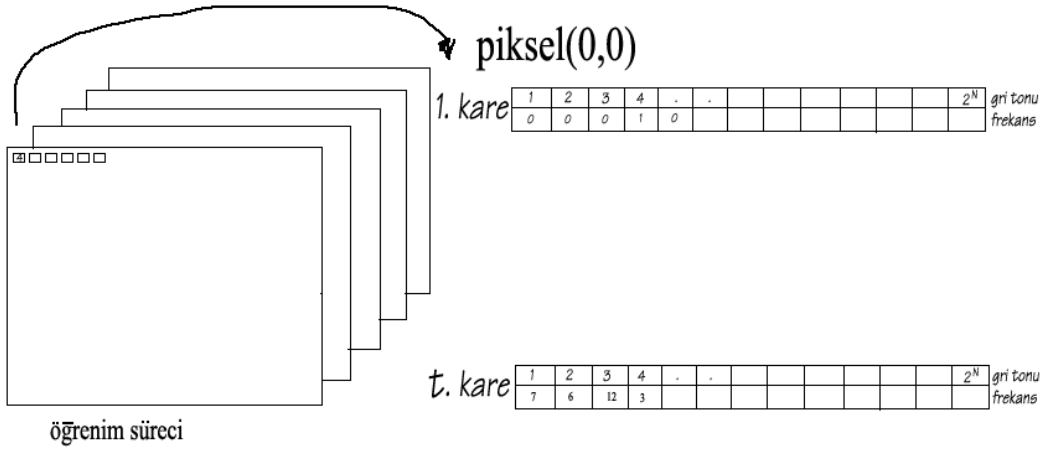
            }

      }
}
```

3.1.2 HRR (En çok tekrarlanma oranı)

HRR modelinin çıkış noktası, görüntüyü oluşturan karelerdeki piksellerin en sık görülenlerinin arka planı oluşturduğu düşüncesidir [Ekinci ve Gedikli, 2005]. Belirli bir öğrenme süresince karelerdeki piksel değerleri alınır, frekansı en çok olan pikseller arka plan değeri olarak atanır. Bir dizide parlaklık değerlerinin görünme frekansı tutulur, öğrenme süresi içinde her karede bu değerler güncellenir. Öğrenme süreci tamamlandığında, en çok frekansa sahip piksel değerleri arka plan modelini oluştur.

Şekil 3.6'da her piksel için tutulan tablo gösterilmiştir. Bu tabloda her piksel için *t* sayıdaki öğrenim karesi boyunca gri ton değerlerinin frekansı tutulmaktadır. Öğrenim sonunda her piksel için en yüksek frekansa sahip gri ton değeri, o pikselin arka plandaki değerini oluşturmaktadır.



Şekil 3.6 HRR Öğrenme süreci gösterimi

Arka plan oluştuktan sonra, modeli arka planda oluşan değişikliklere karşı güncellemek için de öğrenim süreci belli bir süre sonra tekrarlanmalıdır.

Artıları:

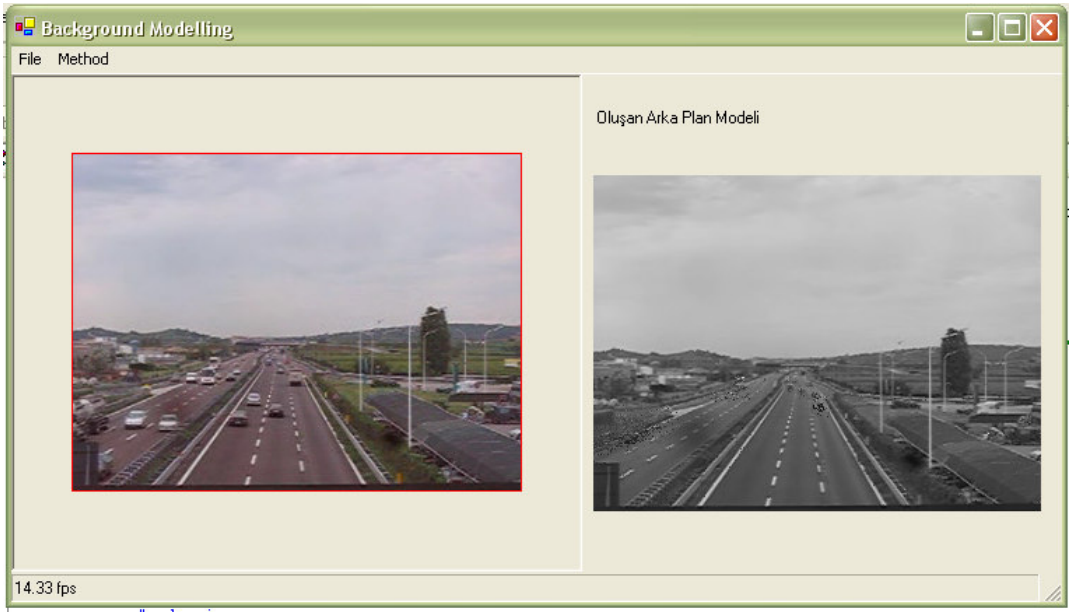
- Modelin, aslından çok az sapmayla oluşması.

Eksileri:

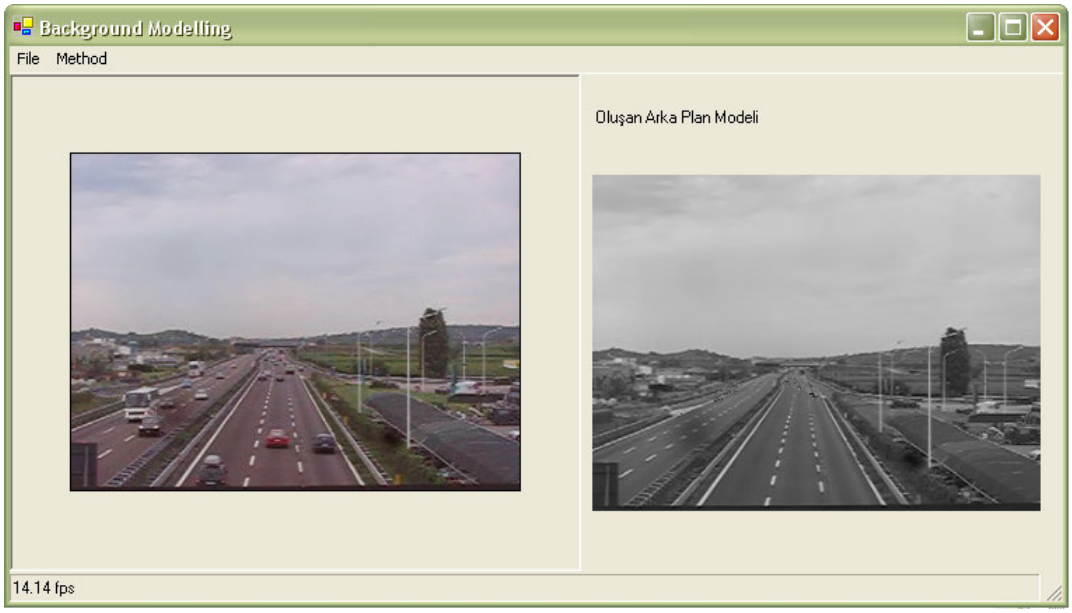
- Öğrenim süresince gerekli olan fazla bellek ihtiyacı
- Model oluşturma aşamasında fazla CPU zamanı tüketimi.
- Arka plan değişiminin tekrar öğrenme sürecine girmeden modele yansımaması.

Örnekler:

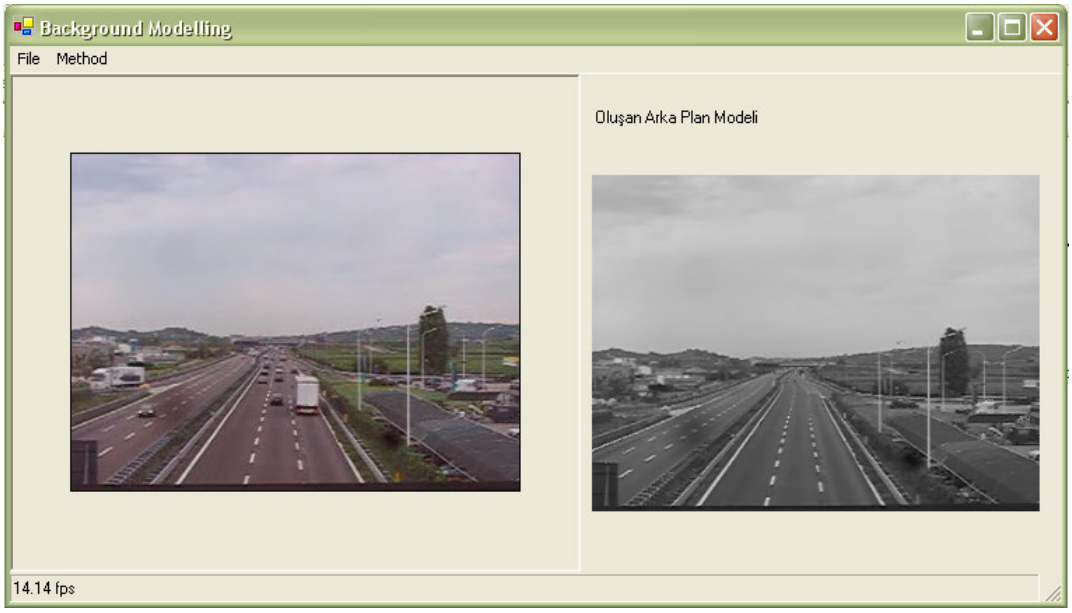
Şekil 3.7, 3.8 ve 3.9'da örnek video görüntüsünün sırasıyla 50, 100 ve 150 karelerinde PFinder yöntemiyle oluşturulan arka plan modeli gösterilmiştir.



Şekil 3.7 50. karede oluşan arka plan



Şekil 3.8 100. karede oluşan arka plan



Şekil 3.9 200. karede oluşan arka plan

Kaynak Kod:

HRR arka plan modelleme yönteminin C# diliyle yazılmış kodu aşağıdaki gibidir.

