

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ŞEKİL EŞLEMEYE DAYALI NESNE TAKİBİ YÖNTEMİ KULLANARAK
VIDEO KODLAMA**

Seyit TUNÇ

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA

2010

Her hakkı saklıdır

TEZ ONAYI

Seyit TUNÇ tarafından hazırlanan “**Şekil Eşlemeye Dayalı Nesne Takibi Yöntemi Kullanarak Video Kodlama**” adlı tez çalışması 15.07.2010 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : *Yrd. Doç. Dr. Hakkı Alparslan ILGIN*

Ankara Üniversitesi, Elektronik Mühendisliği Anabilim Dalı

Jüri :

Başkan : *Yrd. Doç. Dr. Süleyman TOSUN*

Ankara Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı

Üye : *Yrd. Doç. Dr. Hakkı Alparslan ILGIN*

Ankara Üniversitesi, Elektronik Mühendisliği Anabilim Dalı

Üye : *Yrd. Doç. Dr. Fikret ARI*

Ankara Üniversitesi, Elektronik Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Orhan ATAKOL

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

ŞEKİL EŞLEMEYE DAYALI NESNE TAKİBİ YÖNTEMİ KULLANARAK VİDEO KODLAMA

Seyit TUNÇ

Ankara Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Hakkı Alparslan ILGIN

Video sıkıştırma algoritmalarında hareket kestirimi, sıkıştırma kalitesine etki eden önemli bir aşamadır. Bu tezde önerilen hareket kestirim algoritması nesnenin şeklinin takip edilmesine dayanmaktadır. Yöntemde, öncelikli olarak video çerçevelerinde yer alan hareketli nesnelerin şekilleri çıkartılmaktadır. Daha sonra bu şekiller üzerinde öznitelikler çıkartılarak nesne bölgelere ayrılmaktadır. Bölgedeki öznitelik sayısına bağlı olarak kullanılacak hareket modeli belirlenmektedir. Hareket modellerinin belirlenmesinin ardından hareket parametreleri kestirilmektedir. Hareket dengelemesi kısmında ise, nesne üzerine hareket uygulanmakta ve boşluklar oluşmuşsa çift doğrusal ara değerlendirme ile bu boşluklar kapatılmaktadır. Elde edilen görsel kalite ise bloğa dayalı hareket kestirimi yöntemleriyle karşılaştırılmaktadır.

Temmuz 2010, 77 sayfa

Anahtar Kelimeler: Video kodlama, hareketli nesne takibi, hareketli nesne tespiti, video nesne düzlemi, arka plan çıkarımı, öznitelik çıkarımı, hareket kestirimi, parametre kestirimi, hareket modelleri, ara değerlendirme

ABSTRACT

Master Thesis

VIDEO CODING USING A SHAPE MATCHING BASED OBJECT TRACKING METHOD

Seyit TUNÇ

Ankara University

Graduate School of Natural and Applied Sciences

Department of Electronic Engineering

Supervisor: Asst. Prof. Dr. Hakkı Alparslan ILGIN

Motion estimation is an important process which affects compression quality in video compression algorithms. In this thesis the proposed motion estimation algorithm is based on tracking of object shape. In the method, shapes of moving objects in the video frames are firstly extracted. Then the moving objects are segmented to the regions by extracting features on the object. Depending on the number of the features in the region the motion model of the region is determined. By determination of the motion models, motion parameters are estimated. In the motion compensation stage, the motion model is applied on the region and if holes are occurred, they are closed by bilinear interpolation. The obtained visual quality is compared with block based motion estimation methods.

July 2010, 77 pages

Key Words: Video coding, moving object tracking, moving object detection, video object plane, background subtraction, feature extraction, motion estimation, parameter estimation, motion models, interpolation

TEŞEKKÜR

Öncelikle danışman hocam Sayın Yrd. Doç. Dr. Hakkı Alparslan ILGIN'a yüksek lisans programı boyunca beni her daim motive ettiği ve yönlendirdiği, bilgisini ve değerli görüşlerini bana aktardığı için çok teşekkür ederim.

Tez konusunu seçmemde ve The University of Surrey'de araştırma yapma imkanı bulmamda büyük katkısı olan Prof. Ahmet KONDOZ'a, The University of Surrey'de geçirdiğim dönem içerisinde desteğini esirgemeyen Dr. Şafak DOĞAN'a minnetimi sunarım.

Tez çalışmamı destekleyen başta birim amirim Bülent SEMERCİ olmak üzere anlayışlı tutumlarından dolayı tüm çalışma arkadaşlarıma teşekkür ederim.

Test görüntülerini elde etmemde büyük destekleri olan abim Fatih TUNÇ'a, kuzenlerim Furkan ÖZBAY'a ve Semih Sercanay AKTAŞ'a; değerli yorumları ve bilgisiyle tezime yön vermemde yardımcı olan arkadaşım Emre TAŞKIRAN'a; tezdeki şekilleri oluşturmamda desteğini esirgemeyen grafiker arkadaşım Ahmet KÖKSAL'a ve tez çalışmam boyunca her türlü desteği benden esirgemeyen Aslı COŞKUN'a şükranlarımı sunarım.

Yüksek lisans eğitimim süresince verdiği maddi destekten ötürü TÜBİTAK BİDEB'e teşekkür ederim.

Son olarak, yaşamım boyunca yanımda yer alan, ihtiyacım olduğu her an desteklerini arkamda hissettiğim, tüm eğitim hayatım süresince fedakarlıklarını esirgemeyen başta babam olmak üzere tüm aileme teşekkürü bir borç bilirim.

Seyit TUNÇ

Ankara, Temmuz 2010

KISALTMALAR DİZİNİ

AAC	Gelişmiş Ses Kodlama
AKD / DCT	Ayrık Kosinüs Dönüşümü
ARPS	Uyarlamalı Çarmıh Örüntü Arama
ASIFT	İlgin-Ölçek Değişimsiz Öznitelik Dönüşümü
ASP	Gelişmiş Basit Profil
AVC	Gelişmiş Video Kodlama
CABAC	İçerik Tabanlı Uyarlamalı İkili Aritmetik Kodlama
CAVLC	İçerik Tabanlı Uyarlamalı Değişken Uzunluk Kodlama
dB	Desibel
DoG	Gaus Farkı
DS	Baklava Arama
FSS	Dört Adım Arama
GOP	Resimler Grubu
HSV	Renk Tonu, Doygunluk, Parlaklık Değeri
ISO	Uluslar arası Standartlar Organizasyonu
ITU-T	Uluslararası Telekomünikasyon Birliği Telekomünikasyon Standartlaştırma Birimi
JPEG	Birleşik Fotoğraf Uzmanları Grubu
JVT	Ortak Video Takımı
LCL	Kayıpsız Kodek Kütüphaneleri
MAD	Mutlak Farkın Ortalaması
MB	Makro Blok
MPEG	Hareketli Görüntü Uzmanlar Grubu
MSE	Karesel Hatanın Ortalaması

MSER	Azami Kararlı Uç Bölge
MV	Hareket Vektörleri
NTSS	Yeni Üç Adım Arama
PSNR	Tepe Sinyal Gürültü Oranı
RGB	Kırmızı, Yeşil, Mavi
SAD	Mutlak Farkın Toplamı
SES	Basit ve Etkili Arama
SIFT	Ölçek Değişimsiz Öznitelik Dönüşümü
TDA / SVD	Tekil Değer Ayrışımı
TSS	Üç Adım Arama
VCEG	Video Kodlama Uzmanlar Grubu
VO	Video Nesnesi
VOP	Video Nesne Düzlemleri

ŞEKİLLER DİZİNİ

Şekil 2.1 Video kodlama yapısı	3
Şekil 2.2 Video dizisinde uzaysal ve zamansal örnekleme	4
Şekil 2.3 Resimler grubu	7
Şekil 2.4 Karma kodlayıcı blok şeması	8
Şekil 2.5 Karma dekodlayıcı blok şeması	8
Şekil 2.6 Mobile & Calendar video dizisi 100. ve 186. çerçeveler	10
Şekil 2.7 Hareket dengelemesi	13
Şekil 2.8 Üç tane video nesnesi içeren video sahnesi	17
Şekil 2.9 MPEG-4 video kodlayıcı yapısı	17
Şekil 2.10 MPEG-4 video dekodlayıcı yapısı	18
Şekil 3.1 Aşınma ve genişleme	24
Şekil 3.2 Bağlı bileşen analizi	25
Şekil 4.1 Nesne temsilleri	28
Şekil 4.2 Normalleştirme yöntemleri	31
Şekil 4.3 ASIFT yöntemi	32
Şekil 4.4 Aynı posterin farklı açılardan çekilmiş iki görüntüsü	33
Şekil 4.5 Aynı derginin farklı açılardan çekilmiş iki görüntüsü	34
Şekil 4.6 Hareket modelleri	35
Şekil 5.1 Önerilen yöntemin genel blok şeması	38
Şekil 5.2 Ön plan nesnelerinin bulunması	38
Şekil 5.3 Zamansal ortanca süzgeç	39
Şekil 5.4 Küçük nesnelere yok etmek için geliştirilen yöntemin algoritması	40
Şekil 5.5 Ön plan nesnesinin elde edilmesi	41
Şekil 5.6 Nesnenin bölgelere ayrılmasının blok şeması	41
Şekil 5.7 Ortak noktaların çıkartılması	42
Şekil 5.8 Nesnenin bölgelere ayrılmasının aşamaları	43
Şekil 5.9 Nesnenin bölgelere ayrılması için bir örnek	45
Şekil 5.10 Hareket kestirimi blok şeması	46
Şekil 5.11 Hareket dengelemesi blok şeması	50

Şekil 5.12 Nesnenin tekrar oluşturulması	51
Şekil 6.1 Çerçeve de az yer kaplayan nesnenin öteleme testi görüntüleri	54
Şekil 6.2 Çerçeve de az yer kaplayan nesnenin öteleme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri	55
Şekil 6.3 Çerçeve de az yer kaplayan nesnenin öteleme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri	56
Şekil 6.4 Çerçeve de çok yer kaplayan nesnenin öteleme testi görüntüleri	57
Şekil 6.5 Çerçeve de çok yer kaplayan nesnenin öteleme testinde elde edilen PSNR değerleri	58
Şekil 6.6 Çerçeve de çok yer kaplayan nesnenin öteleme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri	59
Şekil 6.7 İzdüşüm testi görüntüleri	60
Şekil 6.8 İzdüşüm testinde ardışık 32 çerçeve üzerinde elde edilen PSNR değerleri	61
Şekil 6.9 İzdüşüm testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri	62
Şekil 6.10 Yanılsama görüntüsü	63
Şekil 6.11 Yüksek çözünürlükte dönme testi görüntüleri	64
Şekil 6.12 Yüksek çözünürlükte dönme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri	65
Şekil 6.13: Yüksek çözünürlükte dönme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri	66
Şekil 6.14 Düşük çözünürlükte dönme testi görüntüleri	67
Şekil 6.15 Düşük çözünürlükte dönme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri	68
Şekil 6.16: Düşük çözünürlükte dönme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri	69

ÇİZELGELER DİZİNİ

Çizelge 2.1: Kodlama ve gösterim sırası	7
Çizelge 2.2: MPEG tarafından geliştirilmiş video kodlama standartlarının özellikleri .	19

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
KISALTMALAR DİZİNİ	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. VİDEO KODLAMA	3
2.1 Giriş	3
2.2 Sayısal Görüntüler ve Video Dizileri	3
2.2.1 Görüntü ve video sıkıştırma	4
2.3 Karma Kodlayıcı-Dekodlayıcı Yapısı	6
2.3.1 Video sıkıştırmada hareket kestirimi ve dengelemesi	9
2.3.1.1 Hareket kestirimi	10
2.3.1.1.1 Bloğa dayalı hareket kestirimi	11
2.3.1.1.2 Nesneye dayalı hareket kestirimi	12
2.3.1.2 Hareket dengelemesi	13
2.3.2 Entropi kodlama	14
2.4 Video Kodlama Standartları	15
2.4.1 MPEG-4 standartları	16
2.4.1.1 MPEG-4	16
2.4.1.2 MPEG-4'ün H.264 ile karşılaştırılması	18
3. HAREKETLİ NESNE TESPİTİ	20
3.1 Giriş	20
3.2 Çerçeve Farkı	20
3.3 Arka Plan Çıkarımı	21
3.3.1 Arka plan çıkarımı algoritmaları	22
3.4 Görüntü İyileştirme	23
4. NESNE TAKİBİ	26
4.1 Giriş	26
4.2 Nesne Temsili	26
4.3 Öznitelik Çıkarımına Dayalı Nesne Takibi	28
4.4 Nesne Takibi için Öznitelik Seçimi.....	29
4.5 Öznitelik Çıkarımı Yöntemlerinin Karşılaştırılması	32
4.6 Hareket Modelleri	34
5. ÖNERİLEN HAREKET KESTİRİMİ VE DENGELEMESİ YÖNTEMİ	37
5.1 Giriş	37
5.2 Ön Plan Nesnelerinin Bulunması	38
5.3 Nesnenin Bölgelere Ayrılması	41

5.4 Hareket Kestirimi	45
5.5 Hareket Dengelemesi	49
6. DENEYSEL BULGULAR	52
7. SONUÇLAR	70
KAYNAKLAR	72
ÖZGEÇMİŞ	78

1. GİRİŞ

Bir video işaretinin kapladığı alanın azaltılması donanım maliyetinden tasarruf sağlarken, düşük bit oranında sıkıştırılarak hızlı iletilmesi ise zamandan kazanç sağlamaktadır. Az yer kaplayan ve sıkıştırma oranı yüksek olan video işaretinin aynı zamanda görüntü kalitesinin yüksek olması izleyen kişinin seyir zevki açısından büyük önem arz etmektedir. Bu sebeple, video işaretlerinin; depolandığı alanların boyutlarının azaltılması, yüksek kalitede izlenebilmesi ve düşük bit oranında sıkıştırılarak hızlı iletilmesi çok önemlidir.

Hareket kestirimi, video dizilerinin zamansal artıklığını düşürmek için video sıkıştırmada kullanılan en önemli aşamadır. H.261 gibi ilk video kodlama standartlarından H.264 gibi en güncel video kodlama standardına kadar, H.120 haricinde, tüm kodlama standartlarında hareket kestirimi kullanılmaktadır. MPEG-1 (ISO/IEC 11172 1993), MPEG-2 (ISO/IEC 13818 1995), MPEG-4 (ISO/IEC 14496-2 2004), H.261 (ITU-T 1993), H.263 (ITU-T 1998) ve H.264'te (ISO/IEC 14496-10 2003) bloğa dayalı hareket kestirimi kullanılmasının yanı sıra MPEG-4'te ayrıca rasgele şekilli nesnelerin hareket kestirimi de desteklenmektedir (Sikora 1997).

Bir hareket kestirim algoritmasının başarısını belirleyen başlıca iki önemli faktör bulunmaktadır. Bunlardan birincisi hesaplama zamanının az olması, ikincisi ise kestirim kabiliyetinin yüksek olmasıdır. Bunlardan ilki gerçek zamanlı kodlama için, ikincisi ise sıkıştırma oranının artırılması için önemlidir. Önerilen yöntemde ise kestirimin iyi olması üzerinde yoğunlaşmaktadır.

Önerilen hareket kestirimi yönteminde, sabit kameradan alınmış bir video dizisinde arka plan modellenerek hareketli ön plan nesnesi/nesneleri bulunmaktadır. Ardışık iki çerçevedeki ön plan nesnesi üzerinde öteleme, ilgin ya da izdüşümsel hareket dönüşümü hesaplanarak hareket kestirimi yapılmaktadır. Bulunan hareket dönüşümünün şimdiki çerçevedeki ön plan nesnesine uygulanmasıyla bir sonraki çerçevedeki ön plan nesnesi elde edilmektedir. Elde edilen ön plan nesnesinin arka plan modeliyle birleştirilmesiyle

de Őimdiki çerçevenin dengelemesi yapılmaktadır. B ylelikle bir sonraki çerçevenin  ng r m  gerçekteŐtirilmektedir.

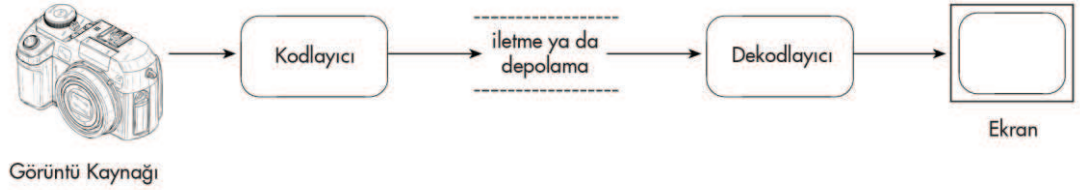
Video kodlama standartlarında kullanılan hareket kestirimi y ntemlerinde sadece  teleme hareket kestirilmektedir.  nerilen y ntemde ise  n plan nesnesinin  teleme hareketinin yanı sıra  lçekleme, d nme ve izd Ő msel hareketleri de kestirilmektedir. B ylece nesnenin hareketi daha iyi modellendiĐi iin hareket kestiriminin baŐarımının da y ksek olması beklenmektedir.

 nerilen hareket kestirimi y ntemi video kodlama standartlarında kullanılabileceĐinden ikinci b l mde video kodlama standartlarından genel olarak bahsedilmektedir. Hareketli nesnenin siluetini elde edebilmek iin kullanılabilecek hareketli nesne tespiti y ntemleri   nc  b l mde yer almaktadır. Nesne silueti  zerinden hareket kestirimi yapabilmek iin uygulanabilecek y ntemlerden ve genel olarak nesne takibi y ntemlerinden d rd nc  b l m olan nesne takibi b l m nde bahsedilmektedir. Sonraki b l mde  nerilen nesneye dayalı hareket kestirimi y ntemi aıklanmaktadır. Altıncı b l mde  nerilen hareket kestirimi y nteminin baŐarısı yedi farklı hareket kestirimi y ntemiyle karŐılaŐtırılmaktadır. Son olarak sonular ise son b l mde verilmektedir.

2. VİDEO KODLAMA

2.1 Giriş

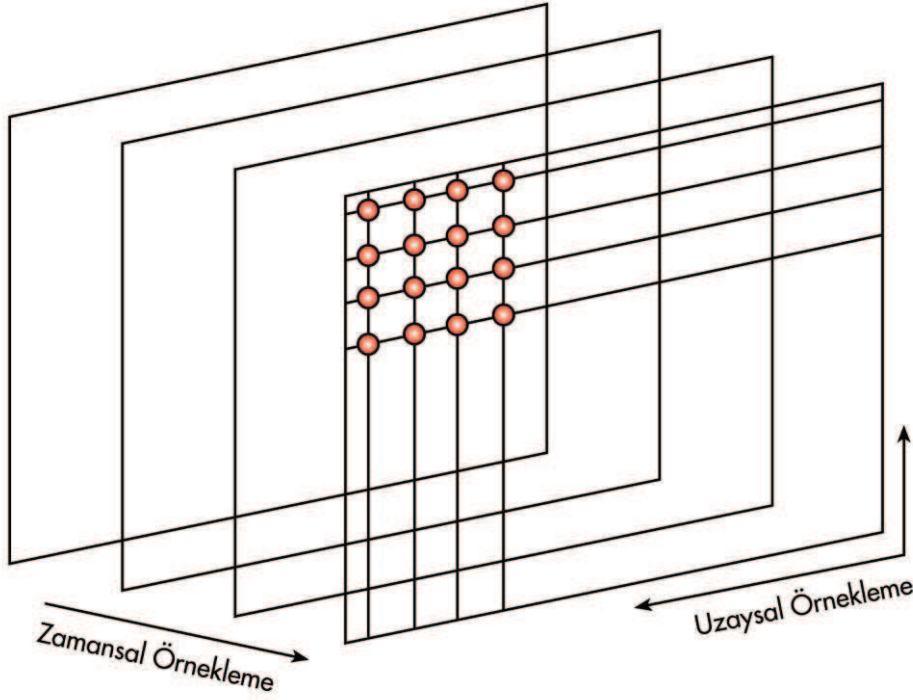
Sayısal video, görsel sahnelerinin uzaysal ve zamansal olarak örneklenip gösterilmesi işlemidir. Örnekleme işlemi sonrasında ya bir çerçeve (frame) ya da birden fazla çerçeveden meydana gelen bir dizi oluşturulmaktadır. Video kodlama ise bu sayısal video işaretlerinin sıkıştırılması ve çözülmesi gibi birbirini tamamlayan iki işlemten meydana gelmektedir. Yapısı Şekil 2.1’de de görüldüğü gibi video kodlamada, görüntü kaynağından alınan görüntü dizisi kodlanarak iletilmekte ya da depolanmaktadır. Daha sonra bu görüntü dizisi dekodlanarak ekranda gösterilmektedir.



Şekil 2.1 Video kodlama yapısı

2.2 Sayısal Görüntüler ve Video Dizileri

Doğal bir görsel sahne zamansal ve uzaysal anlamda süreklidir. Görsel bir sahneyi sayısal olarak temsil etmek için zamansal ve uzaysal olarak örneklemek gerekir. Uzaysal örnekleme kavramını açmak gerekirse; bu örnekleme sonrasında durağan tek bir çerçeve ortaya çıkmaktadır. Ortaya çıkan çerçeve bilgisinde ise resim bileşenleri (Piksel – Picture Element) bulunmaktadır. Başka ifadeyle piksellerin bir araya gelmesiyle çerçeve oluşmaktadır. Zamansal örnekleme ise dikkörgensel görüntü alma işleminin periyodik zaman aralıklarında tekrar etmesidir. Uzaysal örnekleme için t anındaki resmin örnekleme, zamansal örnekleme için ise farklı t anlarındaki resimlerin örnekleme tanımlı yapılabilmektedir. Zamansal-uzaysal örnekleme ile oluşturulan video dizisi Şekil 2.2’de gösterilmektedir.



Şekil 2.2 Video dizisinde uzaysal ve zamansal örnekleme

2.2.1 Görüntü ve video sıkıştırma

Görüntü ve video sıkıştırma, fark edilebilir bir bozulma ya da işlem hatası olmaksızın, görüntü/video verisinin depolamada kapladığı alanın ya da iletimde ihtiyaç duyulan bant genişliğinin azaltılması gibi ortak amaçları paylaşmaktadır. Görüntü sıkıştırma, kayıpsız ve kayıplı olmak üzere iki şekilde yapılmaktadır. Kayıpsız sıkıştırma genellikle pikseller arası korelasyonun düşük olduğu şekil gibi görüntülerde, kayıplı sıkıştırma ise fotoğraf görüntüleri gibi karmaşık ve yüksek korelasyona sahip görüntülerde kullanılmaktadır. Kayıpsız sıkıştırma algoritmalarında, görüntüler, hiç bir bilgi kaybolmaksızın sıkıştırılmaktadır. Bir kayıpsız sıkıştırma algoritmasının verimini, orijinal görüntünün kapladığı alanın sıkıştırılmış görüntünün kapladığı alana oranını veren, sıkıştırma oranı göstermektedir (Santa-Cruz 2000).

Sıkıştırılmamış hali çok büyük olan video dosyaları, kayıpsız yöntemler ile düşük oranlarda sıkıştırılabildikleri için, hareketli görüntü sıkıştırma işlemi çoğunlukla kayıplı yöntemler ile gerçekleştirilir. Bununla birlikte hızlı kodlama ihtiyacı olduğu durumlarda

veya en küçük bir kaybın bile istenmediği profesyonel amaçlı uygulamalar için kayıpsız sıkıştırma da gerekli olabilmektedir. Şimdiye kadar üretilmiş en etkili kayıplı sıkıştırma yöntemi olan H.264/MPEG-4 AVC, istenirse kayıpsız sıkıştırma da yapabilmektedir (ISO 2005).

Kayıpsız hareketli görüntü sıkıştırma yöntemleri genellikle gerçek zamanlı video yakalama gibi hızlı kodlama yapmanın gerekli olduğu durumlarda kullanılmaktadır. Huffyu, SheerVideo, CorePNG, MSU Lossless Video Codec, LCL, Lagarith, SKYUV (Smith 2000), Lossless Motion JPEG gibi bir çok kayıpsız sıkıştırma yöntemi vardır. Pratikte, kayıpsız bir yöntem ile sıkıştırılmış olan bir hareketli görüntü çok yer kapladığı ve saklama ünitelerinin kapasiteleri yetersiz olduğu için, saklama amacıyla daha sonra kayıplı bir yöntem ile tekrar sıkıştırılmaktadır. Son yıllarda üretilen yüksek hızlı işlemciler sayesinde, kayıplı bir yöntem ile de gerçek zamanlı video yakalama işlemleri yapılabilmektedir. Bu nedenle günümüzde hareketli görüntü yakalama işlemlerinde de genellikle kayıplı kodlayıcılar tercih edilmektedir.

İlk video kodlama standardı H.120, o dönemlerdeki adı CCITT olan, ITU-T (International Telecommunication Union Telecommunication Standardisation Sector) tarafından 1984 yılında geliştirilmiştir (Sullivan 2005). Pratik anlamda kayıplı hareketli görüntü sıkıştırma üzerine geliştirilmiş olan ilk standart ise ITU-T tarafından geliştirilen H.261 standardıdır. Ayrık Kosinüs Dönüşümü (AKD, DCT - Discrete Cosine Transform) tabanlı bir video kodlama standardı olan H.261'in ilk sürümü 1990'da, ikincisi 1993'te tamamlanmıştır (ITU-T 1993). H.261, 352 x 288 pikselden oluşan CIF (Common Interchange Format) ve 176 x 144 pikselden oluşan çeyrek CIF büyüklüğündeki QCIF (Quarter CIF) formatlarını desteklemekte ve 64-2048 Kbps aralığında çalışmaktadır. Geliştirilmesinde görüntülü telefon (videotelefon) ve görüntülü konferans (videokonferans) uygulama alanları hedef alınmıştır. ISO'nun geliştirdiği MPEG-1 ve MPEG-2 standartları H.261'i temel almıştır. Videotelefon sistemlerinde, ileri geri sarma gibi işlemler olmadığı için, çerçeveler arası hareket farkı hesaplanması işlemleri daha basittir. Daha basit olması sayesinde daha hızlı kodlanabilir ki bu özellik, gecikme olmaması açısından görüntülü telefon için gerekli bir durumdur. Telekonferans uygulamalarında sıkıştırma algoritmasının hızlı olması, konferanstaki

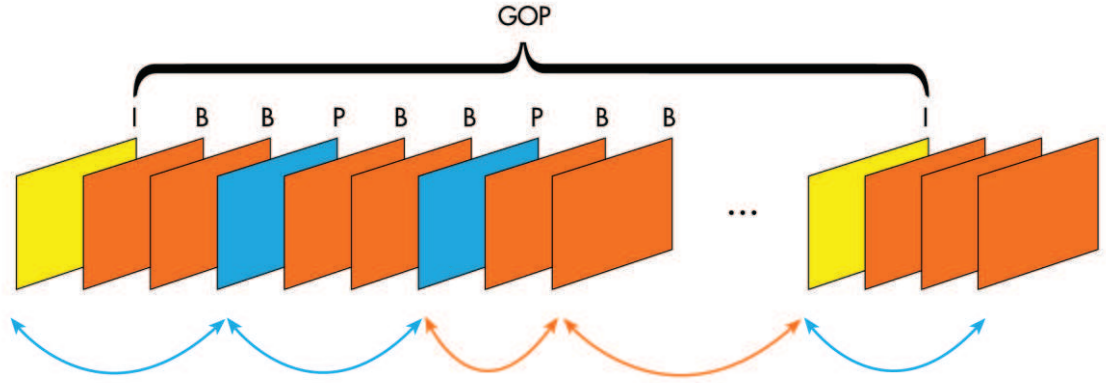
taraflar arasında çok ufak bir gecikmeye sebep olacağı için çok önemlidir. Bu tip uygulamalarda video kalitesi, sıkıştırılmada yaşanacak gecikmeyle kıyaslandığında çok daha az önem arz etmektedir.

H.261'in yerini alan ve 1995 yılında ilk sürümü tamamlanan H.263, sub-QCIF (128 x 96), QCIF, CIF, 4CIF (704 x 576) ve 16CIF (1408 x 1152) video formatlarını desteklemektedir. 1997'de ikinci sürüm (H.263+ olarak bilinir), ve 2000'de üçüncü sürüm (H.263++) tamamlanmıştır. ISO'nun geliştirdiği MPEG-4 standardı, temelde H.263 ile aynı olmasına rağmen ek özellikler de içermektedir. Günümüzde, en iyi sıkıştırma oranı veren standart ise H.264'tür.

2.3 Karma Kodlayıcı-Dekodlayıcı Yapısı

Hareketli görüntü sıkıştırma algoritmalarının temel çalışma prensibi, ardışık çerçeveler arasındaki farkın kodlanmasına dayanmaktadır. Kodlaması yapılacak olan çerçeve, eğer bir sahne değişimi meydana gelmediyse, büyük ölçüde önceki çerçeve ile benzer olmaktadır. O halde, bir çerçevenin tamamını kodlamak yerine, önceki çerçeveden farkını kodlamak daha mantıklıdır. Sıkıştırma algoritmaları I, P ve B olmak üzere üç farklı çerçeve biçimi kullanmaktadır (ISO 1998).

I çerçevesi (intra-frame), çerçevedeki piksellerin tamamının JPEG benzeri bir yöntem ile kodlandığı çerçeve biçimidir ve diğer çerçeve tiplerinden daha fazla yer kaplamaktadır. P çerçevesi (predictive coded frame, inter-frame), bir önceki I veya P çerçevesi ile arasındaki harekete dayalı farkın kodlandığı bir çerçeve biçimidir. B çerçevesi (bidirectionally predictive coded frame) ise, hem bir önceki hem de bir sonraki I veya P çerçevesi ile arasındaki harekete dayalı farkın kodlandığı bir çerçeve biçimidir.



Şekil 2.3 Resimler grubu

Farklı çerçevelerin bir araya gelmesi ile bir resimler grubu (GOP – Group of Pictures) oluşur. GOP, bir hareketli görüntüde rastgele olarak erişilebilecek en küçük birimdir. Şekil 2.3'te de gösterildiği gibi her bir GOP'ta sadece bir tane I çerçevesi bulunmasının yanı sıra diğer çerçeve türlerinden değişik sayılarda olabilmektedir. Bir GOP, ya bir I çerçevesi ile ya da sadece o I çerçevesini temel alarak harekete dayalı tahmin yürüten B çerçeveleri ile başlamaktadır. Bir B çerçevesi kendisinden sonra gelen I veya P çerçevesine bağlı olduğu için, kodlama düzeninde bağlı olduğu çerçeveden sonra yer almalıdır. Bu nedenle kodlama sırası, gösterim sırasından farklıdır. Örnek bir GOP için kodlama sırası ve gösterim sırası Çizelge 2.1'de verilmektedir.

Çizelge 2.1 Kodlama ve gösterim sırası

Gösterim Sırası	1	2	3	4	5	6	7	8	9	10	11	12	13
Çerçeve Tipi	I	B	B	P	B	B	P	B	B	P	B	B	I
Kodlama Sırası	1	3	4	2	6	7	5	9	10	8	12	13	11

işlemler için, hareket vektörleri (motion vector) ve bir artık hata matrisi ile ifade edilmektedir. Inter-MB'lerin artık hatalarına, intra-MB'lere olduğu gibi AKD uygulanmaktadır. Daha sonra, AKD katsayıları nicemlenmektedir. Dönüşümün ardından, enerji dağılımı düşük frekanslı bölgelerde yoğunlaşmaktadır. AKD katsayılarını sıralamak için zigzag tarama (Zigzag Scan) uygulanmaktadır. Böylelikle enerjinin düşük olduğu bölgeler birlikte gruplanabilmektedir. Zigzag taramanın ardından katsayılar, hareket vektörlerinde olduğu gibi entropi kodlanmaktadır.

Hassas kestirim yapabilmek için kodlayıcı tarafında, kodlama verimliliği ve kalite açısından en uygun seçimin yapılması gerekmektedir. Bunun için farklı blok büyüklükleri ve intra ve inter gibi farklı öngörüm modları kullanılmaktadır. Kodlama verimliliği, çıkıştaki bit dizisinin boyu ile ters orantılıdır. Kalite ise, hareket dengelemesi yapılmış MB ile orjinal MB arasında elde edilen ve geriçatım (reconstruction) kalitesinin bir ölçütü olan PSNR (Peak Signal to Noise Ratio) ile belirlenmektedir. Kaliteden ya da kodlama verimliliğinden ne kadar ödün verileceğine mod seçimi aşamasında karar verilmektedir. En uygun öngörüm modu seçilmesinin ardından geriçatılmış MB, referans çerçevesinin tamponuna (buffer) yazılmaktadır. Ayrıca bu sırada, kodlanan bitler çıkış olarak verilmektedir.

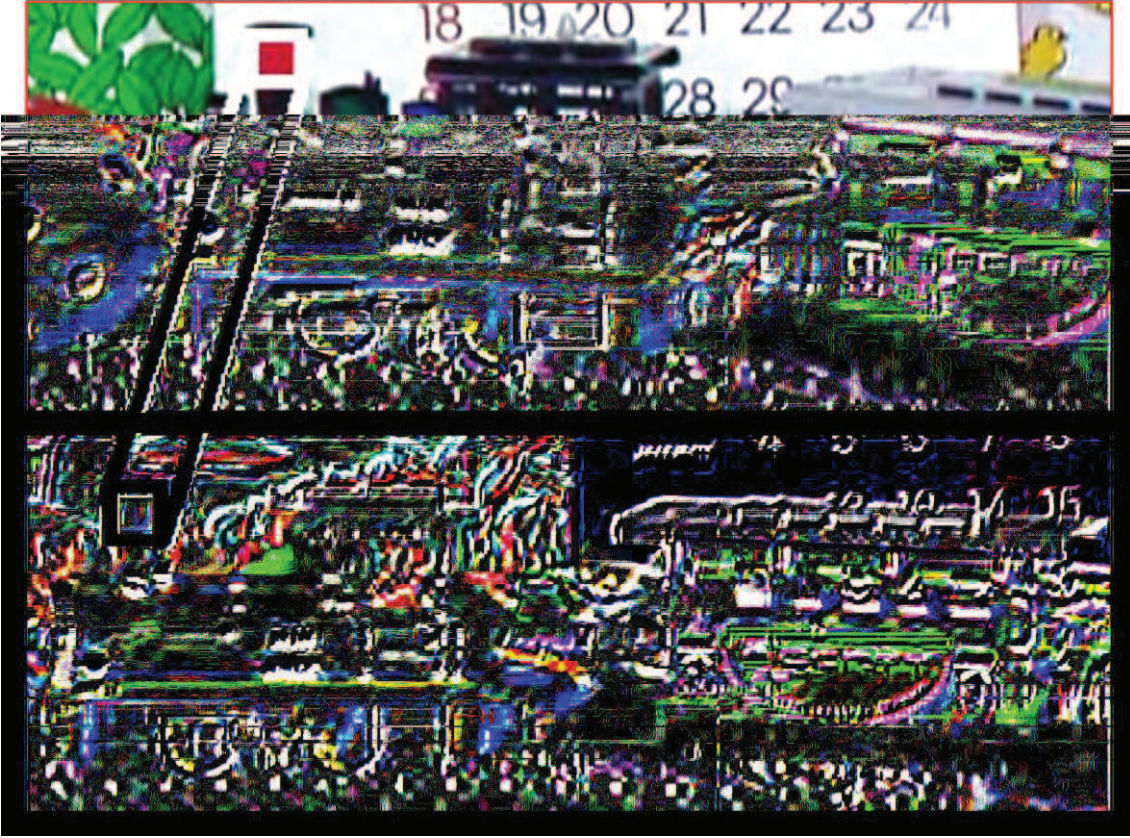
Dekodlayıcı yapısı ise kodlayıcı yapısının tam tersi olmakla birlikte, ihtiyaç duyulmadığı için, dekodlayıcı tarafında mod seçim aşaması bulunmamaktadır.