```
namespace motion
{
	using System;
	using System.Drawing;
	using System.Drawing.Imaging;


	public class HRR
	{

		private byte [,,] frameArray;
		private Bitmap backgroundImage;

		private int width;
		private int height;
		private int stride;
		private int frameCount;

		#region getter & setter:

		public Bitmap BackGroundImage
		{
			get{ return backgroundImage;  }
			set{ backgroundImage = value; }
		}

		public int FrameCount
		{
			get{ return FrameCount; }
			set{ frameCount = value; }

		}

		#endregion


		public HRR( int Width,int  Height )
		{
			width = Width;
			height = Height;

			frameArray = new byte[width,height,256];

		}

		public HRR( )
		{

		}

		public void ClearFrames()
		{
			frameArray = new byte[width,height,256];
		}

		public unsafe void AddFrame( BitmapData sourceData)
		{

			if(width == 0 && height == 0 )
			{
				width = sourceData.Width;
				height = sourceData.Height;
				stride = sourceData.Stride;

				frameArray = new byte[width,height,256];
```

```
                }


                int srcOffset = stride - width;

                byte * src = (byte *) sourceData.Scan0.ToPointer( );

                // her satır için
                for ( int y = 0; y < height; y++ )
                {
                        // her piksel için
                        for ( int x = 0; x < width; x++, src ++ )
                        {
                                frameArray[x,y,src[0]] += 1;
                        }
                        src += srcOffset;

                }

                frameCount++;



        }


        public unsafe Bitmap GetBackGround()
        {


                BitmapData backgroundData = backgroundImage.LockBits( new
Rectangle( 0, 0, width,
height),ImageLockMode.ReadWrite,PixelFormat.Format8bppIndexed );

                int srcOffset = stride - width;

                byte * src = (byte *) backgroundData.Scan0.ToPointer( );

                // her satır
                for ( int y = 0; y < height; y++ )
                {
                        // her piksel
                        for ( int x = 0; x < width; x++, src ++ )
                        {
                                byte maxVal = 0;

                                for(int val = 0 ; val<255 ; val++)
                                {
                                        if( frameArray[x,y,val] > maxVal)
                                        {
                                                maxVal = frameArray[x,y,val];
                                                src[0] = Convert.ToByte(val);
                                        }
                                }

                        }
                        src += srcOffset;

                }

                backgroundImage.UnlockBits(backgroundData);

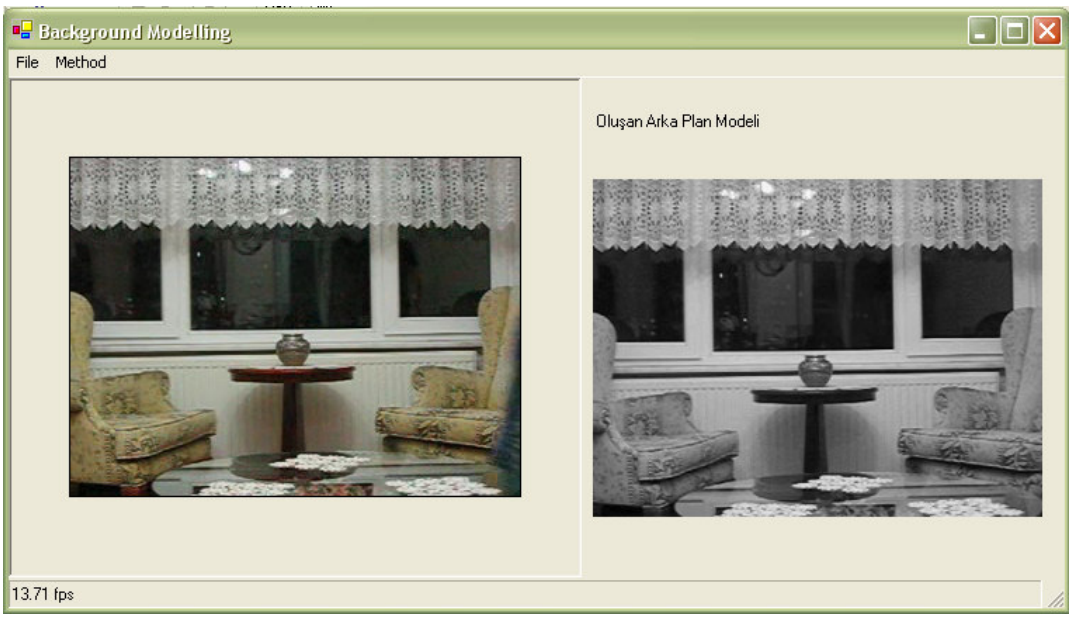
                return backgroundImage; } } }
```

3.2 PFinder ve HRR Yöntemlerinin Karşılaştırması

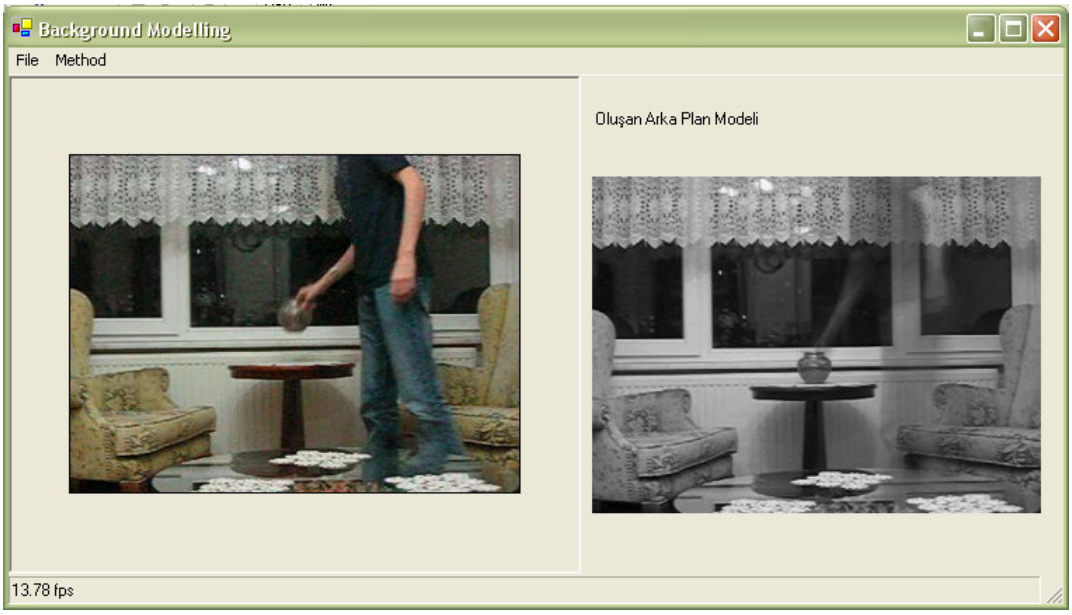
PFinder ve HRR yöntemleri piksel bazında arka plan oluşturma yöntemlerindendir. Bu iki yöntem arasındaki en belirgin fark; PFinder yönteminin dinamik olarak arka plan modelini oluşturup güncellemesine karşın, HRR yönteminin belirli bir öğrenim süreci sonrasında modeli oluşturup, öğrenme süreci yenilenene kadar modeli güncellememesidir.

Şekil 3.10, 3.11, 3.12, 3.13 ve 3.14, 3.15, 3.16'daki örnek görüntüde 120. karede arka planda bulunan vazo kaldırılmıştır.

PFinder yöntemi ile arka plan modeli dinamik olarak güncellenmiştir. (Şekil 3.10, 3.11, 3.12, 3.13'de $\alpha = 0.05$ alınmıştır.)



Şekil 3.10 100. karedeki görüntü ve arka plan modeli



Şekil 3.11 125. karedeki görüntü ve arka plan modeli

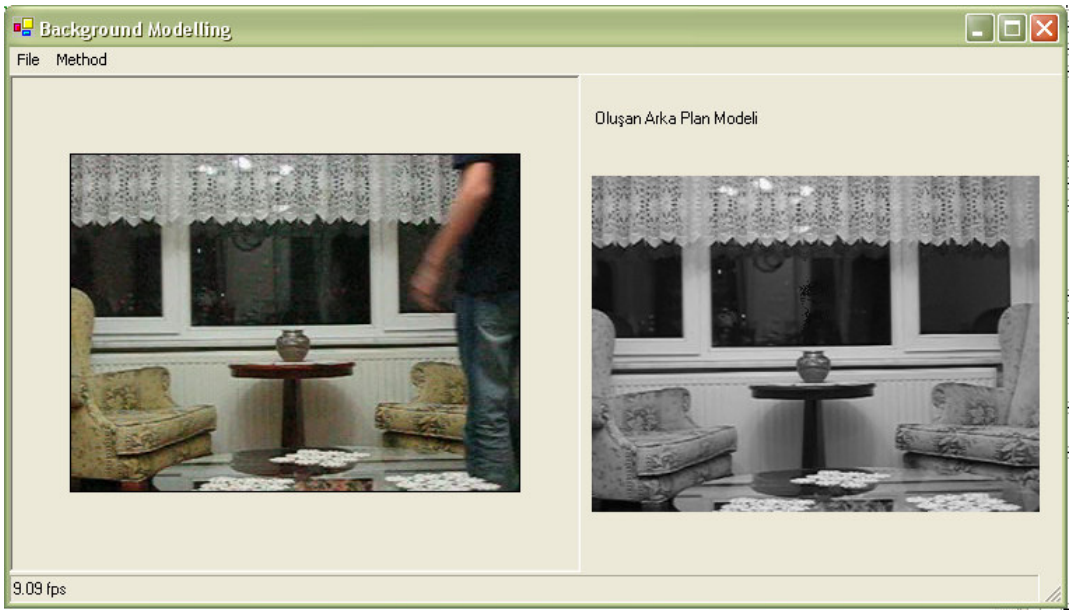


Şekil 3.12 150. karedeki görüntü ve arka plan modeli



Şekil 3.13 175. karedeki görüntü ve arka plan modeli

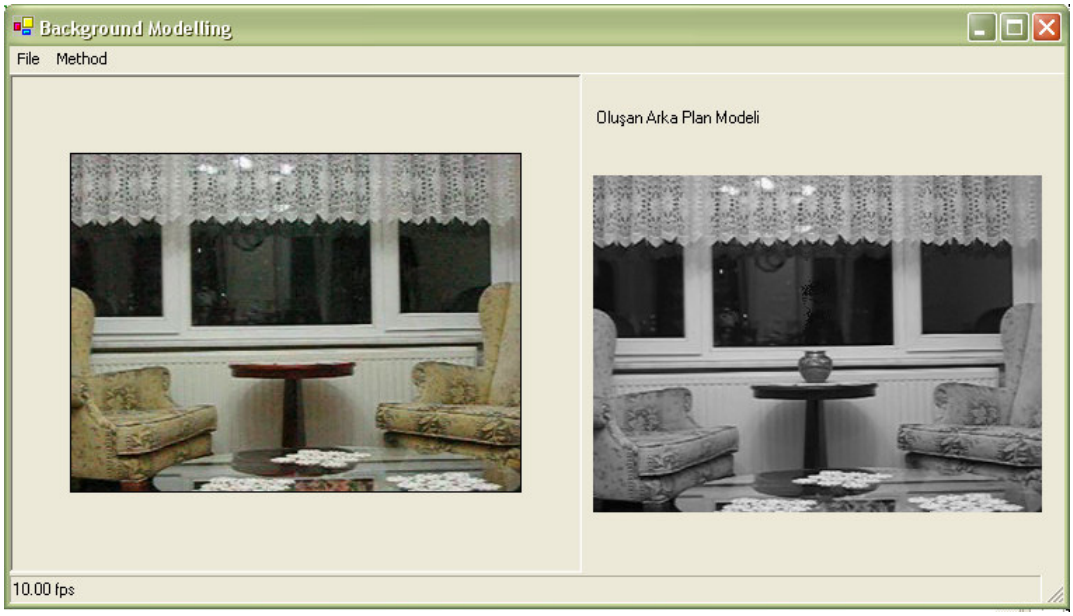
Şekil 3.14., 3.15. ve 3.16'da verilen örnekte de görüldüğü gibi, aynı video görüntüsü kullanılarak HRR yöntemi ile elde edilen arka plan modelinde herhangi bir değişiklik olmamıştır.



Şekil 3.14 100. karedeki görüntü ve arka plan modeli



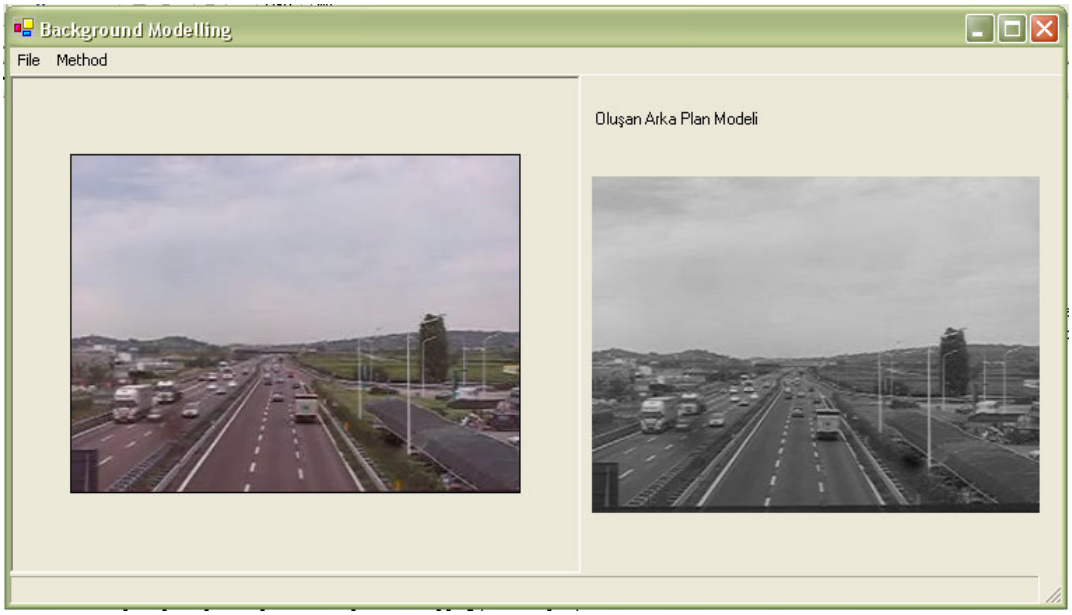
Şekil 3.15 125. karedeki görüntü ve arka plan modeli



Şekil 3.16 150. karedeki görüntü ve arka plan modeli

HRR yöntemi öğrenme süreci içerisinde herhangi bir model sunmazken, PFinder yöntemi ile ilk kareden itibaren arka plan modeli elde edilebilir. PFinder yönteminde ilk karelerde elde edilen arka plan modeli şekil 3.17'de görüldüğü gibi yanıltıcı olabilir.

Bunun sebebi, ilk karenin arka plan olarak alınması ve bunun güncellenmesinin zaman almasıdır.



Şekil 3.17 5.Karede oluşan arka plan modeli (α = 0.05)

PFinder her yeni kare ile arka plan modelini α katsayısı ile güncellemektedir. Bu modelin arka plandaki değişikliklere dinamik olarak uyum sağlamasına imkan verirken, arka plana ait olmayan hareketli görüntünün de α katsayısı ile arka planı güncellemesinden dolayı Şekil 3.18'de de görüldüğü gibi arka planda istenmeyen bozulmalara sebep olabilmektedir. Buna karşın HRR yöntemi ile oluşturulan arka plan modeli, öğrenme süreci haricinde herhangi bir güncelleme yapmadığından, oluşan arka plan PFinder yöntemine göre daha iyi sonuç vermektedir (Şekil 3.19).



Şekil 3.18 PFinder yöntemi ile oluşturulan arka plan görüntüsü



Şekil 3.19 HRR PFinder yöntemi ile oluşturulan arka plan görüntüsü

3.3 Performans Değerlendirme

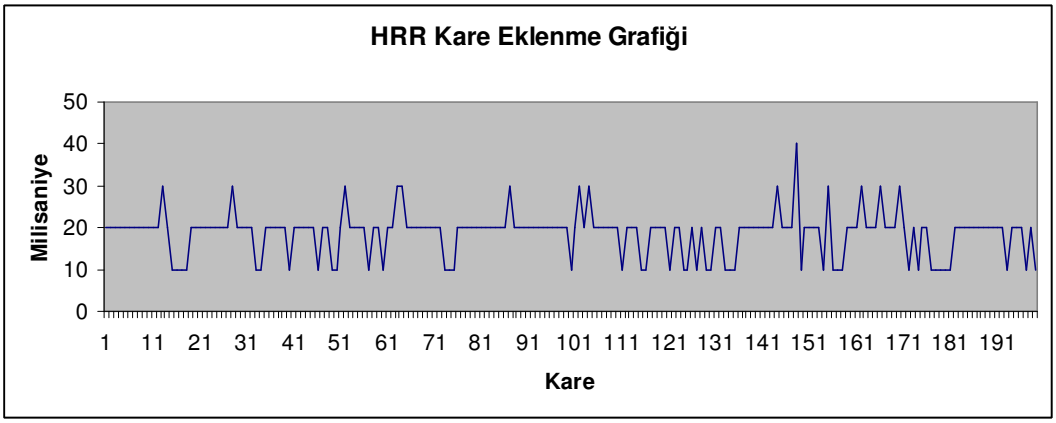
Yapılan testlerde HRR ve PFinder yöntemlerinin arka plan modelleme için kullandığı kaynak tüketimi ve performans değerlendirmeleri incelenmiştir.

Testlerde kullanılan sistem: P4 2.4 GHz Intel Processor, 512 Mb Ram.
Testlerde kullanılan işletim sistemi: Microsoft Windows XP Professional.
Testte kullanılan video dosyası: 320*240, 15fps , DivX coded (Şekil 1).

Test ortamında gerçek zamanlı arka plan modellemesi yapılabilmesi için, var olan yöntemlerin her kare için maksimum 1/15 = 66.66 milisaniye harcaması gerekmektedir. Bu süreyi aşan hesaplama süreleri yeni karenin gösterim süresini etkileyeceğinden, gerçek zamanlı arka plan modelleme mümkün olmamaktadır.

3.3.1 HRR performans ölçümleri

HRR yöntemiyle test videosunun 200 karelik eğitim sürecinde her yeni karenin var olan istatistiksel veriye eklenme süreleri ölçülmüştür.



Şekil 3.20 HRR Öğrenim Grafiği

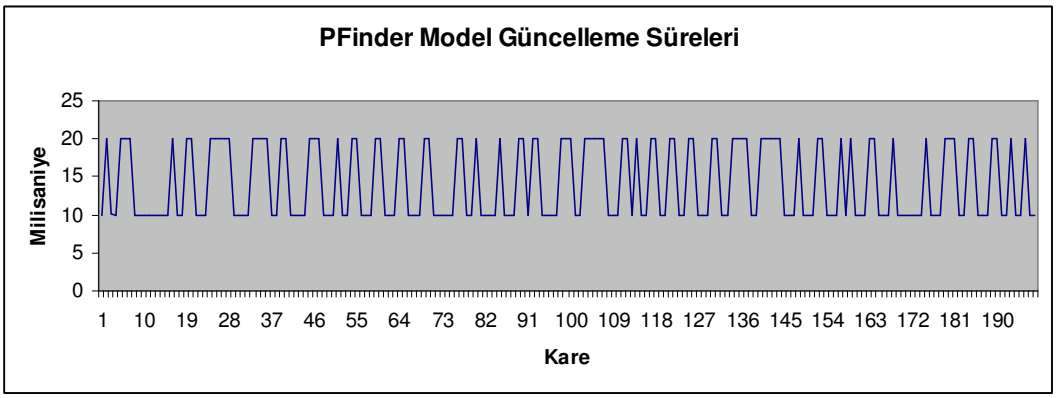
Her karenin öğrenim sürecine eklenmesi ortalama 18.6632 ms almaktadır. Video dosyalarının saniyede 15 – 30 arasında kare içerdiği varsayıldığında, iki kare arasında geçen süre 66.66 – 33.33 ms arasındadır. Test ortamı için gereken maksimum süre 66.66 ms olduğundan, HRR yönteminde öğrenim süreci boyunca gerçek zamanlı hesaplama yapılabilmektedir.

Test videosunun öğrenim süreci bitip arka plan modeli oluşturma aşamasında ise 240.3456 ms harcanmıştır. Bu süre 66.66 ms'den büyük bir değer olduğundan, hesaplama bitip bir sonraki kare gösterilirken 173.68 ms'lik bir bekleme söz konusu olmaktadır.

Her ne kadar HRR yönteminin hesaplama süreleri yüksek gözükse de, öğrenim süresi sonunda arka plan modeli oluştuktan sonra bir sonraki öğrenim sürecine kadar herhangi bir hesaplama yapılmamaktadır. Kaynak tüketimi öğrenim süreci içinde yoğunlaşmıştır.

3.3.2 PFinder performans ölçümleri

PFinder metoduyla, test videosunun 200 karelik aralığı için hesaplanma süreleri ölçülmüştür.



Şekil 3.21 PFinder Model Güncelleme Zaman/Kare grafiği

PFinder tekniği, her yeni kare değeri ile var olan arka plan modelini güncellediğinden herhangi bir öğrenim süreci yoktur. Test ortamında gelen her yeni karenin var olan modeli güncelleme süresi ortalama 14.28516 ms olarak ölçülmüştür. Bu süre, test ortamı için gerekli maksimum süreden düşük olduğundan, PFinder ile gerçek zamanlı arka plan modellemesi mümkündür.

4 HAREKETLİ NESNE ÇIKARIMI

Hareketli nesne bilgisinin kullanım alanlarından en büyüğü, belirlenen ortam içinde hareketin olup olmaması verisini kullanan alarm, elektronik güvenlik sistemleridir. Bu sistemler, elektronik sensörler (infrared, lazer sensörleri vb.) vasıtasıyla belirli alanın içinde hareketin var olup olmadığı bilgisini elde etmektedir.