2.3.1 Video sıkıştırma hareket kestirimi ve dengelemesi

Hareket kestirimi ve dengelemesi, video sıkıştırma yer alan iki önemli konudur. Video sıkıştırma gerçekleştirilen ilerlemelerin büyük bir çoğunluğu video sıkıştırma algoritmalarında kullanılan hareket kestiriminin iyileştirilmesiyle olmaktadır (Ghanbari 2003).

2.3.1.1 Hareket kestirimi

Hareket kestirimi geçmişten günümüze kadar, H.120 hariç, tüm video standartlarında kullanılan bir tekniktir. Hareket kestirimi, bir video dizisinde, iki çerçeve arasında piksellerin yer değiştirme miktarını hesaplamayı ve böylelikle çerçeveler arasında var olan zamansal artıklığı azaltmayı amaçlamaktadır.



Şekil 2.6 Mobile & Calendar video dizisi 100. ve 186. çerçeveler (üst), (alt)

Şekil 2.6’da, “Mobile & Calendar” video dizisinde yer alan iki çerçeve görünmekte ve her iki çerçevede de, trenin bacasının olduğu bölge kırmızı kare içerisinde gösterilmektedir. Kırmızı kareyle gösterilen bölge, 100. ve 186. çerçevelerde de farklı konumlarda yer almaktadır. Konumlardaki bu farklılık, hem trenin hareketinden, hem de kameranın hareketinden kaynaklanmaktadır. Bu video dizisinde, gösterilen çerçeveler arasında, kamera, sola doğru hareket etmekte ve aynı zamanda çok az da olsa uzaklaşmaktadır (zoom out). Sonuçta, piksellerin hareketleri, nesnelerin hareketleriyle

birlikte, kameranın hareketinden de etkilenmiştir. Hareket kestirim algoritmaları, görüntülerdeki pikselleri inceleyerek ve çerçeveler arasında gerçekleşen ötelemeyi belirleyerek toplam hareketi hesap etmektedir.

H.261 gibi video sıkıştırma standartlarında sadece tam piksel hareket kestirimi, daha sonra geliştirilen MPEG-1 ve MPEG-2’de ise daha iyi kestirim yapabilmek için yarım piksel hareket modeli kullanılmaktadır (ITU 1993). Daha hassas hareket çözünürlüğü yakalayabilmek için MPEG-4’te çeyrek piksel hareket kestirimi yapılmaktadır ve bir dönüşüm ile birlikte, çerçevenin geniş bölgelerinin eğilmesi amacıyla, toplam hareket kestirimi de uygulanmaktadır (Lee 1997). H.264’te ise çoklu referans çerçevelerinden hareket öngörümü yapılmaktadır. Ayrıca bir MB’de sadece bir hareketi destekleyen önceki video sıkıştırma standartlarının aksine MB içerisinde çoklu hareket desteği de eklenmiştir (ISO 2003).

2.3.1.1.1 Bloğa dayalı hareket kestirimi

Bloğa dayalı hareket kestirimi, çerçeveler arasında dikdörtgensel piksel bölgelerinin hareketini hesaplamak için MPEG-1, MPEG-2, MPEG-4 ve H.264’te de kullanılan bir tekniktir (Richardson 2003). MPEG-1 ve MPEG-2’de MB’ler, 16 x 16’lık sabit büyüklükteki karesel bloklardır. MPEG-4’te hareket vektörleri, bloklara ek olarak, rastgele şekillenmiş bölgeler için hesaplanabilir. H.264’te ise blok büyüklüğü 16 x 8, 8 x 16, 8 x 8, 8 x 4, 4 x 8, 4 x 4 piksel büyüklüğünde olabilmekte, 16 x 16 piksel büyüklüğü ile sınırlandırılmamaktadır. Bu yolla, bloklar, hareketli bir nesnenin kenarları gibi hareket devamsızlıklarının olduğu bölgelere hizalanabilmektedir.

Şekil 2.6’da, “Mobile&Calendar” dizisindeki 100. ve 186. çerçevelerde, bir blok, kırmızı kare ile gösterilmektedir. Bu bloğa ait piksellerin ötelenmesi, bloğun hareketi olarak belirlenmektedir.

Bir bloğun hareketi, Hareket Vektörleriyle (MV, Motion Vectors) ifade edilmektedir ve denklem 2.1’deki gibi bloktaki piksellerin aydınlanma değerleri arasındaki Mutlak

Farkın Toplamının (SAD – Sum of Absolute Difference) en küçük olduğu yere yatay ve dikey öteleme olarak hesaplanmaktadır.

$$MV = \min \left(\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_n(i, j) - I_{n-1}(i+x, j+y)| \right)_{x=-W..W, y=-W..W} \quad (2.1)$$

Burada, MV , hareket vektörünü, I_n , şimdiki çerçeveyi, I_{n-1} , dizideki bir önceki çerçeveyi, (i, j) , pikselin koordinatlarını, N , blok büyüklüğünü, (x, y) , bloğun yatay ve dikeydeki ötelenmesini ve son olarak W ise blok etrafındaki arama bölgesinin boyutunu ifade etmektedir.

Bir bloğun hareketini hesaplamak için SAD'ın yanı sıra, Mutlak Farkın Ortalaması (MAD – Mean Absolute Difference) ve Karesel Hatanın Ortalaması (MSE – Mean Square Error) gibi karşılaştırma yöntemleri de kullanılmaktadır.

Bir bloğun hareketini bulurken arama bölgesinin boyutunu arttırmak, hatayı azaltırken, işlem yükünü arttırmaktadır. Aramadaki işlem yükünü azaltmak için tam arama (exhaustive search) yapmak yerine, daha hızlı arama yapabilen, Üç Aşamalı Arama (TSS – Three Step Search), Yeni Üç Aşamalı Arama (NTSS – New Three Step Search) (Li 1994), Dört Aşamalı Arama (FSS – Four Step Search) (Po 1996), Basit ve Etkili Arama (SES – Simple and Efficient Search) (Lu 1997), Baklava Arama (DS – Diamond Search) (Zhu 2000) gibi blok karşılaştırma yöntemleri de kullanılmaktadır.

2.3.1.1.2 Nesneye dayalı hareket kestirimi

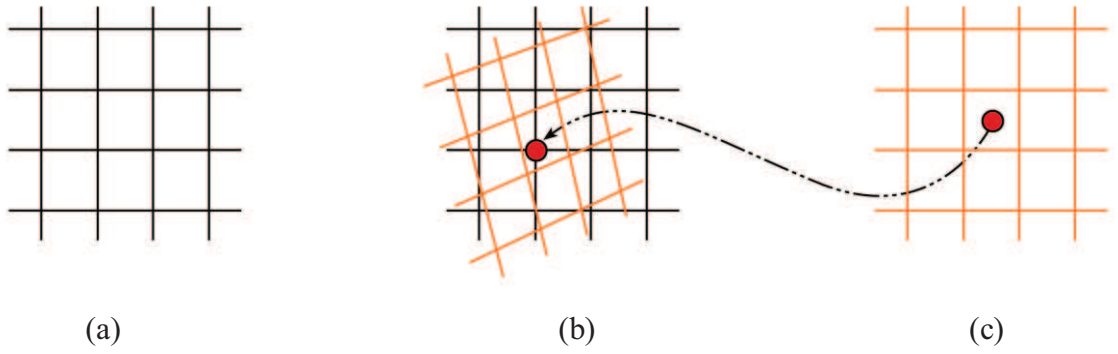
MPEG-4'te kullanılan nesneye dayalı video kodlama, hareketli nesnelerin şekil bilgisine dayandığı için yüksek kodlama veriminin yanı sıra içerik tabanlı işlemlere de olanak sağlamaktadır (Lee 1997).

Nesneye dayalı hareket kestiriminde, video görüntüleri, bloklara ayrılmak yerine hareketli ön plan nesnelere ayrılmaktadır. Her bir nesnenin hareketi birbirinden

bağımsız olarak kestirilir. Bu sebeple, video kodlamada, nesneye dayalı hareket kestirimi en uygun hareket kestirimini gerçekleştirmektedir (Phadtare 2007, Yu 2005).

2.3.1.2 Hareket dengelemesi

Hareket dengelemesi, başka bir çerçeveye hizalanması için, bir çerçevenin koordinat düzlemini değiştiren bir dönüşümdür ve video sıkıştırma, ardışık çerçeveler arasında ilişkiyi arttırmak ve sinyalin entropisini azaltmak için sıklıkla kullanılmaktadır (ISO/IEC-11172-2 1991, ISO/IEC-13818-2 1995, Sikora 1997, ISO/IEC-14406-10 2005). Hareket dengelemesi sayesinde, istenen bir kalitede, bir diziyi kodlamak için gerekli olan bit sayısı düşmektedir.



Şekil 2.7 Hareket dengelemesi

a. Orijinal görüntünün koordinat sistemi. b. ikinci görüntü ile orijinal görüntünün koordinat sisteminin iç içe geçmiş görüntüsü. c. ikinci görüntünün koordinat sistemi

Şekil 2.7'de, bir görüntünün koordinat sisteminin bir başka görüntü üzerine, hareket dengelemesiyle, tekrar örneklenmesi gözükmektedir. Şekil 2.7.a'da, piksellerin yatay ve dikey çizgiler ile belirlendiği referans çerçevesinin koordinat sistemi belirtilmektedir. Şekil 2.7.c'de, birinci çerçeveye göre hareket dengelemesi gerçekleştirilen ikinci çerçevenin koordinat sistemi ve hareket dengelemesi yapılmış çerçevede tekrar örneklenen piksel gösterilmektedir. Şekil 2.7.b'de ise, yatay ve dikey çizgilerle belirtilen referans çerçevesi ve ikinci çerçevenin örtüştüğü noktalar birlikte görünmektedir. Şekil 2.7.b'de ortak nokta çiftleri üstüste gelmektedir.

Hareket dönüşümü uygulanmış çerçevede, bazı yerlerde, boşluklar oluşmaktadır. Bu boşluklar, en yakın komşu (Nearest Neighbour), çift doğrusal (Bilinear), çift kübik (Bicubic) gibi aradeğerleme yöntemleriyle kapatılmaktadır (Turkowski 1990). Bu aradeğerleme yöntemlerinin yanısıra uyarlamalı (adaptive) aradeğerleme algoritmaları da mevcuttur (Su 2004).

2.3.2 Entropi kodlama

Entropi kodlama, video sıkıştırma standartlarında, istatistiksel fazlalığı kaldırmak için AKD katsayılarının nicemlenmesinin ardından kullanılan son aşamadır ve kayıpsız sıkıştırma için kullanılmaktadır.

Entropi kodlayıcı, bir video dizisinin elemanlarını temsil eden semboller dizisini, depolamaya ya da iletme uygun bir bit dizisine sıkıştırmayı amaçlamaktadır (Richardson 2003). Entropi kodlamaya giriş olarak verilen semboller, nicemlenmiş dönüşüm katsayılarını, hareket vektörlerini, video dizisindeki tekrar eşleme (resynchronization) noktasını içeren işaretleyicileri, başlıkları (MB, görüntü, video dizisi başlıkları, vb.) ve ek bilgileri içerebilmektedir.

Huffman ve aritmetik kodlama yaygın olarak kullanılan entropi kodlama çeşitleridir. Bunların sabit ve uyarlamalı (adaptive) modellenmiş uygulamaları da mevcuttur.

H.264'te, entropi kodlamaya iki ayrı alternatif getirilmiştir (Bormans 2004). Bunlardan düşük işlem yükü olanı, CAVLC (Context-based Adaptive Variable Length Coding), yüksek işlem yükü olanı ise CABAC (Context-based Adaptive Binary Arithmetic Coding) olarak bilinmektedir. Her iki yöntem de, Huffman Kodlama (Huffman 1952) gibi daha önceki yöntemlerle kıyasla çok daha üstün sıkıştırma sağlamaktadır.

2.4 Video Kodlama Standartları

Hareketli görüntüleri ve sesleri kodlayabilmek amacıyla, Uluslararası Standartlar Organizasyonu (ISO – International Standards Organisation) tarafından, Hareketli Görüntü Uzmanlar Grubu (MPEG – Moving Picture Experts Group) 1988 yılında kuruldu (Chiariglione 1995). Bu grup, 1993 Ağustos’unda, sesin ve hareketli görüntünün kodlaması yaklaşık 1.5 Mb/s olan ve MPEG-1 olarak anılan standardı yayınladı (Gall 1992, Schafer 1995, ISO/IEC 11172 1993). MPEG, daha sonraları MPEG-2 olarak isimlendirilen standardın çalışmalarına ise, MPEG-1 tamamlanmadan, 1990 yılında başladı (Schafer 1995). MPEG-2’nin ilk sürümü 1994 yılında bitirildi.

MPEG-1, daha çok kayıt (CD-ROM vb) uygulamalarında kullanılan bir standartken, MPEG-2 ise, bit oranı 2 Mb/s ile 30 Mb/s arasında değişen kayıt (DVD vb) ve yayın (HDTV vb) uygulamaları için, daha kaliteli görüntü sağlayan bir standarttır (ISO/IEC 13818 1995). 1999’da nesneye dayalı video kodlama olanağı sağlayan MPEG-4 Visual (MPEG-4 Part 2, MPEG-4 Verification Model), yine MPEG tarafından yayınlandı. Daha sonra, 2001 yılında, MPEG-4 Visual’ın ikinci sürümü yayınlandı (ISO/IEC 14496-2 2004).

MPEG’in yanı sıra, ITU-T tarafından desteklenen Video Kodlama Uzmanlar Grubu (VCEG – Video Coding Experts Group) da video kodlama standartları üzerine çalışan bir gruptur. ITU-T, 1992 önce CCITT olarak anılmaktaydı (ITU 2006). VCEG, H.261, H.263 gibi video standartlarını geliştirdi. H.261 standardı, daha çok videokonferans (videoconferencing) uygulamaları için geliştirildi (ITU-T 1993). H.261’in ardından kodlama verimliliği daha iyi olan H.263 yayınlandı (ITU-T 1998). Bu standartlar, H.264’ün gelişmesine de önderlik etti.

VCEG ve MPEG’in ortaklaşa çalışması sonucu MPEG-4 Part 10 olarak da bilinen “Advance Video Coding” (AVC) 2003 yılında yayınlandı (ISO/IEC 14496-10 2003).

2.4.1 MPEG-4 standartları

ITU-T VCEG'nin, H.263 standardını geliştirmeye başladığı 1993 yılında, ISO MPEG ise MPEG-4 Part 2'yi geliştirmeye başladı. MPEG-4 Part 2, genelde MPEG-4 olarak adlandırılmaktadır. MPEG 4, temelde H.263'e benzemesine rağmen, H.263'ten daha fazla özelliğe sahiptir (Koenen 2002). DivX, XviD, QuickTime, WMV gibi günümüzde en çok kullanılan video sıkıştırma formatları MPEG-4 tabanlıdır.

1997 yılında ITU-T VCEG, H.264'ün eski sürümü olan H.26L'yi geliştirmeye başladı ve 1999 yılında ilk test sürümü tamamlandı. 2001 yılında ISO MPEG ile ITU-T VCEG gruplarından uzmanların bir araya gelmesiyle JVT (Joint Video Team) kuruldu ve ortak bir standart geliştirilmeye başlandı. Bu standart, ITU-T tarafından H.264 olarak adlandırılırken, ISO tarafından ise MPEG-4 Part 10 olarak adlandırılmaktadır. MPEG, daha önce geliştirmiş olduğu ve MPEG-4'te de kullanılan AAC kayıplı ses sıkıştırma standardının video karşılığı gibi düşünerek, sonradan AVC (Advanced Video Coding) ismini kullanmaya başladı. MPEG-4 Part 2, AVC'nin geliştirilmesinden sonra ASP (Advanced Simple Profile) olarak adlandırılmaktadır.

MPEG-4'te nesne tabanlı hareket kestirimi desteklediği ve tezde de nesneye dayalı hareket kestirimi ve dengelemesi yapıldığı için bir sonraki bölümde MPEG-4 kodlama standartından bahsedilmektedir.

2.4.1.1 MPEG-4

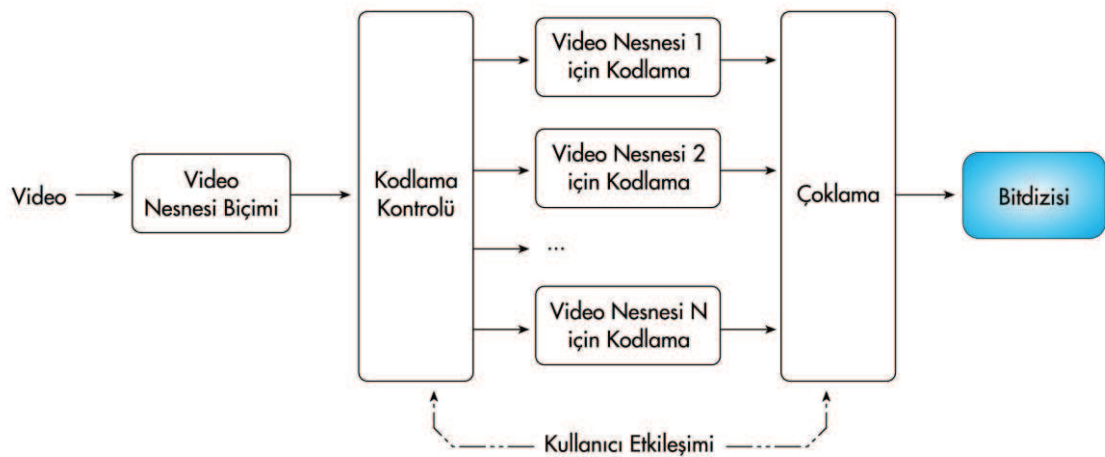
MPEG-1 ve MPEG-2'dekine benzer şekilde hareket dengelemesi ve AKD tabanlı sıkıştırma yapabilmesinin yanı sıra MPEG-4, çerçeveyi, çerçevede yer alan hareketli nesnelere ayıran nesne tabanlı kodlamaya izin vermektedir (Lee 1997). Şekil 2.8'de de görüldüğü gibi nesne tabanlı kodlamada temel olarak, bir sahne içinde yer alan hareketli ön plan nesnelere, arka plandan ayrı birer video nesnesi (VO - video object) olarak düşünülmektedir (Richardson 2003). Sıkıştırma etkinliğini arttırmak için gerekirse

video nesnelerinin her biri için farklı nesne tabanlı tahmin araçları kullanılıp, farklı niceleme ve kodlama yapılabilmektedir (Sikora 1997).