Bunun dışında, hareketli nesne içeren ortam / görüntüler içerisinde elde edilecek pek çok veri bulunmaktadır. Otomatik video gözetleme ve dijital video ortamlarının yaygınlaşmasıyla, video dosyalarını işleyerek daha detaylı bilgi edinme amaçlanmıştır.

Otomatik video gözetleme sistemleri ile elde edilebilecek bir üst seviye veri ise hareket eden nesnelerin çıkarılmasıdır. Bu sayede sabit resimlerle videoyu birbirinden ayıran en önemli veri; hareket elde edilmiş olur. Video dosyalarından hareketi çıkarmanın yöntemlerinden biri, her kareyi bir önceki kareyle karşılaştırarak oluşan farkın hareketi oluşturduğu ardışık kare farkı yöntemidir [Rossin ve Ellis, 1995], [Ekinci vd., 2000], [Vass vd., 1998].

Video dosyalarındaki hareketli nesne çıkarımı için kullanılan diğer düşünce, arka plan görüntüsü dışında kalan alanın hareketli nesneyi oluşturduğudur. Arka plan görüntüsü kullanarak hareketli nesne çıkarımı yapan yöntemler, her karenin arka plan ile farkı alınarak oluşturulur [Collins vd., 2001], [Haritaoglu vd., 2000], [Toyama vd., 1998], [Grimson vd., 1998]. Bu bölümde, Bölüm 2'de anlatılan arka plan modelleme yöntemleri ile hareketli nesne çıkarımını uygulama, çıkarılan hareketli nesnenin düzgün oluşturulabilmesi için çeşitli görüntü işleme tekniklerinin kullanımı ele alınacaktır.

4.1 Ardışık Kare Farkı Yöntemi ile Hareketli Nesne Çıkarımı

Video dosyaları dijital ve analog ortamlarda görüntülenirken gerçek görüntünün yaklaşık 1/30 – 1/15 saniyede bir alınmış sabit karelerin ardı sıra görüntülenmesiyle gösterilir. Hareketsiz bir video görüntüsünde kareler arasında fark olmaması beklenirken, hareketin fazla olduğu video görüntülerinde kareler arası farkın en fazla düzeyde olması beklenir. Bu düşünceden hareketle artarda gelen iki kare arasındaki farkın mutlak değeri alınırsa, iki kare arasındaki hareket eden alanlar tespit edilmiş olur.

(4.1)'deki eşitlikte ardışık iki kare arasındaki fark değerini hesaplama işlemi gösterilmiştir.

$$D_t = \left| I_t - I_{t-1} \right| \tag{4.1}$$

D_t = t andında oluşan fark piksel değeri

I_t = t anındaki piksel değeri

I_{t-1} = t-1 anındaki piksel değeri

Şekil 4.1'de, 8 bit renk derinliğinde 4x4 piksellik bir görüntünün fark hesaplaması gösterilmiştir.

213	145	110	45
122	12	34	33
90	43	230	123
243	59	134	3

t anındaki kare (I_t)

214	145	100	54
122	12	43	32
90	45	225	120
243	60	134	4

t-1 anındaki kare (I_{t-1})

1	0	10	9
0	0	9	1
0	2	5	3
0	1	0	1

fark (D_t)

Şekil 4.1 8 bit örnek bir kare için kare farkı hesaplaması

Şekil 4.2 ve Şekil 4.3'de gösterilen, 15 fps video görüntülerinden alınmış ardı ardına gelen iki kare görüntüsüdür. İki kare farkı metodu uygulanarak elde edilmiş fark karesi Şekil 4.4'de gösterilmiştir.



Şekil 4.2 *t-1* anındaki video kare örneği



Şekil 4.3 *t* anındaki video kare örneği



Şekil 4.4 Oluşan fark görüntüsü *Dt*

Şekil 4.4'de görüldüğü gibi, iki kare arasındaki fark genel olarak hareketli nesnenin dış hatlarını oluşturmuştur. Bu yöntemle hareketli nesnenin tamamı değil, sadece bir önceki kareye göre farklılaşmış alanları elde edilir.

4.2 Arka Plan Modelleme Yöntemleri ile Hareketli Nesne Çıkarımı

Video dosyalarında hareketli nesne çıkarım yaklaşımlarından bir diğeri de her yeni karenin daha önce arka plan modelleme yöntemleriyle hesaplanmış arka plan modeliyle karşılaştırılmasıdır. Bu yöntem Bölüm 4.1'de anlatılan kare farkı yöntemine benzer. Kare farkı yönteminden farklı olarak karşılaştırılacak kare olarak bir önceki kareyi değil, arka plan modelini kullanır.

Bu yöntemi incelerken, arka plan modelleme yöntemi olarak Bölüm 3.1'de anlatılan HRR ve PFinder yöntemleri kullanılmıştır.

(4.2)'deki eşitlik, arka plan modeli ile yeni karenin fark değerini hesaplama işlemini göstermektedir.

$$D_t = \left| I_t - B_t \right| \tag{4.2}$$

D_t = t andında oluşan fark piksel değeri

I_t = t anındaki piksel değeri

B_t = t anındaki var olan arka modelinin piksel değeri

4.2.1 HRR yöntemi kullanarak hareketli nesne çıkarımı

Bu teknik; Bölüm 3.1.2'de anlatılan HRR yöntemi ile oluşturulan arka plan modeli ile arka plan modeli oluşturulan video dosyasındaki her yeni karenin Bölüm 4.1'de anlatılan kare farkı yöntemi ile karşılaştırılmasıyla hareketli nesne çıkarımını sağlamaktadır. HRR, arka plan modeli oluşturması için geçen öğrenme süresinde herhangi bir arka plan modeli sunmamaktadır. Dolayısıyla, bu süreç içerisinde hareketli nesne çıkarımı da mümkün değildir.

Şekil 4.5'de örnek video görüntülerinin HRR metoduyla öğrenme süresi 100 kare olarak hesaplanan arka plan modelleri gösterilmiştir. Aynı video görüntülerinin öğrenme sürecinden sonra alınan kare görüntüleri Şekil 4.6'da, aynı anda oluşan fark görüntüleri ise Şekil 4.7'de gösterilmiştir.



Şekil 4.5 HRR yöntemi ile oluşan arka plan modelleri



Şekil 4.6 Hareket anında alınan örnek kareler



Şekil 4.7 Arka plan modeli ile kare arasında oluşan fark görüntüleri

4.2.2 PFinder yöntemi kullanarak hareketli nesne çıkarımı

Video dosyalarında hareketli nesne çıkarım yöntemlerinden bir diğeri ise Bölüm 3.1.3'de anlatılan PFinder arka plan modeli oluşturma tekniğiyle oluşturulmuş arka planı kullanarak, her yeni kareyi bu arka planla karşılaştırma yöntemidir. PFinder arka plan modeli oluşturma yöntemi herhangi bir öğrenme sürecine ihtiyaç duymadığı için ilk kareden itibaren hareketli nesne çıkarımını sağlamaktadır.