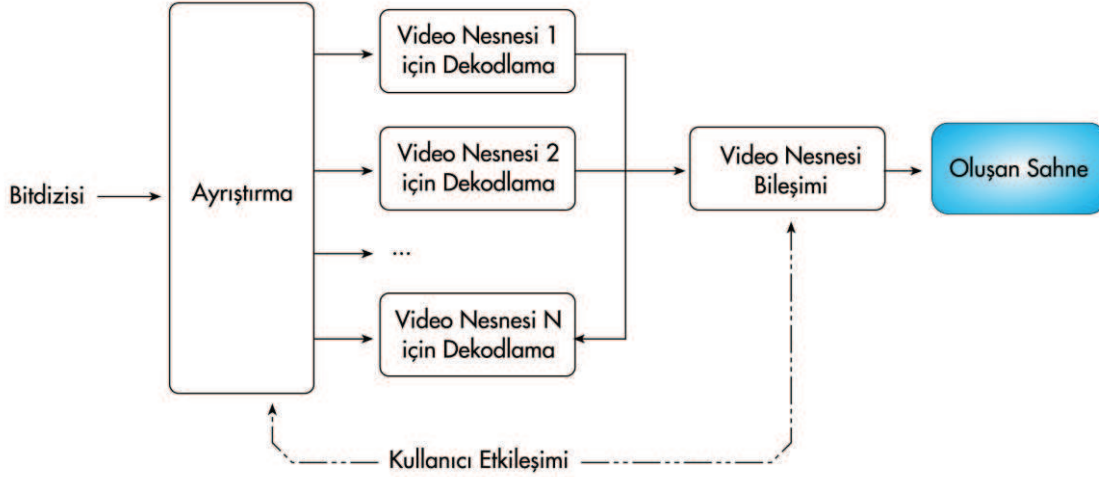
Şekil 2.8 Üç tane video nesnesi içeren video sahnesi

Şekil 2.9, MPEG-4 video kodlayıcının genel yapısının göstermektedir (Nunes 1997). Bu yapının en temel özelliği rastgele şekillenmiş video nesnelerinin birbirinden bağımsız olarak kodlanabilmesidir. Video nesnelerinin karakteristiğine uygun farklı araçlar ve algoritmalar kodlama sırasında kullanılabilir. Bunun da ötesinde, kullanıcıya da her bir video nesnesi için farklı kodlama parametresini seçebilme imkanı sağlanmaktadır. Kodlayıcı, ayrı bir dizide, dekodlayıcıya orijinal sahneyi tekrar oluşturmasına olanak sağlayan bileşim senaryosunu da, video nesnelerinin yanı sıra, göndermektedir.



Şekil 2.9 MPEG-4 video kodlayıcı yapısı

MPEG-4 video dekodlayıcının genel yapısı ise Şekil 2.10'da gösterilmektedir (Nunes 1997). Burada her bir video nesnesi ayrı ayrı dekodlanmakta ve video sahnesi, gönderilen bileşim senaryosuna göre birleştirilmektedir.



Şekil 2.10 MPEG-4 video dekodlayıcı yapısı

2.4.1.2 MPEG-4'ün H.264 ile karşılaştırılması

MPEG-4, geniş işlevselliğe sahip bir video kodlama standardıdır. MPEG-4, nesne tabanlı video kodlamayı da desteklediği için, video düzenleme (video editing), video işleme (video manipulation) gibi uygulamalar için diğer video kodlama standartlarına göre daha uygundur (Richardson 2003).

H.264 ise MPEG-4'ten daha dar bir amaca sahiptir ve öncelikli olarak, dikdörtgenel video çerçevelerinin sağlıklı bir şekilde taşınmasını ve verimli kodlanmasını desteklemek için tasarlanmıştır. Bu amaçla, diğer video kodlama standartlarına göre daha fazla sıkıştırma oranına sahiptir (Richardson 2003).

MPEG-4 ve H.264'ün yanı sıra MPEG-1'den H.264'e kadar MPEG tarafından geliştirilmiş tüm video kodlama standartlarının özellikleri Çizelge 2.2'de özetlenmektedir.

Çizelge 2.2 MPEG tarafından geliştirilmiş video kodlama standartlarının özellikleri

Özellikler	MPEG-1	MPEG-2	MPEG-4	H.264
MB Büyüklüğü	16x16	16x16, 16x8	16x16	16x16
Blok Büyüklüğü	8x8	8x8	16x16, 16x8, 8x8	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
Uzaysal Dönüşüm	DCT 8x8	DCT 8x8	DCT 8x8	Int DCT 4x4, 8x8
Nicemeleme	Sayı	Sayı	Vektör	Sayı
Entropi Kodlama	Huffman	Huffman	Huffman	Exp-Golomb, CABAC, CAVLC
Arama Keskinliği	Tam ve Yarım Piksel	Yarım Piksel	Tam, Yarım ve Çeyrek Piksel	Tam, Yarım ve Çeyrek Piksel
Profil Sayısı	0	6	8	4
Çerçeve Tipleri	I, P, B	I, P, B	I, P, B	I, P, B, SI, SP
Veri Oranı	<1.5 Mbit/s	2-15 Mbit/s	64 Kbit/s - 4 Mbit/s	64 Kbit/s - 240 Mbit/s
Kayıpsız Sıkıştırma	Yok	Yok	Yok	Var

3. HAREKETLİ NESNE TESPİTİ

3.1 Giriş

MPEG-4 video kodlama standardı, video dizilerini, Video Nesne Düzlemlerine (VOP-Video Object Planes) ayırarak, içerik tabanlı işlevselliği desteklemektedir. MPEG-4'te, arka plan modelini temsil eden VOP dışında, her bir VOP bir tane hareketli nesneyi temsil etmektedir (Meier 1998).

Nesne tabanlı hareket kestiriminin başarımı, hareketli nesnelere arka plandan düzgün biçimde ayrılmasıyla fazlaca ilintilidir. Hareketli nesne çıkarımı, nesne tabanlı hareket kestiriminin ilk basamağını oluşturmaktadır ve sonucu doğrudan etkilemektedir.

Hareketli nesne tespiti için kullanılan birçok yöntem mevcuttur. Bu yöntemler genelde çerçeve farkı ve arka plan çıkarımı olarak iki sınıfa ayrılmaktadır. Hareketli nesnelere elde edilmesinin en kolay yolu mevcut çerçeve ile bir önceki çerçevenin birbirinden çıkarılmasıdır. Hareketli nesnelere tutarlı ve düzgün bulunması ise ancak arka planın modellenmesi ve bu model kullanılarak hareketli nesnenin bulunmasıyla gerçekleşmektedir.

3.2 Çerçeve Farkı

Çerçeve farkı yöntemi, hareketli nesnelere tespit etmek için sabit kameradan alınan videolarda uygulanan en basit yöntemdir. Çerçeve farkı yönteminde, denklem 3.1'de belirtildiği gibi ardışık çerçeveler birbirinden çıkartılarak ve bir eşik değeri uygulanarak hareketli nesnelere elde edilmektedir.

$$Fg(x, y) = \begin{cases} 1, & |I_k(x, y) - I_{k-1}(x, y)| > T \\ 0, & |I_k(x, y) - I_{k-1}(x, y)| \leq T \end{cases} \quad (3.1)$$

Burada, $I_k(x, y)$, mevcut çerçeveyi, $I_{k-1}(x, y)$, bir önceki çerçeveyi, $Fg(x, y)$ mevcut çerçevenin hareketli bölgelerinin görüntüsünü ve T ise eşik değeri ifade etmektedir.

$Fg(x, y)$ 'de ön plan pikselleri beyaz (1) ile arka plan pikselleri ise siyah (0) ile temsil edilmektedir.

Çerçeve farkı yöntemi, arka plan çıkarımıyla hareketli nesnelerin tespit edilmesi yöntemine göre daha basit ve daha hızlıdır. Fakat düşük başarımı vardır. Sadece iki ardışık çerçevedeki farka bakıldığı için parlaklık değişimi yöntemin başarımını çok fazla etkilememektedir. Bu yöntemde karşılaşılan en büyük sorun ise eşik değerin belirlenmesidir. Eşik değerin belirlenmesi için Isodata (Ridler 1978), Triangle ve Otsu yöntemi gibi birçok yöntem bulunmaktadır (Jain 1995).

3.3 Arka Plan Çıkarımı

Arka plan çıkarımı yöntemlerinin temel amacı referans çerçevesi oluşturmaktır. Bu referans çerçevesi, genelde “arka plan modeli” ya da “arka plan görüntüsü” olarak adlandırılmaktadır.

Sabit kameradan alınmış video dizilerinde ön plan nesnelerini belirlemek için arka plan ayrımı sıklıkla kullanılan bir yöntemdir (McIvor 2000). Sabit kameradan alınmış bir videoda, arka plan modeli elde edilmişse, modeldeki her hangi bir değişiklik hareketli nesne olarak tanımlanmaktadır. Bu değişimler, mevcut çerçeve ile arka plan modelinin birbirinden çıkarılmasıyla hesaplanabilmektedir. Arka plan modelindeki ufak değişimleri ön plan olarak algılamamak için fark görüntüsü uygun bir eşik değer ile karşılaştırılarak, denklem 3.2'de de belirtildiği gibi ikili görüntüye (binary image) çevrilmektedir.

$$Fg(x, y) = \begin{cases} 1, & |I(x, y) - Bg(x, y)| > T \\ 0, & |I(x, y) - Bg(x, y)| \leq T \end{cases} \quad (3.2)$$

Burada, $I(x, y)$, orijinal görüntüyü, $Bg(x, y)$ ise, arka plan modelinin görüntüsünü ifade etmektedir. Hareketli nesnelerin tespiti için arka plan çıkarımı yöntemi, çerçeve farkı yönteminden daha karmaşıktır fakat daha başarılı sonuçlar vermektedir.

Arka plan çıkarımını gerçekleştirmek için birçok yöntem kullanılmaktadır (Koller 1994, Piccardi 2004, Friedman 1997, Panahi 2008). Bu yöntemlerin hepsi, video dizisindeki görüntülerden arka plan modelini kestirmeye çalışmaktadır (Piccardi 2004).

3.3.1 Arka plan çıkarımı algoritmaları

Arka plan çıkarımında genel olarak aydınlanma değişimleri, gölge, kameranın sallanması ve arka planda gerçekleşen farklı değişiklikler gibi zorluklarla uğraşmaktadır. u zorlukların üstesinden gelebilmek için bir çok yöntem geliştirilmiştir. Ancak hala ön plan arka plan ayrımı bilgisayarla görü alanında üzerinde uğraşılan bir konudur (Benezeth 2008).

(McFarlane 1995)'te sunulan algoritmada ortancayı (median) bulabilmek için yinelemeli bir süzgeç (filtre) kullanılmaktadır. Bu yöntemin öne çıkan avantajları, hesaplama verimliliği, gürültüye karşı sağlamlığı ve basitliğidir.

(Grimson 1999)'da önerilen yöntemde, uyarlamalı bir arka plan modellemesi sunulmaktadır. Burada, her bir piksel Gauss karışımı olarak modellenmektedir ve modelin güncellenmesi için çevrimiçi bir yaklaşım kullanılmaktadır.

(Elgammal 2000)'de arka plan için parametrik olmayan bir model önerilmektedir. Kısa süreli ve uzun süreli arka plan modelleri oluşturabilmek için parlaklık değerlerinin bir örneği kullanılmaktadır. Bu şekilde kısa ve uzun süreli iki arka plan modelinin kullanılmasıyla arka plandaki değişikliklere hızlı bir şekilde uyum sağlanmaktadır.

(Mittal 2004)'de parlaklık değerleri hareket bilgileriyle birleştirilmektedir. Olasılık Yoğunluk Hesabı (Probability Density Estimation) için uyarlamalı bir çekirdek kullanılmaktadırlar.

(Norriega 2006)'da sunulan algoritmada, uzaysal bilgileri dikkate alan yerel çekirdek histogramları tanıtılmaktadır. Bu yöntem, mevcut ve referans çerçeveler üst üste kesişen

pencerelere ayırmaktadır ve her pencere için renk histogramı hesaplamaktadır. Daha sonra, iki ortak pencerenin histogramının karşılaştırılmasıyla ön plan tespit edilmektedir. Aydınlanma değişimlerine karşı zayıf bir yöntemdir. Bu yöntemde kullanılan renk histogramları yerine (Norriega 2006)'da gradyan histogramları kullanılarak aydınlanma değişimlerine karşı sağlamlık kazandırılmıştır.

(Sun 2006)'da önerilen yöntemde, ön planı elde etmek için renk ve karşıtlık bilgileriyle temel arka plan çıkarımı birleştirilmektedir. Enerji fonksiyonunu biçimlendirmek için renk ve karşıtlık bilgileri kullanılmaktadır. Bu yöntemde, sistemin sağlamlığını arttırmak için tüm arka plan renk modellerinin ve yarım piksel arka plan renk modellerinin uyarlamalı karışımı kullanılmaktadır.

Geliştirilen yöntemlerin her birisi bazı uygulamalar için daha uygundur. Bu sebeple uygulamanın özelliğine uygun arka plan çıkarımı yönteminin seçilmesi daha doğru olmaktadır.

3.4 Görüntü İyileştirme

Çerçeve farkı ve arka plan çıkarımı yöntemleri ile elde edilen nesne silueti (ikili görüntü), nesnenin gerçek siluetini tam olarak karşılayamamaktadır. Kamera gürültüsü ya da görüntüdeki aydınlanma değişimlerinin sebep olduğu gürültüler, eşik değer uygulanmasına rağmen, elde edilen nesne siluetinde ortaya çıkabilmektedir. Ayrıca arka plan ile nesne dokusu arasındaki benzerlikler de nesne silueti üzerinde boşluklara sebep olmaktadır.

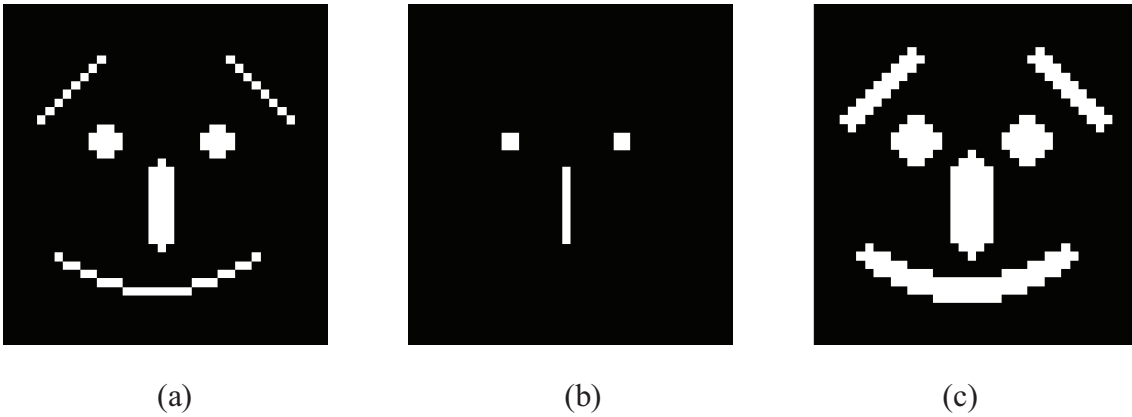
Bu istenmeyen bölgeleri ortadan kaldırmak için nesne silueti üzerinde morfolojik işlemler uygulanmaktadır. Aşınma (erosion) ve genişleme (dilation) temel olarak kullanılan iki morfolojik işlemdir (Bovik 1999). Aşınma nesnenin kenarlarını yok etmektedir, diğer taraftan genişleme ise nesnenin kenarlarına pikseller eklemektedir. Eklenecek ya da çıkartılacak piksel sayısı kullanılan yapısal elemanın (structuring

element) büyüklüğüne ve şekline bağlıdır. Şekil 3.1’de elde edilen sonuçlar için kullanılan yapısal eleman denklem 3.3’te belirtilmektedir.

$$SE = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.3)$$

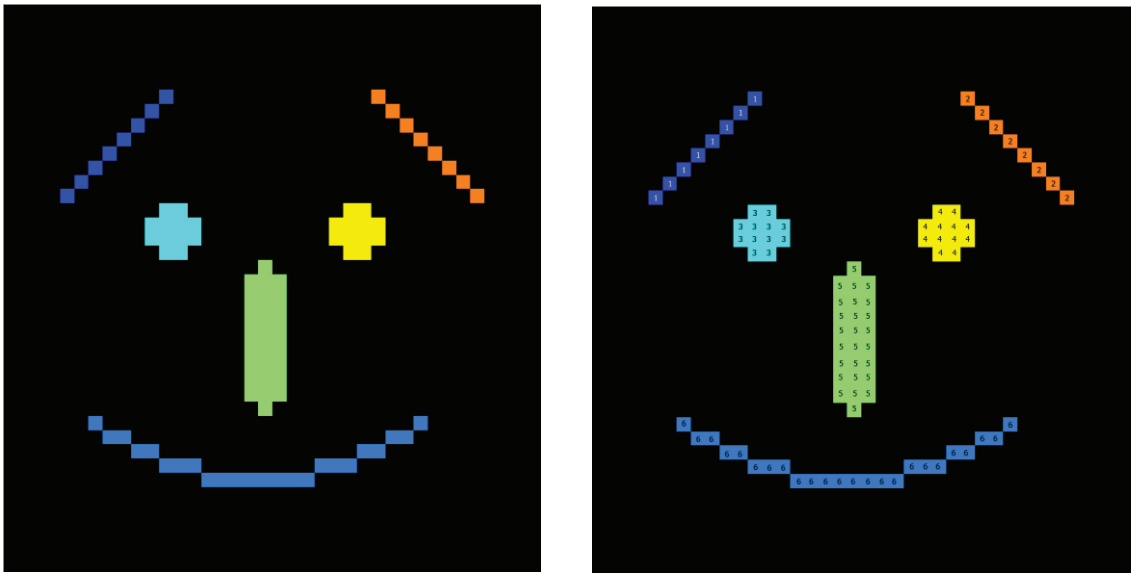
Aşınma işleminin temel kullanılma amacı çıkartılan ön plan nesnesinin silueti üzerinde oluşan küçük gürültüleri yok etmektir. Aşınma, nesne silueti üzerindeki her bir piksele uygulanmaktadır. İşlem sonucunda piksel, komşuları arasından değeri en küçük olanın değerini almaktadır. İkili görüntüde komşularından her hangi birisi 0 değerindeyse piksel de aşınma sonucunda 0 değerini almaktadır. Aşınma kullanılarak ön plan nesnesinden, ön plan nesnesi olarak görünen ve beyaz piksellerle tamamen çevrelenmeyen pikseller çıkartılmaktadır.