Şekil 4.8 PFinder yöntemi ile oluşan arka plan modelleri



Şekil 4.9 Hareket anında alınan örnek kareler



Şekil 4.10 Arka plan modeli ile kare arasında oluşan fark görüntüleri

Şekil 4.8'de örnek video görüntülerinin $\alpha = 0.05$ değerini kullanan PFinder yöntemi ile oluşturulan arka plan modeli gösterilmiştir. Şekil 4.9'da aynı videonun bir karesi ve güncel kare ile arka plan modelinin farkı şekil 4.10'da gösterilmiştir.

4.3 Hareketli Nesne Çıkarımı Yöntemlerinin Karşılaştırmaları

Bölüm 4'de anlatılan bütün yöntemler, video görüntüsündeki her kare ile başka bir karenin farkının alınmasına dayanmaktadır. Bu yöntemler, mevcut kareyi karşılaştırılmak için kullanılan karenin seçiminin farklılaşmasından dolayı birbirinden ayrılır. Bölüm 4.1'deki ardışık kare fark yöntemi karşılaştırmak için bir önceki video

karesini kullanırken Bölüm 4.2'de anlatılan yöntemler, arka plan modelleme yöntemleriyle oluşturulmuş arka plan modellerini kullanmaktadır.

Bölüm 4 altında anlatılan 3 yöntemin de ortak özelliği olan fark alma metodunun eksilerinden biri, hareket eden nesnenin piksel değerleri ile arka planın aynı piksel değerlerine sahip olmasından dolayı oluşan sorundur.



a) Örnek kare b) Ardışık kare farkı

c) PFinder d) HRR

Şekil 4.11 Örnek kare ve hareketli nesne çıkarımları

Şekil 4.11'de, videodan alınmış bir kare ve 3 metot ile bulunan hareketli nesne görüntüleri bulunmaktadır.

Her iki yöntemde de Şekil 4.11 b, Şekil 4.11 c, Şekil 4.11 d'de görüldüğü gibi arka plan ile yakın piksel değerinde bulunan hareketli kısmın fark değeri az olduğundan, oluşan fark karesindeki belirliliği azdır.

Aynı şekilde piksel değerlerini kullanarak hareketli nesne çıkarmanın eksilerinden bir diğeri de, hareketli nesnenin bulunduğu arka planın piksel değerlerinin dağılım aralığının yüksek olmasıdır.



a) Örnek kare b) Ardışık kare farkı

c) PFinder d) HRR

Şekil 4.12 Örnek kare ve hareketli nesne çıkarımları

Şekil 4.12'de videodan alınmış bir kare ve 3 metot ile bulunan hareketli nesne görüntüleri bulunmaktadır.

Şekil 4.12 a'da bir karesi gösterilen video görüntüsünün arka planını oluşturan alanın piksel dağılımı fazla olduğundan, hareketli nesne çıkarım yönteminde de (Şekil 4.12 b, c, d) oluşturulan hareketli nesne fark karesi hareketli nesneleri tam olarak çıkaramamaktadır.

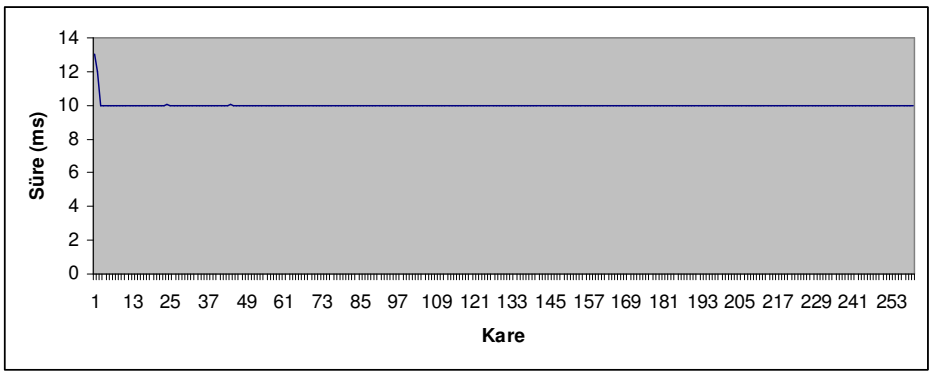
4.3.1 Ardışık kare farkı yöntemi ile hareketli nesne çıkarımının artı eksileri ve performans değerlendirme

Uygulama kolaylığı ve kaynak tüketimi açısından bakıldığında en performanslı hareketli nesne çıkarım tekniği, ardışık kare farkı yöntemidir. Hareket tespiti yapılan alandaki ışık, zemin gibi değişikliklere de en çabuk tepkiyi ardışık kare farkı yöntemi vermektedir. Buna karşın Şekil 4.13'de de görüldüğü gibi hareketli nesnenin sadece bir önceki kareye göre farklılaşmış piksellerini bize vermektedir. Bu, hareketli nesnenin tamamını değil, sadece dış hatlarını oluşturmaktadır. Sadece hareketin var ya da yok olduğu bilgisini edinmek için en kolay ve en performanslı yoldur. Bunun dışında hareketli nesnenin fiziksel özellikleri, boyutu gibi verilerde diğer teknikler kadar başarılı değildir.



Şekil 4.13 Ardışık kare farkı ile elde edilen hareketli nesne çıkarımı

Performans Ölçümleri:



Şekil 4.14 Ardışık kare farkı yöntemi performans ölçümü

Yapılan ölçümlerde test videosunda ardışık kare farkı yöntemi ile hareketli nesne takibinin her kare için ortalama 10.0144 ms süre aldığı gözlenmiştir. Video üzerinde gerçek zamanlı işlemler için ortalama gereken süre 66.66 ms olduğu göz önüne alınırsa bu süre gerçek zamanlı görüntü işleme için uygun bir süredir.

4.3.2 PFinder arka plan modeli kullanılarak hareketli nesne çıkarımının artı eksileri ve performans değerlendirme

Hareketli nesne çıkarımında arka plan modeli kullanmanın en önemli avantajı, ardışık kare farkı yönteminin aksine hareketli nesnenin tamamının elde edilebilmesidir (Şekil 4.15). Bunun yanında PFinder yönteminin arka plan modelindeki artılarından olan dinamik olarak arka plan modelini güncellemesi, hareketli nesne çıkarımında da bir artıdır.



Şekil 4.15 PFinder yöntemi ile elde edilen hareketli nesne çıkarımı

PFinder arka plan modelleme yöntemindeki α değerinin düşük seçilmesi arka plan modelinin dinamik güncellenmesini azaltırken, yüksek seçilmesi her yeni karenin arka plana etkisini artırdığından, arka plan her karenin değerleri α katsayısınca yansımaktadır. Bu da hareketli nesne çıkarımında bozulmalar oluşmasına sebep olur. Şekil 4.16, 4.17, 4.18'de aynı video görüntüsünün sırasıyla α=0.025, 0.050, 0.100 değerleri için oluşan hareketli nesne çıkarımları gösterilmektedir. α değeri arttıkça hareketli nesne çıkarımında bozulmalar yaşanmaktadır.



Şekil 4.16 α = 0.025 alınarak hesaplanmış PFinder arka plan çıkarımı

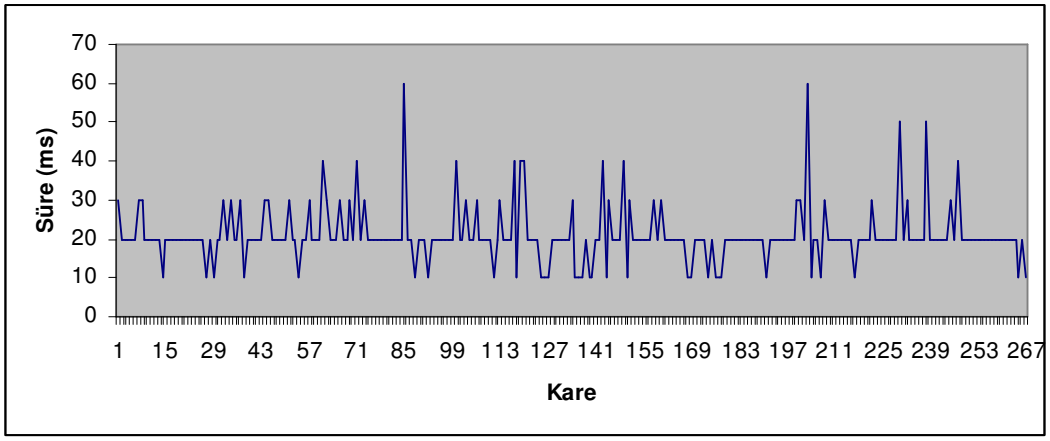


Şekil 4.17 α = 0.050 alınarak hesaplanmış PFinder arka plan çıkarımı



Şekil 4.18 α = 0.100 alınarak hesaplanmış PFinder arka plan çıkarımı

Performans Ölçümleri:



Şekil 4.19 PFinder yöntemi ile performans ölçümü

Ölçümlerde bir karenin PFinder arka plan yöntemi ile arka plan modeli oluşturulması ve bu modelle var olan karenin karşılaştırılması, ortalama 21.11651 ms süre almıştır. Video üzerinde gerçek zamanlı işlemler için ortalama gereken süre 66.66 ms olduğu göz önüne alınırsa, bu süre gerçek zamanlı görüntü işleme için uygun bir süredir.