Genişleme işlemi, genelde aşınma işlemi sonucu yok edilen ön plan nesnelerinin kenarlarını tekrar oluşturmak için aşınma işleminden sonra kullanılmaktadır. Genişleme de aşınma gibi nesne silueti üzerindeki her bir piksele uygulanmaktadır. Genişleme sonucunda piksel, komşuları arasından değeri en büyük olanın değerini almaktadır. Aşınmanın tersine ikili görüntüde komşularından her hangi birisi 1 değerindeyse piksel de genişleme sonucunda 1 değerini almaktadır.



Şekil 3.1 Aşınma ve genişleme
a. Orijinal ikili görüntü. b. Aşınma ve c. genişleme işlemi sonucu oluşan görüntüler

Her bir pikselin diğer piksellere bağlı olduğu bir piksel grubuna, bağlı bileşen (connected component) denilmektedir (Jain 1995). Eğer görüntüde birden fazla ön plan nesnesi var ise bu nesnelere birbirinden ayırabilmek için Bağlı Bileşen Analizi (Connected Component Analysis) (Haralick 1992) kullanılmaktadır. Şekil 3.2’de, Şekil 3.1.a’daki orijinal ikili görüntü üzerine bağlı bileşen analizi yapılması sonucunda elde edilen bölgeler ve bölgelerin etiketleri görülmektedir. Şekil 3.2.b’deki görüntüde arka plan 0 olarak etiketlenmiştir. Ön plan nesnelere ise 1’den 6’ya kadar etiketlenmiştir.



(a)

(b)

Şekil 3.2 Bağlı bileşen analizi
a. Bölgelerin ayrılması b. bölgelerin etiketleri

Çıkartılan ön plan nesnesi üzerinde var olan gölgeleri yok etmek için de bir çok yöntem mevcuttur (Prati 2003, Salvador 2001). Fakat hareket kestirimi uygulamalarında, gölge de hareket ettiği için onu da ön plan nesnesi olarak belirlemek daha iyi sonuç vereceğinden gölge yok etme yöntemleri bu tezde uygulanmamaktadır.

4. NESNE TAKİBİ

4.1 Giriş

Nesne takibi, bilgisayarla görü alanında önemli bir konudur. Günümüzde, güçlü bilgisayarların ve yüksek kalitede görüntü verebilen ucuz video kameraların var oluşu ve ayrıca otomatik video analiz yöntemlerine duyulan ihtiyacın artışı, nesne takibi algoritmalarını ilgi odağı haline getirmektedir. Bir video analiz yöntemi temel olarak üç temel aşamadan oluşmaktadır. İlk aşamada videodaki ilgilenilen hareketli nesne/nesneler tespit edilmekte, sonraki aşamada bu nesne akan görüntüde çerçeveler arasında takip edilmekte, en son olarak da nesne takibinin analizi sonucu nesnenin davranışı incelenmektedir.

Nesne takibi algoritmaları; harekete dayalı teşhis, otomatik gözetleme, video veritabanları için video indeksleme, insan-bilgisayar etkileşimi, trafik izleme ve araç konum belirleme gibi bir çok uygulamada kullanılmaktadır.

Nesne takibinin temel amacı, ardışık çerçeveler üzerinde istenilen nesnenin, yerini belirleyerek, hareket doğrultusunu çıkarmaktır. Ayrıca, nesne takibi algoritmaları nesnenin her bir anda videoda kapladığı tüm alanı ortaya çıkartmak için de kullanılmaktadır.

Nesne takibi işlemine başlamadan önce hedef nesneye, ortama ve uygulamaya göre belirlenmesi gereken bazı kıstaslar vardır. Bunlar temel olarak, nesne temsili, takip amacıyla kullanılan öznitelikler ve kullanılacak hareket modelidir.

4.2 Nesne Temsili

Nesne takibi aşamasında önemli rol oynayan seçimlerden biri takip edilecek nesnenin nasıl temsil edileceğidir. Uygulamanın amacına göre takip edilecek nesne değişiklik göstermekte, takip edilecek nesneye göre de nesnenin nasıl temsil edileceği

belirlenmektedir. Nesneyi temsil etmek için, nesnenin şekil ve hareket yapısına uygun olan temsil biçimi seçilmektedir. Şekil 4.1’de nesne şekillerinin farklı temsil çeşitleri görülmektedir (Yılmaz 2006).

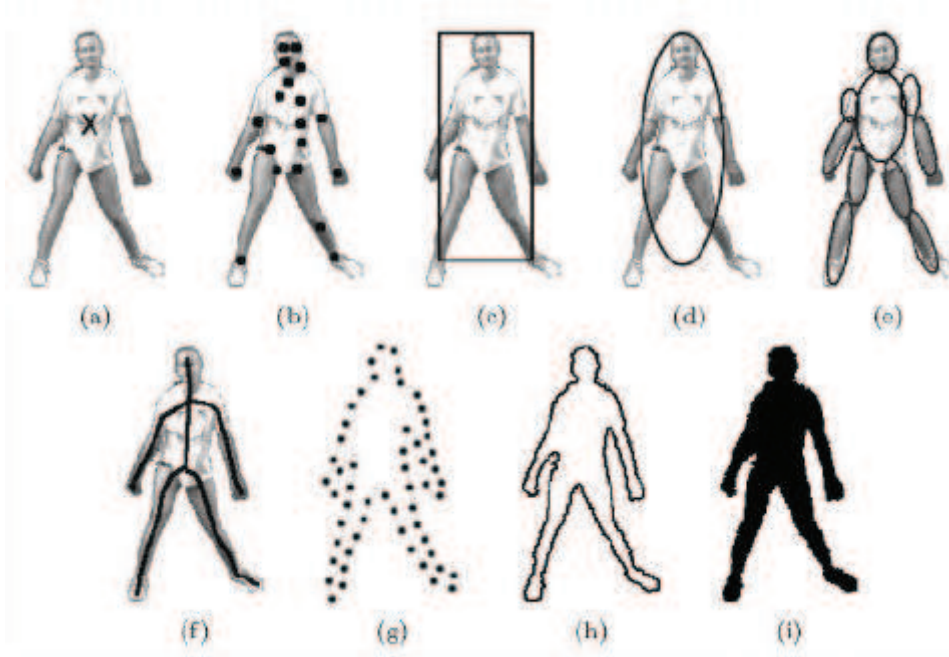
Noktalar ile temsil: Nesne, merkezinde yer alan bir nokta ile (Şekil 4.1.a) veya noktalar kümesi ile gösterilmektedir (Şekil 4.1.b). Nokta temsili görüntüde az yer kaplayan nesnelere takip etmek için uygundur.

Geometrik şekil ile temsil: Nesne dikdörtgen, elips gibi bir geometrik şekil ile gösterilmektedir (Şekil 4.1.c ve d). Bu şekillerle gösterilen nesnenin hareketi genellikle ötelemeli, doğrusal veya izdüşümsel olmaktadır. Geometrik şekil ile temsil genellikle katı, esnek olmayan nesnelere için uygun olmakla birlikte, esnek nesnelere takip etmek için de kullanılmaktadır.

Nesne silüeti ve çevre çizgisi ile temsil: Çevre çizgisi nesnenin sınır çizgilerini temsil etmektedir (Şekil 4.1.g ve h). Çevre çizgisinin içinde kalan bölge ise nesnenin silüeti olarak tanımlanmaktadır (Şekil 4.1.i). Çevre çizgisi ve silüet temsillerinin karmaşık ve esnek yapıda nesnelere takibinde kullanımı uygundur (Ekinci 2005, Yılmaz 2004).

Eklemlili şekil modelleri ile temsil: Eklemlili nesnelere birbirine eklem ile bağlanan parçalardan oluşmaktadır. Örneğin insan; gövdesi, kolları, bacakları, kafası, elleri ve ayaklarının eklemlilerle birbirine bağlanmasından oluşan bir canlıdır. Bu temsilde parçalar arası ilişkiler kinematik hareket denklemleri ile çıkartılabilmektedir. Eklemlili bir nesnenin temsiliinde, nesnenin eklemliler ile bağlanan her bir parçası silindir veya elips ile ifade edilebilmektedir (Şekil 4.1.e).

İskelet model ile temsil: Nesnenin iskeleti, nesnenin silüetine orta eksen dönüşümü uygulanarak çıkarılabilmektedir. Bu model genellikle nesne tanımada şekil gösterimi olarak kullanılmaktadır. İskelet temsili, hem eklemlili hem de esnek yapılı nesnelere temsili için kullanılabilmektedir (Şekil 4.1.f).



Şekil 4.1 Nesne temsilleri

a. Merkez nokta, b. çoklu nokta, c. dikdörtgen, d. elips, e. parça tabanlı çoklu elips, f. iskelet, g. çevre çizgisinde kontrol noktaları, h. çevre çizgisi, i. nesne silueti

Tezde, arka plandan ayrılan ön plan nesnelere noktalar ile temsil edilmektedir. Bu noktalar ise ön plan nesnesinin öznitelikleridir. Bu sebeple sonraki bölümlerde, öznitelik çıkarımına dayalı nesne takibinden ve öznitelik çıkarma yöntemlerinden bahsedilmektedir.

4.3 Öznitelik Çıkarımına Dayalı Nesne Takibi

Arka arkaya gelen görüntülerde, bulunan nesnelere öznitelikler ile temsil edilmektedir. Bu yaklaşım her bir görüntüde hedef nesneyi bulacak harici bir nesne bulma algoritmasına ihtiyaç duymaktadır. Nokta eşleştirme; özellikle kaybolma, yanlış bulma, nesnelere görüntüye giriş ve çıkışlarının var olduğu durumlarda karmaşık bir problemdir.

4.4 Nesne Takibi için Öznitelik Seçimi

Nesne takibinde öznitelik seçimi kritik bir rol oynamaktadır. Seçilecek özneliğin veya özneliklerin takip edilecek nesneyi ayırt edici kılması gerekmektedir. Görsel özneliklerin kullanılmasındaki en önemli sebep de her bir nesnenin kendine özel görsel özneliklerinin olmasıdır. Öznitelik seçimi, nesne temsili ile yakından ilgilidir. Örneğin histogram tabanlı temsillerde nesnenin renk özneliği kullanılırken, çevre çizgisi tabanlı temsillerde nesnenin kenar özneliği tercih edilmektedir. Çoğu nesne takibi algoritmasında bu tarz kombinasyonlar kullanılmaktadır (Yılmaz 2006). Nesne takibinde genellikle renk, kenar, optik akış ve doku gibi öznelikler tercih edilmektedir.

Renk: Bir nesnenin rengi, ışık kaynağının ışık tayfının güç dağılımı ve nesnenin yüzey yansıma özellikleri olmak üzere iki esas fiziksel faktörden etkilenmektedir. Görüntü işleme çalışmalarında, rengi temsil etme amacıyla genellikle RGB (Red, Green, Blue – Kırmızı, Yeşil, Mavi) renk uzayı kullanılmaktadır. Bunun yanında YUV ve HSV (Hue, Saturation, Value – Renk Tonu, Doygunluk, Değer) uzayları da kullanılmaktadır. Nesne takibinde hangi renk uzayının kullanılmasının daha verimli olduğuna dair kesin bir karar yoktur.

Kenar: Nesne dış çizgileri görüntü yoğunluğunda genellikle büyük değişimlere yol açmaktadır. Kenarların en önemli özelliği renk öznelikleri ile kıyaslandığında ışık değişimlerine daha az duyarlı olmasıdır. Nesnelerin dış çizgisini takip eden algoritmalar öznelik olarak genellikle kenar özneliğini kullanmaktadır. Sonuçlarının basitliği ve kesinliği sebebiyle en çok tercih edilen kenar bulma yaklaşımı Canny Kenar Bulma (Canny Edge Detector) (Canny, 1986) yöntemidir.

Optik Akış: Optik akış, bir bölgedeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Arka arkaya gelen görüntüdeki eş piksellerin parlaklıkları kullanılarak hesaplanmaktadır. Optik akış, hareket tabanlı bölütleme ve takip uygulamalarında sıklıkla kullanılan bir özneliktir. Yoğunluk optik akışı hesaplamak için kullanılan popüler yöntemlerden bazıları: (Horn 1981, Lucas, 1981, Black 1996, Szeliski 1997)'dir.

Doku: Doku, pürüzsüzlük ve düzenlilik gibi öznitelikleri ölçen yüzeyin yoğunluk değişim ölçüsüdür. Renk ile karşılaştırıldığında doku tanımlayıcılarını oluşturabilmek için filtre uygulama gibi ön işlemlere ihtiyaç duymaktadır. Kenar öznitelikleri gibi doku öznitelikleri de ışık değişimlerine daha az duyarlıdır.

Günümüz görüntü eşleştirme algoritmaları genelde algılayıcı (detector) ve açıklayıcı (descriptor) aşamalarından oluşmaktadır. Bu algoritmalarda ilk olarak, karşılaştırılan görüntülerde ilgi noktaları (interest points) algılanmakta ve ilgi noktalarının etrafında bir bölge seçilmektedir, daha sonra her bir bölge değişimsiz (invariant) bir açıklayıcıyla ya da öznelikle ilişkilendirilmektedir. Açıklayıcıların eşleştirilmesiyle de ortak nokta çiftleri elde edilmektedir. Algılayıcı ve açıklayıcı ne kadar değişimsizse algoritma da o kadar başarılı kabul edilmektedir.

Görüntü eşleştirme algoritmaları, artan değişimsizlik özelliklerine göre sınıflandırılabilir. Bu algoritmaların tümü öteleme (translation) hareketine karşı değişimsizdirler. Harris nokta algılayıcı, öteleme hareketinin yanı sıra dönme (rotation) hareketine karşı da değişimsizdir (Harris 1988). Harris-Laplace, Hessian-Laplace ve difference-of-Gaussian (DoG) bölge algılayıcıları ise dönme ve ölçekleme (scaling) değişimlerine karşı değişimsizdir (Mikolajczyk 2001, Mikolajczyk 2004, Lowe 2004, F'evrier 2007). Harris-Affine (Mikolajczyk 2002) Hessian-Affine (Mikolajczyk 2004) bölge algılayıcıları, kenar tabanlı bir bölge algılayıcı (Tuytelaars 1999), parlaklık tabanlı bir bölge algılayıcı (Tuytelaars 2000), dağınım tabanlı bir bölge algılayıcı (Kadir 2004), MSER (Maximally Stable Extremal Region) (Matas 2004), LLD (Level Line Descriptor) (Muse 2006) ise ilgin (affine) dönüşümlere karşı da değişimsizdir. Bu ilgin dönüşümlere karşı değişimsiz olan algılayıcılar arasında MSER, diğerlerine göre daha başarılı sonuç vermekte ve Hessian-Affine ve Harris-Affine ise onun ardından gelmektedir (Mikolajczyk 2005b).

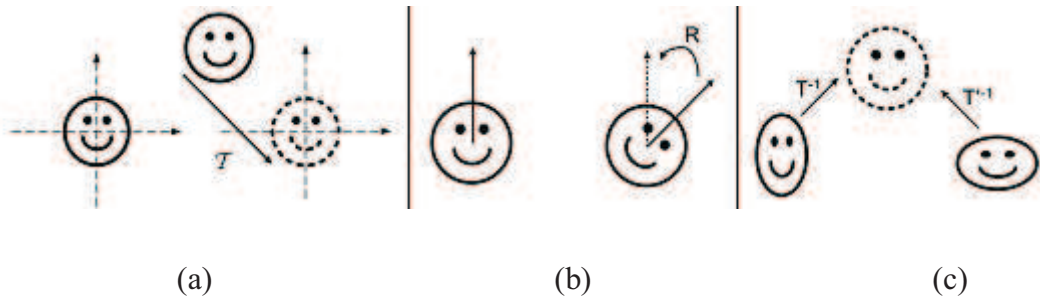
Görüntü eşleştirme algoritmaları arasında dönüm noktası olarak kabul edilen Ölçek Değişimsiz Öznitelik Dönüşümü (SIFT, Scale-Invariant Feature Transform) ölçekleme ve dönmeye karşı tamamen değişimsiz aydınlanma ve bakış açısı değişimlerine ise kısmi değişimsizdir. SIFT yönteminde, öteleme, dönme ve ölçekleme değişimlerine

karşı deęişimsiz olan DoG bölge algılayıcısı ile aydınlanma ve bakış açısı deęişimlerine kısmı deęişimsiz olan, bölgedeki gradyan yönelim dağılımları tabanlı bir açıklayıcı birlikte kullanılmaktadır (Lowe 2004). SIFT'teki bu iki aşama, SIFT algılayıcı ve SIFT açıklayıcı olarak adlandırılmaktadır. SIFT algılayıcı, ilgin dönüşümlere, Hessian-Affine ve Harris-Affine algılayıcılardan daha az deęişimsizdir (Mikolajczyk 2004). Fakat SIFT algılayıcı, SIFT açıklayıcıyla birlikte kullanıldığında, ilgin dönüşümlere karşı deęişimsizliği oldukça fazla artmaktadır (Mikolajczyk 2005).

Yukarıda bahsedilen günümüz görüntü eşleştirme algoritmaları tatmin edici bir başarı yakalamaktadır. Fakat hiç birisi ilgin dönüşümlere karşı tamamen deęişimsizliği sağlayamamaktadır. (Yu 2009)'da, SIFT'in ilgin deęişimsiz bir uzantısı olan, tamamen ilgin deęişimsiz bir yöntem sunulmuştur.

Harris-Affine, Hessian-Affine ve MSER algılayıcıları ölçekleme deęişimlerine karşı zayıftırlar (Cao 2008). Diğer taraftan SIFT ve ASIFT (Yu 2009) yöntemlerinin her ikisi de ölçekleme deęişimlerine karşı sağlamdırlar.

İlgün dönüşüm altı parametreyle elde edildiği için bu parametrelerin hepsini benzetim (simulation) ile çözmek işlem yükünü çok fazla arttırmaktadır. Bir çok araştırmacı ilgin dönüşümü, benzetime alternatif olarak, Şekil 4.2'deki gibi normalleştirmeyle (normalization) hesaplamaya çalışmaktadır.

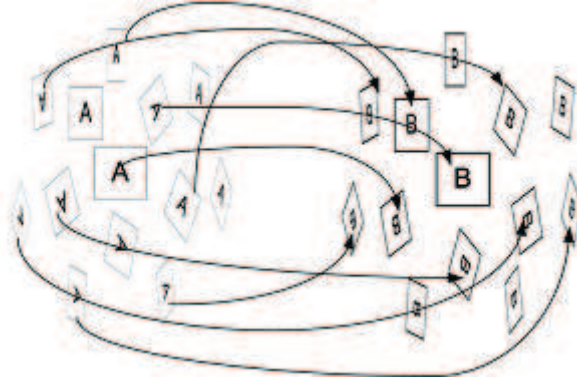


Şekil 4.2 Normalleştirme yöntemleri

a. Öteleme normalleştirme, b. dönme normalleştirme, c. eğme normalleştirme

SIFT'in başarımı, normalleştirmeye birlikte benzetimin de kullanılmasıyla sağlanmaktadır. SIFT algılayıcı, dönme ve ötelemeyi normalleştirmekte, ölçeklemeyi ise benzetmektedir. Bu sebeple SIFT sadece ölçeklemeye değişimsizdir.

Şekil 4.3'te de ifade edildiği gibi ASIFT yöntemi, kamera eksenindeki değişimlerin sebep olduğu tüm bozuklukları yeterli bir doğrulukla benzetmekte ve ardından SIFT yöntemini uygulamaktadır (Yu 2009). Başka bir deyişle, ASIFT, ölçekleme ve eğmeyi benzetmekte, öteleme ve dönmeyi ise normalleştirmektedir. ASIFT'te eğme hareketinin de benzetilmesine karşın, iki-çözünürlük planı uygulandığı için, işlem yükü SIFT'in sadece iki katıdır.



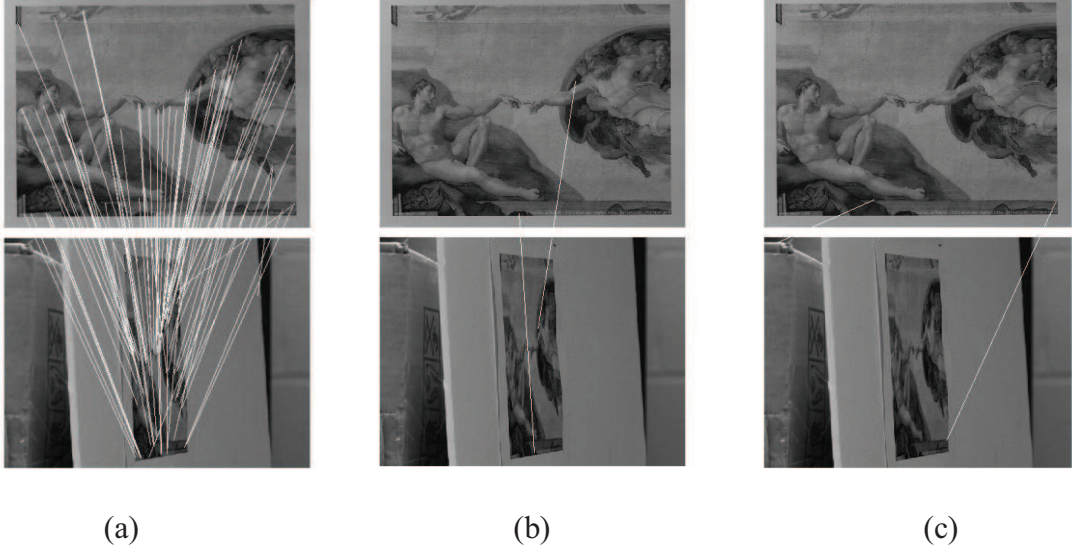
Şekil 4.3 ASIFT yöntemi

4.5 Öznitelik Çıkarımı Yöntemlerinin Karşılaştırılması

Günümüz görüntü eşleştirme algoritmaları arasında Hessian-Affine, Harris-Affine, MSER, SIFT ve ASIFT yöntemleri, aynı sahnenin farklı noktalardan alınmış iki görüntüsü arasında, buldukları ortak nokta sayılarına göre karşılaştırılmaktadırlar (Yu 2009). Karşılaştırmada kullanılan görüntülerin boyutu 600 x 450'dir.

Şekil 4.4'te aynı posterin farklı açılardan çekilmiş iki görüntüsü görünmektedir. Bu iki görüntü üzerine ASIFT, SIFT, Harris-Affine, Hessian-Affine ve MSER yöntemleri uygulandığında sırasıyla 77, 1, 0, 0, 2 ortak nokta çiftleri elde edilmektedir. Harris-Affine ve Hessian-Affine yöntemleri kullanılarak hiç ortak nokta çifti elde edilemediği

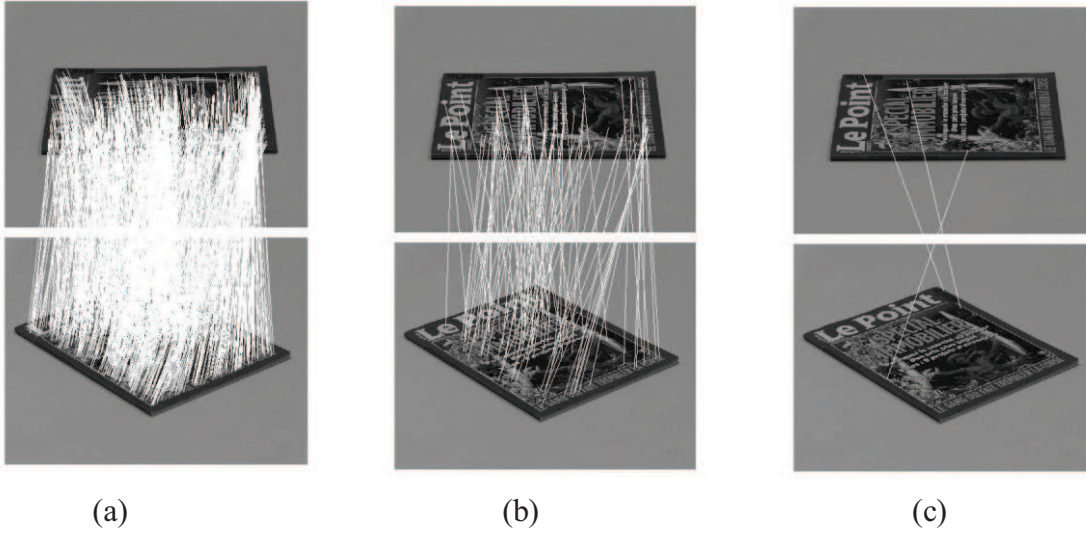
için Şekil 4.4'te bu yöntemlerin sonuçları gösterilmemektedir. Eğriltilmiş ve 10 kat yakınlştırılmış kamera ile 84° döndürülmüş poster üzerinde alınan sonuçlar incelendiğinde ASIFT yönteminin diğer yöntemlere göre çok daha başarılı olduğu görülmektedir. Posterin döndürölme açısı oldukça fazla olduđu için bu görüntülerde SIFT yöntemi başarısız olmaktadır.



Şekil 4.4 Aynı posterin farklı açılardan çekilmiş iki görüntüsü

a. ASIFT, b. MSER, c. SIFT kullanılarak elde edilen ortak nokta çiftleri

Şekil 4.5'teki dergi üzerine ASIFT, SIFT, Harris-Affine, Hessian-Affine ve MSER yöntemleri uygulandıđında sırasıyla 745, 3, 1, 3, 87 ortak nokta çiftleri elde edilmektedir.



Şekil 4.5 Aynı derginin farklı açılardan çekilmiş iki görüntüsü
a. ASIFT, b. MSER, c. SIFT kullanılarak elde edilen ortak nokta çiftleri

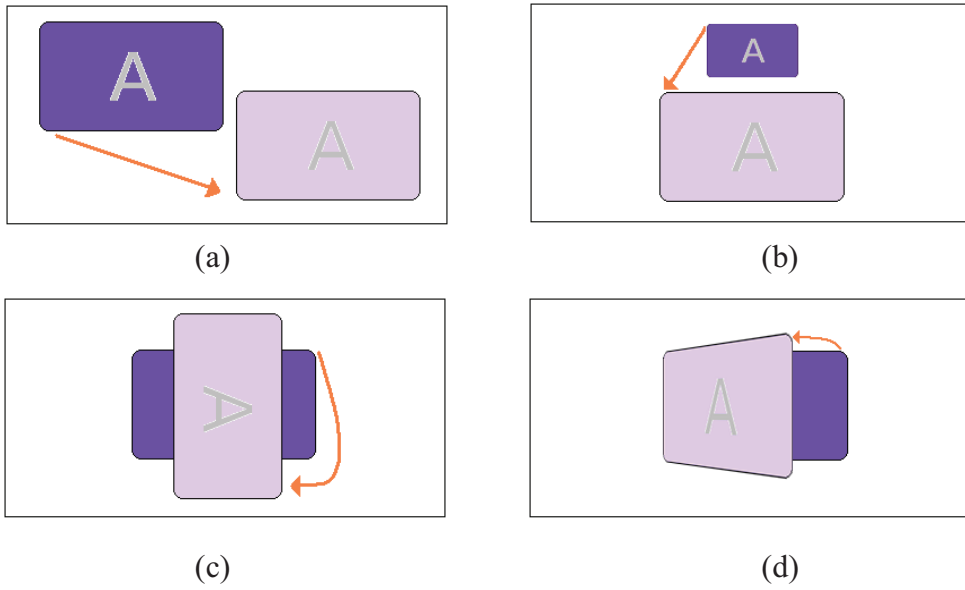
Şekil 4.5'teki durum Şekil 4.4'teki duruma göre çok daha kolay bir örnek olmasına karşın Harris-Affine, Hessian-Affine ve SIFT, yeterli sayıda ortak nokta verememektedir. Görüntüler arasında ölçekleme değişikliği olmadığı için MSER başarılı sonuç vermektedir. ASIFT yöntemiyle elde edilen sonuç ise oldukça tatmin edicidir.

Çıkartılan özniteliklerin takip edilebilmesi için hareket modellerinin belirlenmesi gerekmektedir. Bir sonraki bölümde kullanılan hareket modellerinden bahsedilmektedir.

4.6 Hareket Modelleri

Nesnenin hareketini temsil etmek için seçilen ortak nokta sayısına bağlı olarak hareket modeline karar verilmektedir. Bu noktaların sayısı, nesnenin hareket modelini kısıtlamaktadır. Örneğin hareket, sadece bir nokta çifti ile temsil ediliyorsa kullanılacak model, Şekil 4.6.a'daki gibi yalnızca öteleme hareketidir. Hareket, iki nokta ile temsil ediliyorsa eşyönsüz ölçekleme ve öteleme hareket modeli nesnenin hareketi için kullanılabilir. Nesnenin hareketini temsil etmek için elimizde üç tane ortak nokta çifti olduğu durumda ise hareket, ilgin hareket modeli ile

hesaplanabilmektedir. Dört ya da daha fazla ortak nokta çifti mevcut ise izdüşümsel hareket modeli ile nesnenin hareketi bulunabilmektedir (Dubrofsky 2009). Bu temsiller esnek olmayan nesnelere hareketini çözümlmek için uygundur. Esnek bir nesnenin temsili içinse, silüet ve çevre çizgisi modelleri en uygundur. Nesnenin hareketini tanımlamak için hem parametrik hem de parametrik olmayan modeller kullanılabilir (Yılmaz 2006).



Şekil 4.6 Hareket modelleri

a. Öteleme hareketi, b. ölçekleme hareketi, c. dönme hareketi, d. izdüşümsel hareket

Nesne üzerinde bir tane ortak nokta çifti $((x, y), (x', y'))$ bulunması halinde, nesnenin x yönündeki öteleme miktarı m ve y yönündeki öteleme miktarı n , aşağıdaki denklemde m ve n yalnız bırakılarak, bulunabilmektedir.

$$x' = x + m \quad (4.1)$$

$$y' = y + n$$

Eğer bölge içerisinde bir tane daha çift olduğu takdirde, eşyönlü ölçekleme ve öteleme hareketi, denklem 4.2'deki gibi

$$\begin{aligned}x' &= ax + m \\y' &= by + n\end{aligned}\tag{4.2}$$

ya da dönme, eşyönlü ölçekleme ve öteleme hareketi, denklem 4.3'teki gibi modellenmektedir.

$$\begin{aligned}x' &= a \cos \theta x - a \sin \theta y + m \\y' &= a \sin \theta x + a \cos \theta y + n\end{aligned}\tag{4.3}$$

Bölge içerisinde üçüncü ortak noktanın da var olması durumunda denklem 4.4'teki tüm ilgin hareket modeli hesaplanabilmektedir (Lee 1997).

$$\begin{aligned}x' &= ax + by + m \\y' &= cx + dy + n\end{aligned}\tag{4.4}$$

Bölge içerisinde dört ya da daha fazla ortak nokta bulunması durumunda da, öteleme, dönme (Şekil 4.6.c), ölçekleme (Şekil 4.6.b) hareketlerinin yanı sıra Şekil 4.6.d'deki gibi izdüşümü de içeren izdüşümsel hareket modeli, denklem 4.5'teki gibi bulunmaktadır (Hartley 2003).

$$\begin{aligned}x' &= \frac{ax + by + m}{px + qy + 1} \\y' &= \frac{cx + dy + n}{px + qy + 1}\end{aligned}\tag{4.5}$$

5. ÖNERİLEN HAREKET KESTİRİMİ VE DENGELEMESİ YÖNTEMİ

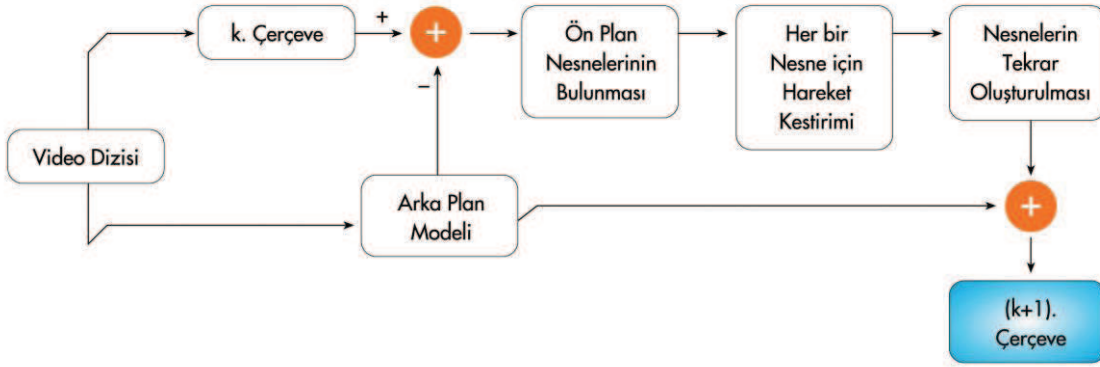
5.1 Giriş

Bu tezde, hareket kestirimi ve dengelemesi için nesneye dayalı bir yöntem önerilmektedir. Önerilen yöntemde, bir video dizisinde yer alan arka plan modellenerek ön plan nesnelерinin bulunmasında kullanılmaktadır. Her bir çerçeve (k . çerçeve), bu arka plan modelinden çıkartılarak çerçeve içerisinde yer alan ön plan nesneleri elde edilmektedir.

Ön plan nesnesinin hareketi temel olarak beş aşamanın ardından hesaplanmaktadır. İlk olarak, ardışık çerçevelerdeki her bir hareketli ön plan nesnesi üzerinde, bir öznitelik çıkarma yöntemi kullanılarak, ortak nokta çiftleri bulunmaktadır. Daha sonra, çıkartılan ortak nokta çiftleri, nesne üzerinde oluşturulacak bölgelerin çekirdeklerini oluşturmak üzere, kümelere ayrılmaktadır. Ardından, kümelenen ortak nokta çiftlerinin etrafındaki pikseller de kümeye dahil edilerek nesne üzerinde bölgeler ortaya çıkartılmaktadır. Daha sonra, aynı bölgeye ait ortak nokta çiftlerinin sayısına ve birbirine olan durumlarına bakılarak uygulanacak hareket modeli belirlenmektedir. Son olarak, her bir bölgenin hareket dönüşümünün, belirlenen hareket modeline göre, hesaplanmasıyla ön plan nesnesi için hareket kestirimi tamamlanmaktadır.

Hareket dengelemesi aşamasında, ilk olarak mevcut çerçevedeki ön plan nesnesinin her bir bölgesine o bölge için kestirilen hareket dönüşümü uygulanmaktadır, daha sonra aradeğerleme (interpolation) yapılarak bölge içerisinde ve bölgeler arasında oluşan boşluklar doldurulmaktadır. Son olarak da, tekrar oluşturulan nesne, arka plan modeline eklenerek bir sonraki çerçeve ($(k+1)$. çerçeve) oluşturulmaktadır.