4.3.3 HRR arka plan modeli kullanılarak hareketli nesne çıkarımının artı eksileri ve performans değerlendirme

HRR arka plan modelleme yönteminin değişikliklere karşı dinamik olarak arka planı güncelleme özelliği olmadığından bu yöntemi kullanarak hareketli nesne çıkarımı yapıldığında da ortamın dinamik değişikliklerine nesne çıkarımında cevap veremeyecektir. Örneğin arka planda var olan bir nesne ortamdan çıkarıldığında Şekil 4.22'de de görüldüğü gibi arka plan modeli tekrar güncellenene kadar ortamdan çıkarılan nesne hareket ediyor olarak saptanır. Arka plan modeli güncellendiğinde nesne hareket ediyor olarak gözükmez. HRR arka plan modelleme yöntemi öğrenme

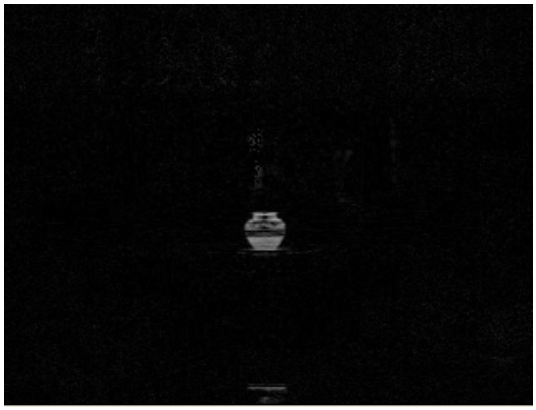
sürecinden sonra arka plan modeli oluşturduğu için, bu süreç içerisinde hareketli nesne çıkarımı yapılamamaktadır.



Şekil 4.20 HRR arka plan modeli



Şekil 4.21 Videodan alınan kare

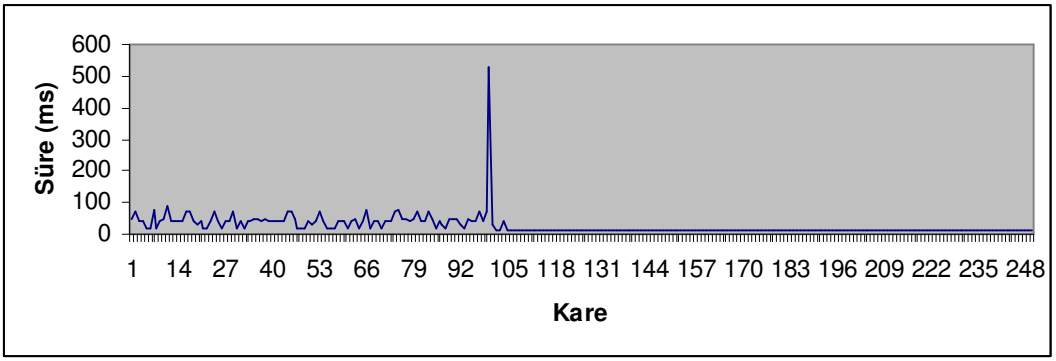


Şekil 4.22 HRR Hareketli nesne çıkarımı

Şekil 4.21'de bir karesi alınmış olan örnek video görüntüsünden Şekil 20'deki arka plan modeli oluştuktan sonra arka plan modeline dahil olan bir nesne ortamdan çıkarılmıştır. Nesne ortamdan çıktıktan sonra arka plan modeli güncellenmediği için Şekil 4.22'deki hareketli nesne çıkarımında bu nesne hareket ediyor olarak gözükmektedir.

Buna karşın HRR yöntemi kullanılarak oluşturulan arka plan modeli PFinder'a göre daha hatasız olduğundan, PFinder ile hareketli nesne çıkarımında yaşanan bozulmalar bu yöntemde yaşanmamaktadır. Ortam değişikliği olmadığında hareketli nesne çıkarımını en başarılı olarak yapmaktadır.

Performans Ölçümleri:



Şekil 4.23 HRR Performans ölçümleri

Yapılan ölçümlerde öğrenme süreci 100 kare olarak alınmış HRR arka plan modelleme yöntemi ile hareketli nesne çıkarımının ortalama olarak 25.19623 ms olduğu ölçülmüştür. Video üzerinde gerçek zamanlı işlemler için ortalama gereken süre 66.66 ms olduğu göz önüne alınırsa, bu süre gerçek zamanlı görüntü işleme için uygun bir süredir.

İlk 100 kare boyunca HRR modeli için veri toplanmış, 100. kareden sonra hareketli nesne çıkarımı yapılmıştır. 100. karede geçen 530.7632 ms HRR arka plan modelinin oluşturulduğu süreci göstermektedir.

4.4 Hareketli Nesne Çıkarımında Kullanılan İyileştirme Teknikleri

Bölüm 4'de anlatılan hareketli nesne çıkarım yöntemleriyle elde edilen veri; her yeni karenin arka plandan ya da bir önceki kareden piksel değeri olarak farklarıdır. Bu veri; Şekil 24'de görüldüğü gibi hareketli nesne ile birlikte video sıkıştırma yöntemlerinden oluşan farkları, arka plan modelleme yönteminden oluşan istenmeyen verileri içerebilir. Bu tür istenmeyen verilerden hareketli nesneyi ayırt etmek için çeşitli görüntü işleme filtreleri kullanmak gerekir.



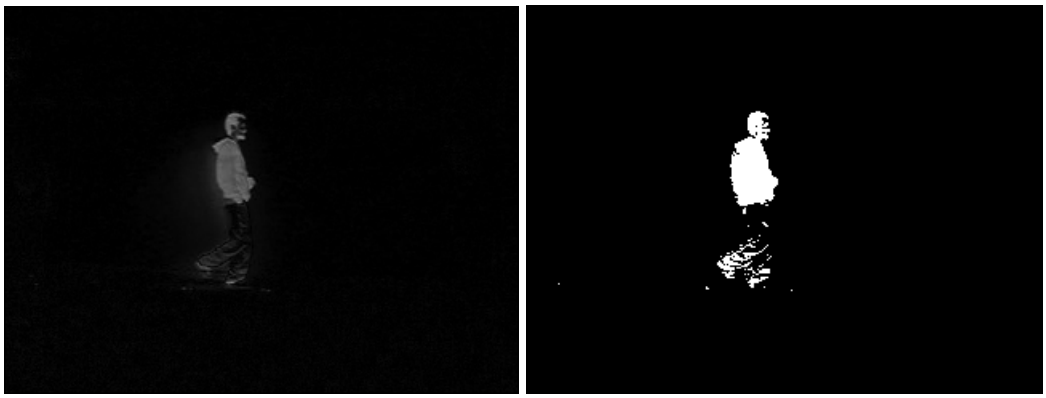
Şekil 4.24 PFinder yöntemiyle oluşan hareketli nesne çıkarımındaki hatalar

4.4.1 Eşik filtresi

Eşik filtresi, görüntü bölütlemenin en basit yöntemlerinden biridir. Seçilen eşik değerinin üzerinde olan pikseller 1, altında olan pikseller ise 0 değerini alacak şekilde ikili hale getirilir. Bu filtre ile hareketli nesne çıkarım metotları ile elde edilen çıktılar daha belirgin hale getirilmektedir.



Filtreden Önce Filtreden Sonra



Filtreden Önce Filtreden Sonra

Şekil 4.25 Hareketli nesne çıkarım sonuçlarının filtre uygulanmış halleri

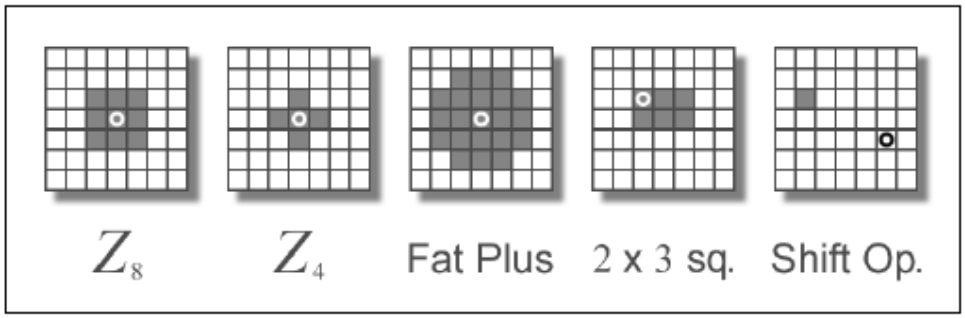
Şekil 4.25'de hareketli nesne çıkarım yöntemlerinden elde edilen çıktılarına eşik değeri 30 olarak alınan eşik filtresi uygulanmıştır. Filtre sonucu elde edilen veri ikili veri olduğundan bilgisayar tarafından daha kolay yorumlanacak haldedir.

4.4.2. Morfolojik filtreler

Matematiksel Morfoloji; görüntünün karakteristiğini koruyarak görüntü verisini basitleştiren / sadeleştiren kuramsal modeller bütünüdür [Haralick, 1987] . Morfolojik filtreler ikili görüntüler için tanımlanmış olsa da siyah beyaz görüntüler üzerinde de uygulamaları mümkündür.

Matematiksel morfolojinin temelleri Minkowski küme teoremine dayanmaktadır. Görüntü işlemede; görüntü iyileştirme, görüntü bölütleme, kenar çıkarımı, gürültü temizleme, şekil analizi, görüntü restorasyonu gibi işlemlerde matematiksel morfoloji kullanılmaktadır [Peters II, 2007].

Morfolojik işlemlerde **yapılandırma elemanı** olarak adlandırılan ufak boyutta matris kullanılmaktadır. Bu matris kayan pencere olarak kullanılıp, merkez noktasındaki piksele komşulukları belirtmektedir.



Şekil 4.26 Sık kullanılan yapılandırma elemanları

Şekil 4.26'da sık kullanılan yapılandırma elemanları gösterilmiştir. Bu şekillerdeki yuvarlak noktalar matrisin merkezini belirtmektedir.

Temel morfolojik filtreler; erozyon, genişleme ve bunların kullanımıyla oluşan açma ve kapama filtreleridir. Hareketli nesne çıkarım yöntemlerinden elde edilen çıktılar, ikili görüntüye dönüştürüldükten sonra hareketli nesne olarak belirlenmiş çok küçük alanların bu görüntüden temizlenmesi gerekir. Bu tür gereksinimler için ikili hale getirilmiş hareketli nesne çıktısı üzerine morfolojik filtreler uygulanabilir.



Şekil 4.27 Örnek ikili görüntü

4.4.1.1 Erozyon