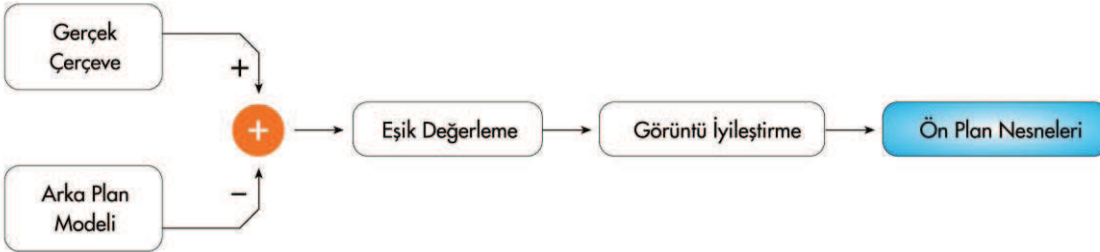
Yukarıda da aşamalarından bahsedilen ve önerilen yöntemin genel blok şeması Şekil 5.1'de gösterilmektedir.



Şekil 5.1 Önerilen yöntemin genel blok şeması

5.2 Ön Plan Nesnelerinin Bulunması

Ön plan nesnelerinin bulunması için uygulanan yöntemin blok şeması Şekil 5.2'deki gibidir. Hareketli ön plan nesnelerinin bulunması için öncelikli olarak video dizisinin arka planı ayrılmaktadır. Video dizisindeki arka planı ayırmak için zamansal ortanca süzgeç (temporal median filter) (Massey 1996) uygulanmaktadır.

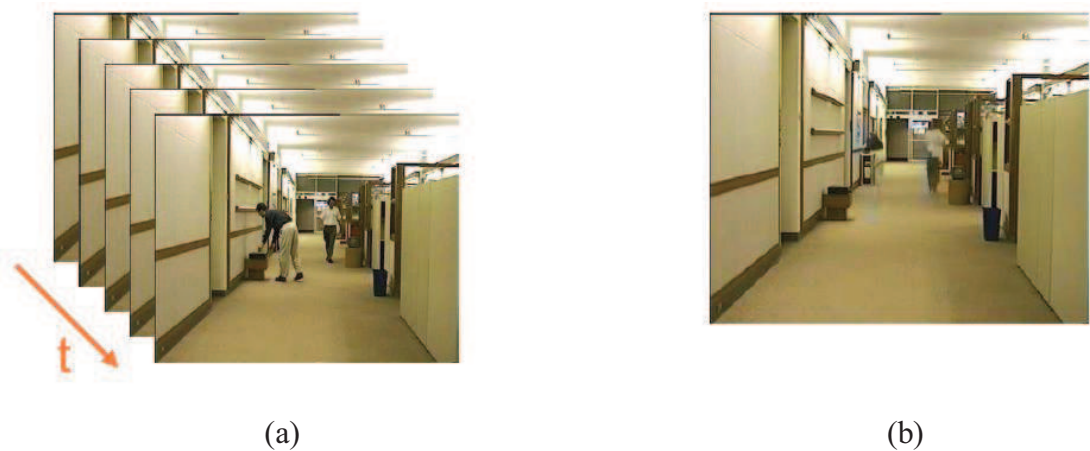


Şekil 5.2 Ön plan nesnelerinin bulunması

Zamansal ortanca süzgeç yönteminde, her bir çerçevede aynı konumda yer alan pikseller bir dizide tutulmaktadır. Elde edilen bu dizinin ortanca değeri arka plan modelinin o konumunun piksel değerini vermektedir. Örnek vermek gerekirse, bir video dizisinde her bir çerçevenin (100, 150) konumundaki pikselleri tek bir dizide tutulmakta ve bu dizinin ortanca değeri arka plan modelinin (100, 150) pikselinin değeri olarak atanmaktadır. Arka plan modelinin güncel olmasını sağlayabilmek için denklem 5.1 kullanılmaktadır.

$$\mathbf{B}_{s,t+1} = (1 - \alpha)\mathbf{B}_{s,t} + \alpha\mathbf{I}_{s,t} \quad (5.1)$$

Burada, $\mathbf{B}_{s,t+1}$, s pikselinde $t+1$ zamandaki arka planı, $\mathbf{I}_{s,t}$, s pikselinde t zamanındaki video çerçevesini ve α ise güncelleme sabitini ifade etmektedir ve 0 ile 1 arasında değer almaktadır. Güncelleme sabiti (α) için uygulamaya göre değişik değerler kullanılmakla birlikte genelde 0.05 değeri tercih edilmektedir. Zamansal ortanca süzgeçten elde edilen arka plan modeli Şekil 5.3'te görülmektedir.



Şekil 5.3 Zamansal ortanca süzgeç

a. Video dizisi, b. zamansal ortanca süzgeç ile elde edilmiş arka plan modeli

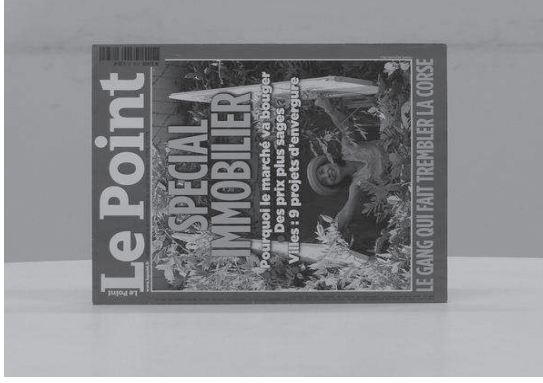
Zamansal ortanca süzgeç sonucu belirlenen görüntü, video dizisinin arka plan modeli olarak kabul edilmektedir. Arka planın modellenmesinin ardından, tüm çerçeveden arka plan modeli çıkarılmaktadır. Daha sonra bir eşik değer uygulanarak nesnenin silueti ikili görüntü (binary image) olarak elde edilmektedir. Eşik değeri, otomatik olarak hesaplanmakta ve bu eşik değeri bulabilmek için Otsu yöntemi kullanılmaktadır.

Nesnenin siluetini gerçeğine daha yakın elde edilebilmesi için siluet üzerinde morfolojik işlemler uygulanmaktadır. İlk uygulanan morfolojik işlem, çoğunluk (majority) işlemidir. Bu işlemde, pikselin değerini belirlemek için sekiz komşusuna bakılır ve çoğunluğun değerine göre pikselin değeri belirlenmektedir. Örneğin bir pikselin komşuluğunda beş veya daha fazla sayıda 1 (beyaz) varsa, piksel 1 değerini almaktadır.

Çoğunluk işleminin ardından algoritması Şekil 5.4'te ifade edilen yöntemle istenmeyen küçük beyaz nesnelere yok edilmekte ve küçük siyah boşluklar doldurulmaktadır. Yöntemde, ilk olarak bağlı bileşen analizi kullanılarak ikili görüntüde yer alan nesnelerin sayısı elde edilir ve nesnelere etiketlenir. Daha sonra her bir nesnenin alanı bulunarak daha önceden belirlenmiş bir eşik değeri ile karşılaştırılarak nesnenin küçük ya da büyük olduğuna karar verilir. Ardından küçük nesnelere görüntüden çıkartılır. Bu şekilde görüntüde yer alan küçük beyaz nesnelere görüntüden çıkartılmaktadır. Ön plan nesnesi içerisinde yer alan küçük siyah boşlukları doldurmak için ise yine aynı algoritma kullanılmaktadır fakat girdi olarak ikili görüntünün tersi algoritmaya verilmektedir. Böylelikle nesnenin gerçek silüetine yakın bir silüet elde edilebilmektedir.

1. Çerçeve içinde yer alan nesnelere bağlı bileşen analizi ile etiketle ($i=1, \dots, n$),
2. Bir eşik değeri (th) belirle,
3. En küçük etikete ($i=1$) sahip nesneyi seç,
4. Nesnenin alanını ($Alan_i$) bul,
5. Eğer $Alan_i < th$ ise nesneyi küçük olarak işaretle,
6. $i < n$ ise bir sonraki nesneyi ($i=i+1$) seç, 4. adıma git,
7. Küçük olarak işaretlenen nesnelere görüntüden çıkart.

Şekil 5.4 Küçük nesnelere yok etmek için geliştirilen yöntemin algoritması



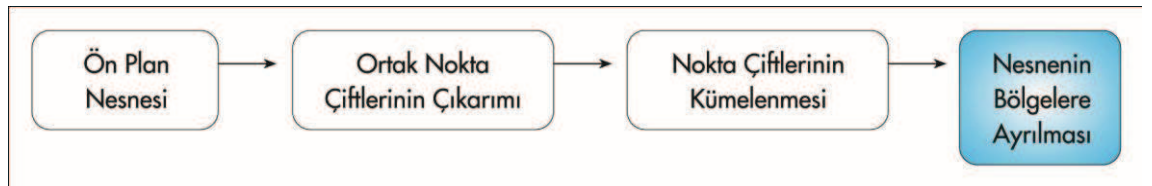
Şekil 5.5 Ön plan nesnesinin elde edilmesi

a. Orijinal çerçeve, b. ön plan nesnesi

Şekil 5.5.b’de arka plandan ayrılmış ön plan nesnesi (dergi) görülmektedir. Parks tarafından 2008 yılında yayınlanan, arka plan çıkarımı yöntemlerinin başarımlarının karşılaştırıldığı makaleye göre ön plan nesnelere bulunması için kullanılan bu yöntemin başarımının yüksek olmasının yanı sıra karmaşıklığı az ve kodlaması da kolaydır. Bu sebeple tezde ön plan nesnelere elde etmek için bu yöntem kullanılmaktadır.

5.3 Nesnenin Bölgelere Ayrılması

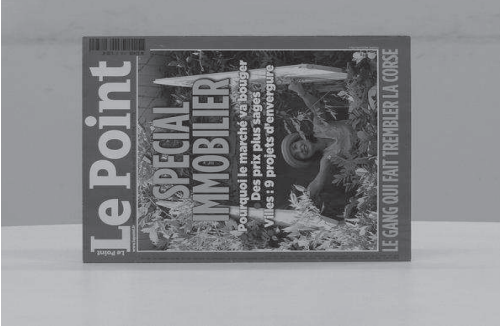
Bazı ön plan nesnelere hareketini tek bir modelle temsil etmek mümkün olmayacağı için nesne, hareketlerin farklı olduğu bölgelere, Şekil 5.6’da gösterildiği gibi ayrılmaktadır.



Şekil 5.6 Nesnenin bölgelere ayrılmasının blok şeması

Nesne üzerinde bölgeleri oluşturmak için öncelikle ardışık nesnelere üzerinde ortak nokta çiftleri çıkarılmaktadır (Şekil 5.7). Ortak nokta çiftleri bulunurken ilgin

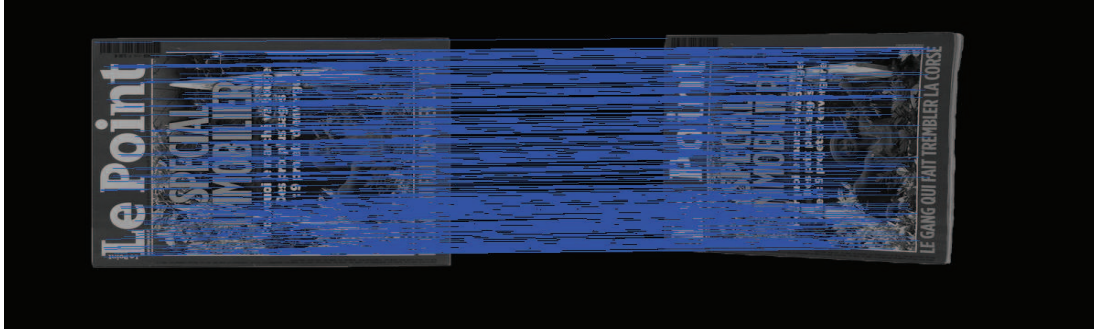
dönüşümlere karşı da tamamen değişimsiz olan ASIFT yöntemi kullanılmaktadır (Yu 2009).



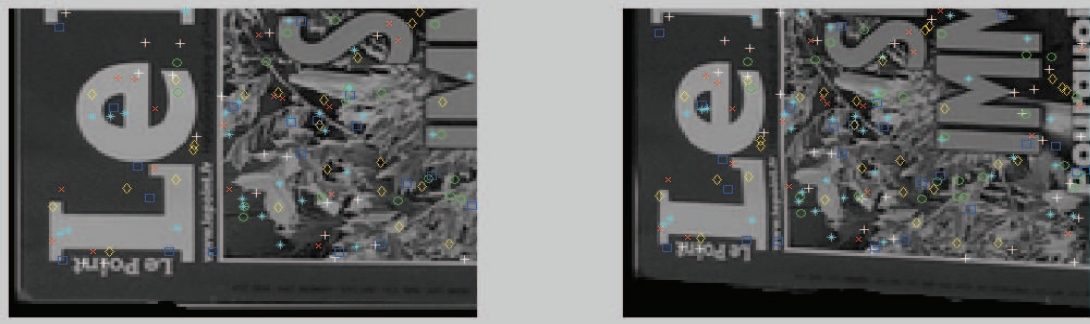
(a)



(b)



(c)



(d)

Şekil 5.7 Ortak noktaların çıkartılması

- a. Mevcut çerçeve, b. sonraki çerçeve, c. ön plan nesnelere üzerinde ASIFT kullanılarak ortak nokta çiftlerinin çıkartılması, d. nesnenin yakınlıştırılarak ortak nokta çiftlerinin gösterilmesi

Ortak nokta çiftlerinin çıkarılmasında ASIFT tercih edilmesinin en önemli nedeni çok sayıda ortak nokta çiftinin elde edilebiliyor olmasıdır. En doğru hareket dönüşümünü

hesaplayabilmek için elde edilen bu ortak nokta çiftleri arasından en uygun olanları seçilecektir. ASIFT yöntemi kullanılarak elde edilen ortak nokta çiftleri Şekil 5.7.c'de görülmektedir. Burada da görüldüğü gibi ASIFT ile aynı nesnenin farklı görüntüleri üzerinde çok sayıda ortak nokta sayısı elde edilebilmektedir. ASIFT sonucunun doğruluğunu gösterebilmek amacıyla Şekil 5.7.c'deki nesne yakınlıştırılarak üzerinde elde edilen ortak nokta çiftleri Şekil 5.7.d'de gösterilmektedir. Şekil 5.7.d'de, birbirinden ayırt edilebilmesi için ortak nokta çiftleri, baklava (\diamond), kare (\square), yuvarlak (\circ), yıldız (*), çarpı (x) ve artı (+) şekilleriyle ve farklı renklerde gösterilmektedir.



(a)



(b)



(c)

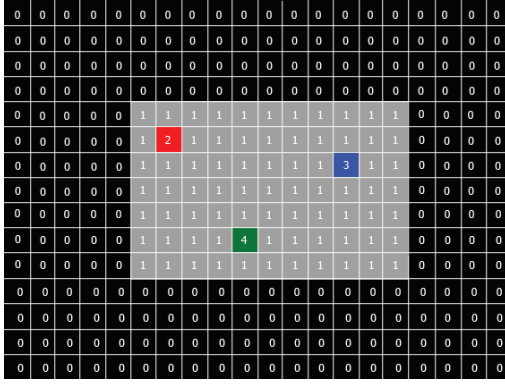
Şekil 5.8 Nesnenin bölgelere ayrılmasının aşamaları

- a. Mevcut çerçeve üzerindeki ortak noktalar, b. ortak noktaların kümelenmesi, c. nesnenin bölgelere ayrılması

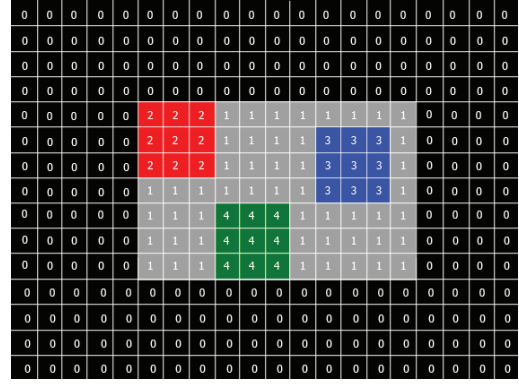
Şekil 5.8.a'da, ASIFT kullanılarak elde edilen ortak nokta çiftleri görülmektedir. Bu noktalar Şekil 5.8.b'deki gibi *K-Means Clustering* (Lloyd 1982) ile etiketlenerek kümelenebilir. *K-Means Clustering* ile kümelenen noktalar, bölgelerin çekirdek piksellerini oluşturmaktadır. Daha sonra bu çekirdek pikselleri kullanılarak Şekil 5.8.c'de görüldüğü gibi nesne bölgelere ayrılmaktadır.

Nesneyi bölgelere ayırmak için yinelemeli bir yöntem geliştirilmiştir. Geliştirilen yöntem, Şekil 5.9'da gösterilen bir örnek ile anlatılmaktadır. Örnekte *K-Means Clustering* ile kümelendiği varsayılan çekirdek pikseller Şekil 5.9.a'da görünmektedir. Yöntemde, ön plan nesnelерinin pikselleri etiketlenirken, çekirdek piksel sayısı fazla olan kümeye öncelik verilmektedir. Böylece daha fazla çekirdek piksel sayısına sahip olan kümenin etki alanı daha fazla olmaktadır. Örnekte iki ile etiketlenen küme en öncelikli kümedir, üç ile etiketlenen küme de dört ile etiketlenen kümeden daha önceliklidir. Algoritmada ilk yinelemede, çekirdek piksellerin birinci derece komşuluğundaki etiketlenmemiş ön plan pikselleri Şekil 5.9.b'deki gibi o çekirdek pikselin etiketiyle etiketlenmektedir. Bir sonraki yinelemede, çekirdek piksellerin ikinci derece komşuluğundaki etiketlenmemiş ön plan pikselleri Şekil 5.9.c'deki gibi o çekirdek pikselin etiketiyle etiketlenmektedir. Burada önceliğin sırasıyla iki, üç ve dört ile etiketlenen kümelerde olduğu görülmektedir. Yöntemde, ön plan piksellerinin her biri etiketlenene kadar yinelemeye devam edilmektedir. Yöntem sonucunda nesnenin bölgelere ayrılmış görüntüsü Şekil 5.9.d'de görülmektedir.