İkili görüntülerde erozyon işlemi beyaz alanları yumuşatmak amacıyla kullanılmaktadır. (4.3)'deki eşitlikte erozyon işlemi gösterilmektedir. Eşitlikte, yapılandırma elemanı olarak Z8 matrisi alındığında giriş pikselinin komşu piksellerinden en az biri bile 0 değerine sahipse, çıkış pikselinin değeri 0 olarak atanmaktadır.

$$E(A,B)(r,s) = \min_{(j,k)\in B} (A(r+j, s+k) - B(j,k)) \quad (4.3)$$

A: Erozyon filtresi uygulanacak görüntü

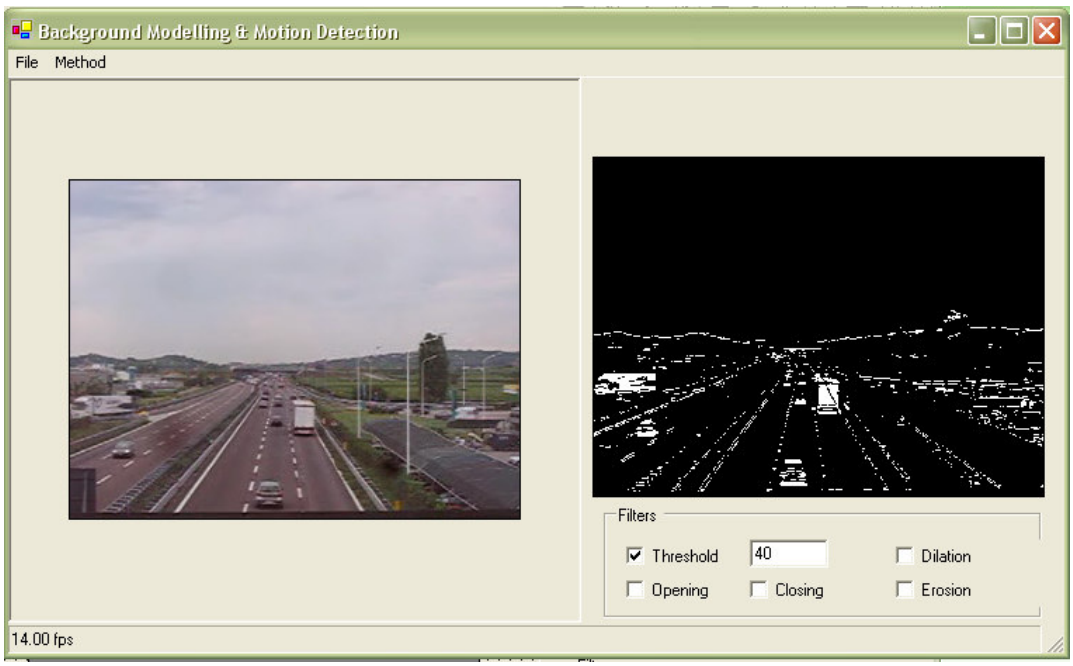
B: Yapılandırma matrisi

Şekil 4.28'de Şekil 4.27'deki örnek ikili görüntüye erozyon filtresi uygulanması sonucu elde edilen çıktı bulunmaktadır.

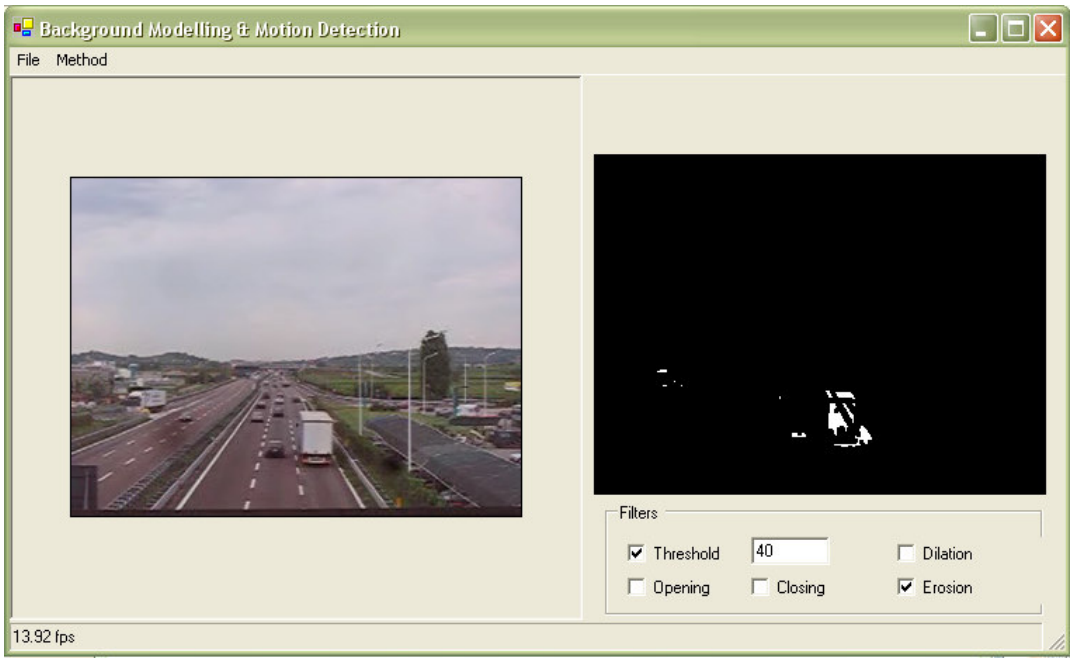


Şekil 4.28 Erozyon filtresi uygulanmış örnek görüntü

Erozyon filtresi, hareketli nesne çıkarım yöntemlerinden elde edilen çıktılar üzerine uygulandığında, hareketli nesne olarak belirlenmiş alanlar erozyona uğratılarak küçük alanlar göz ardı edilmiş olur. Şekil 4.29'da sabit kamera ile alınan görüntüden HRR yöntemi ile elde edilen hareketli nesne çıkarımı gösterilmiştir. Bu görüntüdeki kamera uygun olmayan çevre koşullarından dolayı (titreşim, rüzgar vs.) hareket etmiş ve hareketli nesne çıkarımında sabit alanlar hareket ediyor olarak algılanmıştır. Şekil 4.30'daki görüntüde elde edilen hareketli nesne çıkarımına erozyon filtresi uygulanmış, kameranın hareketinden dolayı saptanan alanlar göz ardı edilip, hareketli büyük nesnelerin saptanması yapılmıştır.



Şekil 4.29 HRR ile elde edilen hareketli nesne çıkarımı



Şekil 4.30 HRR ile elde edilen hareketli nesne çıkarımına erozyon filtresi uygulanması

4.4.1.2 Genişleme

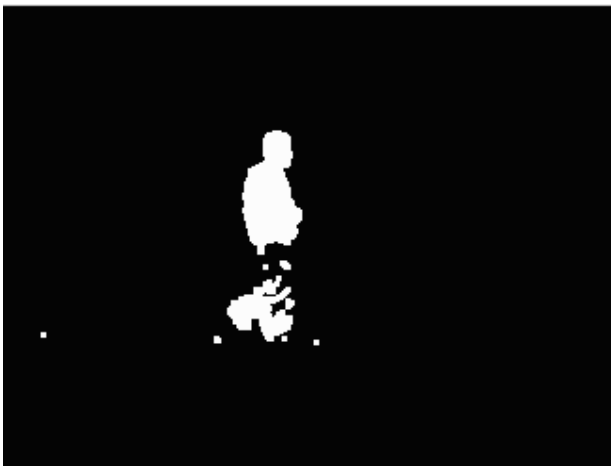
Genişleme işlemi, erozyon işleminin tersine siyah alanları yumuşatmak amacıyla kullanılmaktadır. (4.4)'deki eşitlikte genişleme işlemi gösterilmektedir. Yapılandırma elemanı olarak Z8 matrisi alındığında giriş pikselinin komşu piksellerinden en az biri bile 1 değerine sahipse, çıkış pikselinin değeri 1 olarak atanmaktadır.

$$D(A,B)(r,s) = \max_{(j,k)\in B} (A(r-j, s-k) + B(j,k)) \quad (4.4)$$

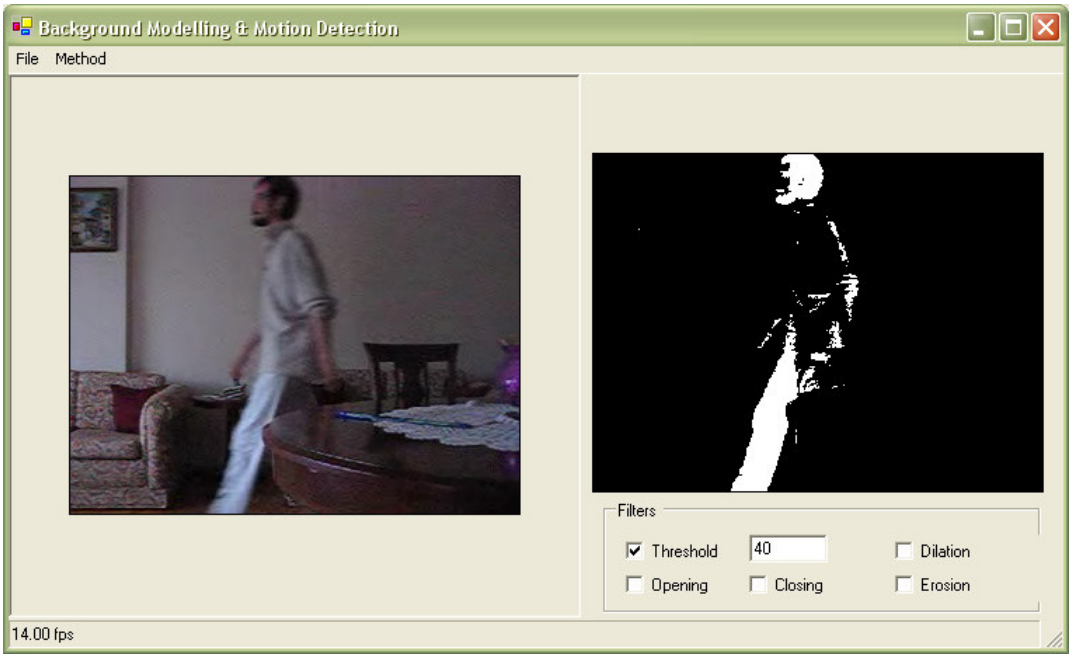
A: Erozyon filtresi uygulanacak görüntü

B: Yapılandırma matrisi

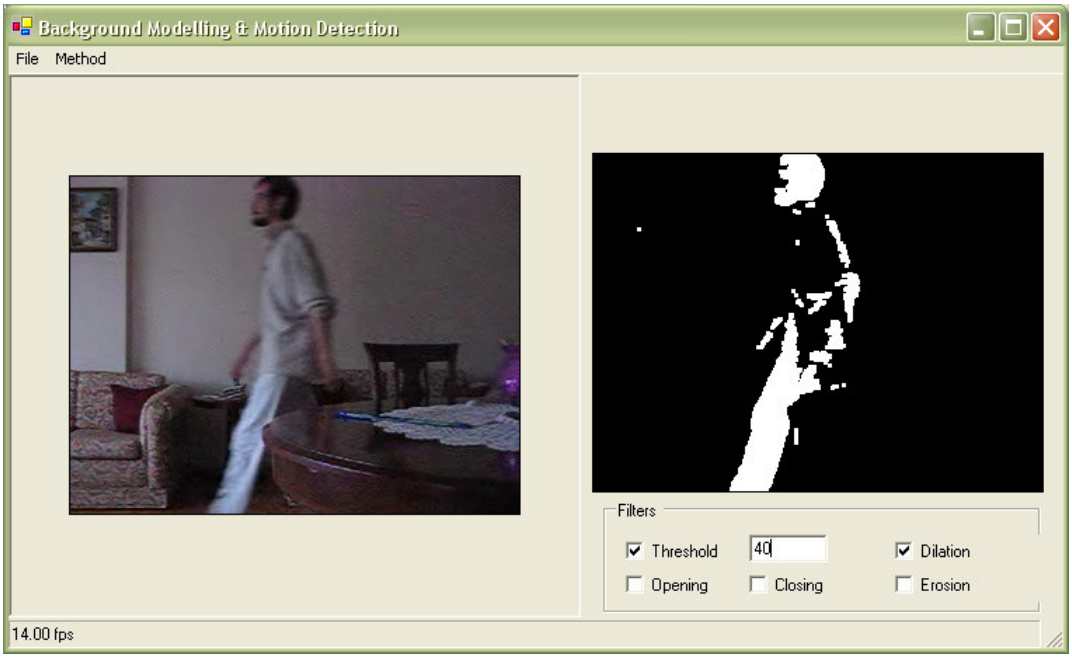
Şekil 4.29'da Şekil 4.27'deki örnek ikili görüntüye genişleme filtresi uygulanması sonucu elde edilen çıktı bulunmaktadır.



Şekil 4.31 Genişleme filtresi uygulanmış örnek görüntü



Şekil 4.32 HRR ile elde edilen hareketli nesne çıkarımı



Şekil 4.33 HRR ile elde edilen hareketli nesne çıkarımına genişleme filtresi uygulanması

Genişleme filtresi, hareketli nesne çıkarım yöntemlerinden elde edilen çıktılar üzerine uygulandığında hareketli nesne olarak belirlenmiş alanlar bu filtre ile

genişletilerek hareketli nesnenin tümüne ulaşılmaya çalışılır. Şekil 4.32'deki örnek hareketli nesne çıktısında hareketli nesnenin bazı alanlarının gri değerleri arka plan modeline yaklaşık değerlere sahip olduğundan hareketli nesne tam olarak çıkartılamamış, parçalı olarak çıkarılmıştır. Şekil 4.33'de ise genişleme filtresi ile elde edilen hareketli nesne çıkarımı sonucundaki beyaz alanlar genişletilerek parçalı alanlar birleştirilerek, olması gereken hareketli nesne çıkarımına daha çok yaklaşılmıştır.

4.4.1.3 Açma

İkili görüntüye sırasıyla erozyon ve genişleme filtresi uygulanmasıdır. Bu şekilde ikili resim üzerindeki gürültü azaltılabilir. Erozyon filtresine benzemekle birlikte beyaz alanlar üzerinde yapılacak aşındırma ihtiyacı daha az olduğunda kullanılır. Şekil 4.34'de örnek ikili görüntüye açma filtresi uygulanmıştır. Elde edilen sonuç Şekil 4.28'deki erozyon filtresinin sonucuna göre daha az aşınmıştır.



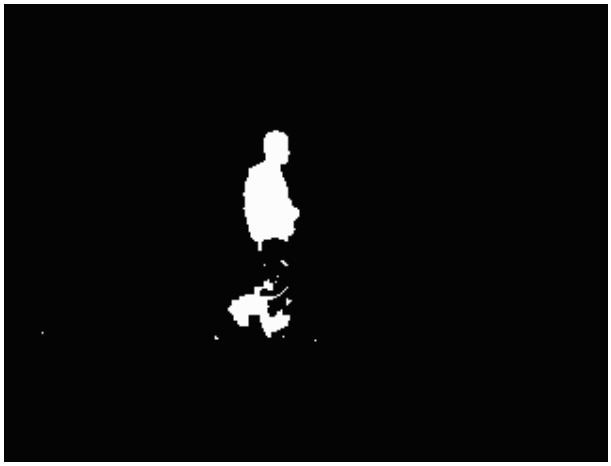
Şekil 4.34 Açma filtresi uygulanmış örnek görüntü

Erozyon filtresinin kullanılması gerektiği durumlarda erozyon filtresi yerine açma filtresi de kullanılabilir. Açma filtresi sırasıyla erozyon ve genişleme filtrelerini uyguladığından, çıktısı erozyon filtresinin beyaz alanlarının genişletilmiş hali olacaktır. Beyaz alanların aşındırmasının daha düşük düzeyde tutulmasının istendiği ya da

aşındırma sonrası elde edilen beyaz alanların genişletilmesi istendiğinde açma filtresi kullanılmalıdır.

4.4.1.4 Kapama

İkili görüntüye sırasıyla genişleme ve erozyon filtrelerinin uygulanmasıdır. Kapama filtresinin çıktısı genişleme filtresine benzemekle birlikte, filtre sonrası kapama filtresinde genişleme sonrası erozyon filtresi uygulandığından, beyaz alanlar genişleme filtresinin çıktısına göre daha aşınmıştır. Genişleme filtresinin gerektiği fakat genişleme işleminin daha az düzeyde tutulması istendiğinde kapama filtresi kullanılabilir. Şekil 4.35'deki örnek ikili görüntüye kapama filtresi uygulanmıştır. Elde edilen sonuçtaki beyaz alanlar Şekil 4.31'deki genişleme uygulanan görüntüye göre daha azalmıştır.



Şekil 4.35 Kapama filtresi uygulanmış örnek görüntü

SONUÇ

Bu tez çalışmasında dijital video dosyaları üzerinde arka plan modelleme yöntemleri ve bu yöntemleri kullanan hareketli nesne çıkarım yöntemleri incelenmiş, yöntemlerin birbirine göre üstünlükleri ve eksikleri saptanmıştır. Var olan yöntemler incelenirken gerçek zamanlı işlem yapabilmesi, var olan arka plan değişikliğine göre modelin güncellenmesi gibi kıstaslara uyum göstermesi göz önüne alınmıştır.

Arka plan modelleme yöntemlerinden PFinder metodu arka plan modelinin dinamik olarak güncellemesi bakımından HRR yöntemine göre üstünlük sağlasa da HRR yöntemi ile elde edilen arka plan modelindeki hatalar PFinder'a göre daha azdır. Performans açısından PFinder yöntemi sürekli aynı kaynak tüketimini gerçekleştirirken HRR yöntemi öğrenim sürecinde ve süreç sonunda tüketim PFinder'a göre fazla, öğrenim sonrası kaynak tüketimi PFinder'a göre daha azdır.

Hareketli nesne çıkarım yöntemlerinde ardışık kare farkı yöntemi ve arka plan modelleme yönteminde oluşturulan arka plan modelleri kullanılarak nesne çıkarımı yapılmıştır. Bu yöntemlerde de PFinder dinamik değişikliklere yanıt verme açısından daha iyiyken HRR, hareketli nesne çıkarımında daha düzgün sonuçlar vermektedir. Ardışık kare farkı yöntemi, nesne çıkarımında düzgün sonuçlar vermese de düşük kaynak tüketimi açısından diğer iki yöntemden üstündür. Her üç yöntemden elde edilen sonuçlara morfolojik filtreler ile iyileştirme sağlanabilmektedir.

SÖZLÜK

Kısaltma	Türkçe	İngilizce
	Açma Filtresi	Opening Filter
AVI		Audio Video Interleave
	Akış	Streaming
	Arka Plan Modelleme	Background Modelling
	Arka Plan Çıkarımı	Background Subtraction
	Bilgisayar Görüş	Computer Vision
DCT	Ayrık Kosinüs Dönüşümü	Discrete Cosine Transform
HRR	En Çok Tekrarlanma Oranı	Highest Redundancy Ratio
	Erozyon Filtresi	Erosion Filter
	Eşik Fitresi	Threshold Filter
	Genişleme Filtresi	Dilation Filter
	Görüntü Bölütleme	Image Segmentation
IFF	Bilgi Alış Veriş Dosya Formatı	Interchange File Format
ISDN	Tümleşik Sayısal İletişim Ağı	Integrated Services Digital Network
ITU	Uluslar arası Haberleşme Birliği	International Telecommunication Union
	Kare	Frame
	Kapama Filtresi	Closing Filter
RIFF	Kaynak Bilgi Alış Veriş Dosya Formatı	Resource Interchange File Format
	Tümevarım	Bottom-Up
	Yapılandırma Elemanı	Structring Element

KAYNAKLAR

R.T. Collins, A. J. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for Cooperative Multi sensor Surveillance, Proceeding of IEEE, Vol. 89. No.10, 2001.

I. Haritaoglu, D. Harwood, L.S. Davis, W4: Real-Time Surveillance of People and Their Activities, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No.8, 2000.

A. Bobick and J. Davis, The Recognition of Human Movements Using Temporal Templates, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No.3, March 2001.

O. Javed and M. Shah, Tracking and Object Classification for Automated Surveillance, ECCV`2002, European Conference on Computer Vision, Copenhagen, Denmark, 2002.

C. Wren, A. Azabayejani, T. Darrell and A. Pentland: Pfinder: Real-time tracking of the human body IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 780-785.

M. Ekinci, E.Gedikli Turk J Elec Engin, VOL.13, NO.2 2005, TUBITAK
Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance

I. Haritaoglu, M. Flickner, Detection and Tracking of Shopping Groups in Stores, Proceeding of the 2001 IEEE Computer Vision and Pattern Recognition, Vol. 1, 8-14 December, 2001.

D. Taşkın, Sıkıştırılmış Video Akımının Düzensiz Haritalar ve Başlangıç Kodlarına Dayalı Şifrelenmesi
Trakya Üniversitesi, Doktora Tezi - 2007

P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-Based Probabilistic Tracking, Proc. of European Conference on Computer Vision, Copenhagen, 27 May- 2 June 2002, Denmark.

P.L. Rosin, T. Ellis, Image Difference Threshold Strategies and Shadow Detection, in Proceeding. British Machine Vision Conference, 1995.

M. Ekinci, F. W. Gibbs, B. T. Thomas, Knowledge-Based Navigation for Autonomous Road Vehicles, Turkish Journal of Electrical Engineering and Computer, Vol. 8, No. 1, 2000.

J. Vass, K. Palaniappan, X. Ahuang, Automatic Spatio-Temporal Video Sequence Segmentation, in Proceeding IEEE International Conference on Image Processing, 1998.

K. Toyama, J. Krumn, B. Brumit, B. Meyers, Wallflower: Principles and Practice of Background Maintenance, 7th IEEE International. Conference on Computer Vision, November, 1999.

W. Grimson, C. Stau_er, R. Romano, L. Lee, Using Adaptive Tracking to Classify and Monitor Activities in a Site, in Proceeding of IEEE Conference on Computer Vision and Recognition, 1998.

A. Kirillov, AForge.Net Image Processing Framework for .Net http://code.google.com/p/aforge/

R. A. Peters II, Lecture Notes on Matematical Morphology Binary Images Department of Electrical Engineering and Computer Science Fall Semester 2007

Haralick, R.M., Sternberg, S.R. and Zhuang, X. (1987). Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9, 532-550.

Microsoft Corp. DV Data in the AVI File Format Specification, Version 1.01, June 25 1997

Kohtaro Asai and Fuminobu Ogawa, MPEG Coding Technologies Technical Report, Mitsubishi Electric, December 1998

The International Telegraph and Telephonne Consultative Comitee, Line Transmission on Non-Telephone Signals / Video Codec for Audiovisual services at p X 64kbit/s, Geneva, 1990

TEŞEKKÜR

Bu çalışmanın hazırlanmasında bana yol gösteren, destek ve yardımlarını esirgemeyen danışman hocam Yrd. Doç. Dr. Nurşen SUÇSUZ'a, değerli hocalarıma ve aileme teşekkürlerimi sunarım.

ÖZGEÇMİŞ

30.01.1983'de İstanbul'da doğdum. Lise eğitimimi Sakıp Sabancı Anadolu Lisesi'nde tamamladım. 2001-2005 yılları arasında Trakya Üniversitesi Bilgisayar Mühendisliği Bölümü'nde lisans eğitimimi tamamladım. 2006 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı'nda yüksek lisansa başladım. 2006 yılından bu yana Idea Teknoloji Çözümleri şirketinde yazılım geliştirme yöneticisi olarak çalışmaktayım.