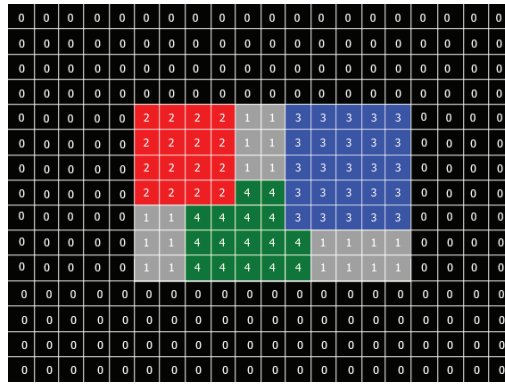
Geliştirilen bu yöntem, bölge büyümesine (Region Growing) (Prat 2007) benzemesine karşın yöntemde, bölge büyümesinde olduğu gibi komşu piksellerin parlaklık değerleri arasında benzerlik aranmamaktadır. Bu yöntem, uzaklığa bağlı olarak bölge ataması yapan diğer yöntemlere nazaran daha hızlı çalışmaktadır. Bu yöntem ile ön plan nesnesi üzerinde hareket kestirimi yapılacak bölgeler hızlı bir biçimde ayrılmaktadır.



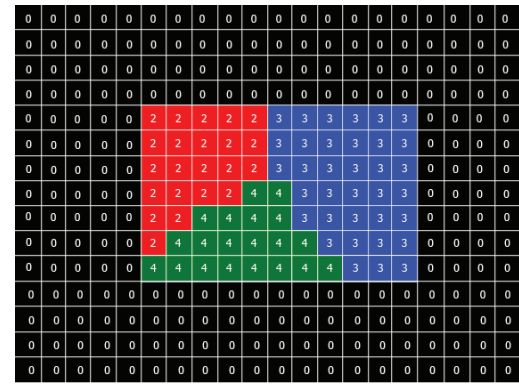
(a)



(b)



(c)

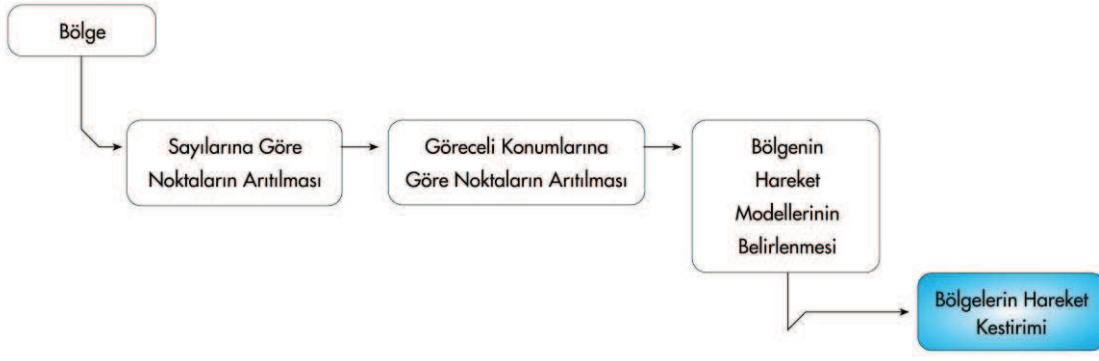


(d)

Şekil 5.9 Nesnenin bölgelere ayrılması için bir örnek
a.İlk yineleme, b. ikinci yineleme, c. üçüncü yineleme, d. son durum

5.4 Hareket Kestirimi

Hareket kestirimi için önerilen yöntemin blok şeması Şekil 5.10'da gösterilmektedir. Önerilen yöntemde, ön plan nesnesinin bölgelerine ayrılmasından sonra bölge içerisinde yer alan ortak nokta sayısına ve noktaların birbirlerine olan konumlarına bağlı olarak bölgenin hareket modeline karar verilmektedir. Bölge içerisindeki noktalardan uygun olanlar seçilirken öncelikle ortak nokta çiftlerinin sayısına bakılmaktadır. Daha sonra ortak nokta çiftlerinin birbirleriyle göreceli konumlarına göre hareket modeline en uygun noktalar seçilmektedir.



Şekil 5.10 Hareket kestirimi blok şeması

Öteleme hareket modeli, x ve y yönündeki ötelemeden oluşmaktadır. Ötelemeyi hesaplayabilmek için sadece bir tane ortak nokta çiftinin elde edilmesi yeterli olmaktadır. İlgin hareket modeli, bilinmeyen altı parametre içermektedir. Bu altı parametrenin bulunmasıyla ilgin hareket modeli elde edilebilmektedir. Altı bilinmeyenli denklem sistemini çözebilmek için birbirinden bağımsız en az altı denkleme ihtiyaç duyulmaktadır. Bu sebeple, ilgin hareket modelini çözebilmek için üç ortak nokta çiftinin elde edilmesi yeterlidir. İlgin hareket modelini mümkün olduğunca doğru hesaplayabilmek için ise noktaların dar açılı bir üçgen oluşturması gerekmektedir. Bölge içerisinde bir tane daha ortak nokta çiftinin elde edilebilmesi durumunda ise sekiz parametreden oluşan izdüşümsel hareket modeli hesaplanabilmektedir. Mümkün olduğunca en doğru izdüşümsel hareket modelini elde edebilmek için, seçilen noktaların dikdörtgensel bir şekil oluşturması gerekmektedir.

Denklem 4.1'deki öteleme hareket modelini kullanarak hareketi kestirebilmek için eşitlik,

$$PT = P' \quad (5.2)$$

biçimine getirilmektedir. Burada P , dönüşüm uygulanacak noktalar matrisini, T , dönüşümü ve P' , dönüşüm sonucu elde edilen matrisi ifade etmektedir. 2×2 'lik bir matris halinde öteleme hareketini modellemek mümkün olmadığı için (x, y) ve (x', y') nokta çiftlerine denklem 5.3'teki gibi bir üçüncü eleman eklenmektedir (Rogers 1976).

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = \begin{bmatrix} x+m & y+n & 1 \end{bmatrix} = \begin{bmatrix} x' & y' & 1 \end{bmatrix} \quad (5.3)$$

Denklem 4.4'teki ilgin hareket modelini çözmek için eşitliği matris halinde yazdığımızda

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix} = \begin{bmatrix} ax_1+cy_1+m & bx_1+dy_1+n & 1 \\ ax_2+cy_2+m & bx_2+dy_2+n & 1 \\ ax_3+cy_3+m & bx_3+dy_3+n & 1 \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix} \quad (5.4)$$

elde edilmektedir. Burada $a=1$, $b=0$, $c=0$ ve $d=1$ olduğunda sadece öteleme hareket modeli elde edilebilmektedir. Öteleme hareket modeli, m ve n yalnız bırakılarak çözülebilmektedir.

İlgin hareket modelini hesaplamak için öncelikli olarak denklem 5.4'te P matrisinin tersi alınmakta ve eşitliğin her iki tarafına, denklem 5.5'teki gibi soldan çarpılmaktadır. Matrisin tersi hesaplanırken Tekil Değer Ayrışımı (TDA, SVD-Singular Value Decomposition) (Hestenes 1958) kullanılmaktadır.

$$P^{-1}PT = T = P^{-1}P' \quad (5.5)$$

Denklem 4.5'teki izdüşümsel hareket modelini çözmek için eşitliği, n tane ortak nokta çifti için, matris halinde tekrar yazdığımızda,

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} a \\ c \\ m \\ b \\ d \\ n \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.6)$$

elde edilmektedir. Bu eşitlik,

$$\mathbf{Ax} = \mathbf{0} \quad (5.7)$$

gibi bir eşitlik olduğu için, Homojen En Küçük Doğrusal Kareler (Homogeneous Linear Least Squares) problemi olarak bilinmektedir. Denklem 5.7,

$$\mathbf{Ax} = \mathbf{b} \quad (5.8)$$

ile benzer bir yapıda olmasına rağmen denklem 5.8, \mathbf{A} 'nın tersi ya da yalancı tersi (pseudo inverse) alınarak çözülebilmektedir. Fakat denklem 5.7 için böyle bir çözüm uygun değildir.

Denklem 5.7 ise TDA yardımıyla çözülebilmektedir. \mathbf{A} matrisinin TDA'sı alındığında,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^9 \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (5.9)$$

elde edilmektedir. Burada \mathbf{U} , $2n \times 2n$ 'lik, $\mathbf{\Sigma}$, $2n \times 9$ 'luk ve \mathbf{V} ise 9×9 'luk bir matris, \mathbf{u}_i ve \mathbf{v}_i , sırasıyla \mathbf{U} ve \mathbf{V} 'nin birbirine dik sütun vektörleri, σ_i , $\mathbf{\Sigma}$ 'nin köşegen elemanları yani tekil değerlerdir. \mathbf{V} matrisinin son sütünü (\mathbf{v}_9), en küçük tekil değere karşılık gelen sağ tekil vektörü vermektedir. Bu vektör, denklem 5.6'nın sol tarafını 0'a en çok yaklaştıracak vektördür (Langer 2009). Böylelikle,

$$\mathbf{v}_g = \mathbf{h} = \begin{bmatrix} a \\ c \\ m \\ b \\ d \\ n \\ p \\ q \\ r \end{bmatrix} \quad (5.10)$$

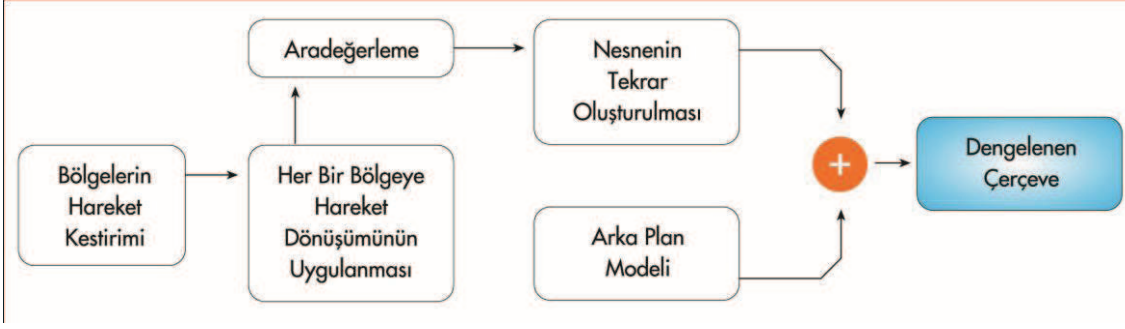
elde edilmektedir. Buradan \mathbf{h} vektörü 5.11'deki gibi 3 x 3'lük bir matris haline getirilmektedir.

$$\mathbf{T} = \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & r \end{bmatrix} \quad (5.11)$$

Bu yolla izdüşümsel dönüşüm matrisi \mathbf{T} bulunabilmektedir (Kriegman 2007).

5.5 Hareket Dengelemesi

Çerçeveler arasında ön plan nesnesinin hareket kestirimi yapılmasının ardından sonraki çerçevedeki nesneyi tekrar oluşturmak için Şekil 5.11'de blok şeması gösterilen hareket dengelemesi yapılmaktadır.



Şekil 5.11 Hareket dengelemesi blok şeması

Ön plan nesnesi üzerindeki her bölge için hesaplanan hareket dönüşümü T , o bölgeye uygulanmaktadır. Dönüşüm uygulandıktan sonra, özellikle hareket bölgesinin kameraya yaklaştığı durumlarda ve bölgeler arasında, boşluklar meydana gelebilir (Şekil 5.12.a). Böyle durumlarda, boşluklar, karmaşıklığı düşük olduğu ve yeterli sonuç verdiği için, çift doğrusal aradeğerleme (Bilinear Interpolation) ile doldurulmaktadır (Lee 1997) (Şekil 5.12.b). Şekil 5.12.c'de de sonraki orijinal çerçevedeki nesnenin görüntüsü görülmektedir. Böylelikle sonraki çerçevenin nesnesi tekrar oluşturulmaktadır. Tekrar oluşturulan nesne, arka plan modeli üzerine yerleştirildiğinde sonraki çerçeve için hareket dengelenmesi tamamlanmaktadır.



(a)



(b)



(c)

Şekil 5.12 Nesnenin tekrar oluşturulması

a. Mevcut çerçeveye kestirilen hareket dönüşümü uygulandıktan sonra, b. oluşan boşlukların çift doğrusal aradeğerleme ile kapatılması, c. orijinal nesne

6. DENEYSEL BULGULAR

Önerilen yöntemin başarımını test etmek için her biri öteleme, dönme ve izdüşüm hareketi yapan nesnelerin görüntüleri kullanılmaktadır. Öteleme testinde, çerçevede nesnenin az ve çok yer kapladığı iki video kullanılmaktadır. Bu şekilde farklı büyüklükteki nesnelerin yöntemlerin başarımına etkisi gözlemlenmektedir. Dönme testinde ise düşük (320 x 240) ve yüksek (800 x 592) çözünürlükte iki video kullanılarak çözünürlüğün hareket kestirimi yöntemleri üzerinde etkisi incelenmektedir. Bu hareket modellerini içeren hazır test görüntüleri bulunamadığı için test videoları farklı ortamlarda görüntü alınarak hazırlanmıştır.

Dengelenen çerçevenin orijinal çerçeveye benzerliğini ölçmek için kullanılan nesnel video kalitesi ölçütü genellikle Tepe Sinyal Gürültü Oranıdır (PSNR, Peak Signal to Noise Ratio) (Huynh-Thu 2008). PSNR, MSE'ye bağlı olarak denklem 6.2'deki gibi elde edilmektedir.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_O(i, j) - I_D(i, j)]^2 \quad (6.1)$$

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (6.2)$$

Bu denklemlerde, I_O , $m \times n$ boyutlarındaki orijinal çerçeveyi, I_D , dengelenmiş çerçeveyi, MAX_I ise görüntünün alabileceği en yüksek değeri ifade etmektedir.

Önerilen yöntemin başarısı, bloğa dayalı hareket kestirimi yöntemleriyle karşılaştırılmıştır. Bu yöntemler; Tam Arama (ES, Exhaustive Search), Üç Adım Arama (TSS, Three Step Search), Yeni Üç Adım Arama (NTSS, New Three Step Search) (Li 1994), Basit ve Etkili Üç Adım Arama (SESTSS, Simple and Efficient Three Step Search) (Lu 1997), Dört Adım Arama (FSS, Four Step Search) (Po 1996), Baklava

Arama (DS, Diamond Search) (Zhu 2000) ve Uyarlamalı armih rüntü Aramadır (ARPS, Adaptive Rood Pattern Search) (Nie 2002).

Öteleme Testleri:

Çerçeve de az yer kaplayan nesnenin öteleme testini gerçekleştirmek için kameranın doğrultusuna dik yönde hareket eden bir arabanın görüntüsü 800 x 592 piksel boyutlarında alınmıştır. Ardışık 30 çerçeve üzerine, önerilen yöntem ve bloğa dayalı hareket kestirimi yöntemleri uygulanmış ve PSNR sonuçları ile nesnel video kalitesi elde edilmiştir. Bloğa dayalı hareket kestirimi yöntemleri için 16 x 16 piksellik blok büyüklüğü, +/- 7 x 7 piksellik de arama alanı kullanılmıştır. Bloğa dayalı hareket kestirimi yöntemleri arasında en başarılı sonucu veren tam arama yönteminde ayrıca 8 x 8 piksellik blok büyüklüğü kullanılarak elde edilen sonuçlar gösterilmiştir.

Çerçeve de az yer kaplayan nesnenin öteleme testi için hazırlanan videonun 14. ve 15. çerçevesi Şekil 6.1.a ve b'de görölmektedir.



(a)



(b)



(c)



(d)

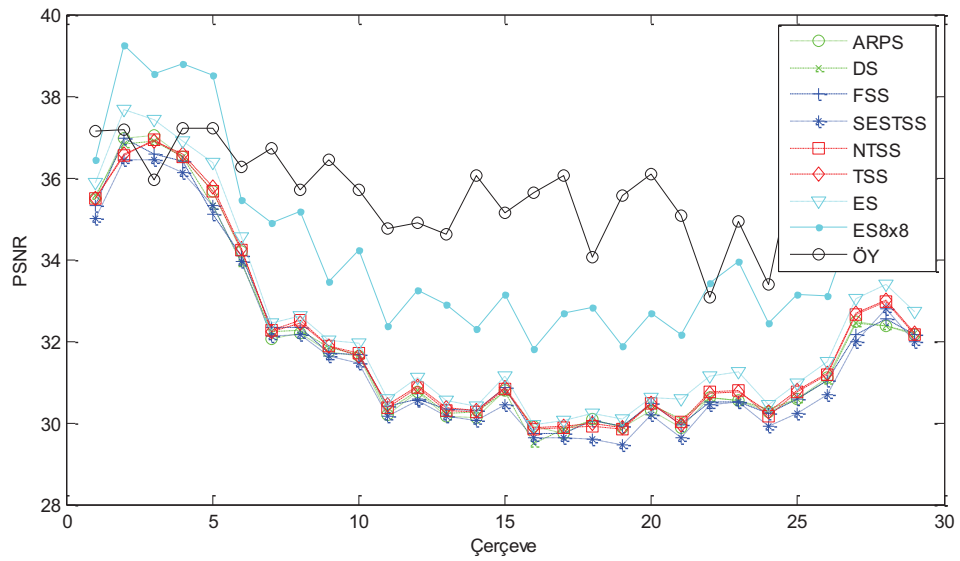


(e)

Şekil 6.1 Çerçeve az yer kaplayan nesnenin öteleme testi görüntüleri

a. 14. çerçeve, b. 15. çerçeve, c. 16 x 16 piksellik tam arama yöntemiyle dengelenmiş 15. çerçeve (PSNR=30.3963), d. 8 x 8 piksellik tam arama yöntemiyle dengelenmiş 15. çerçeve (PSNR=32.3038), e. önerilen yöntemle dengelenmiş 15. çerçeve (PSNR=36.0549)

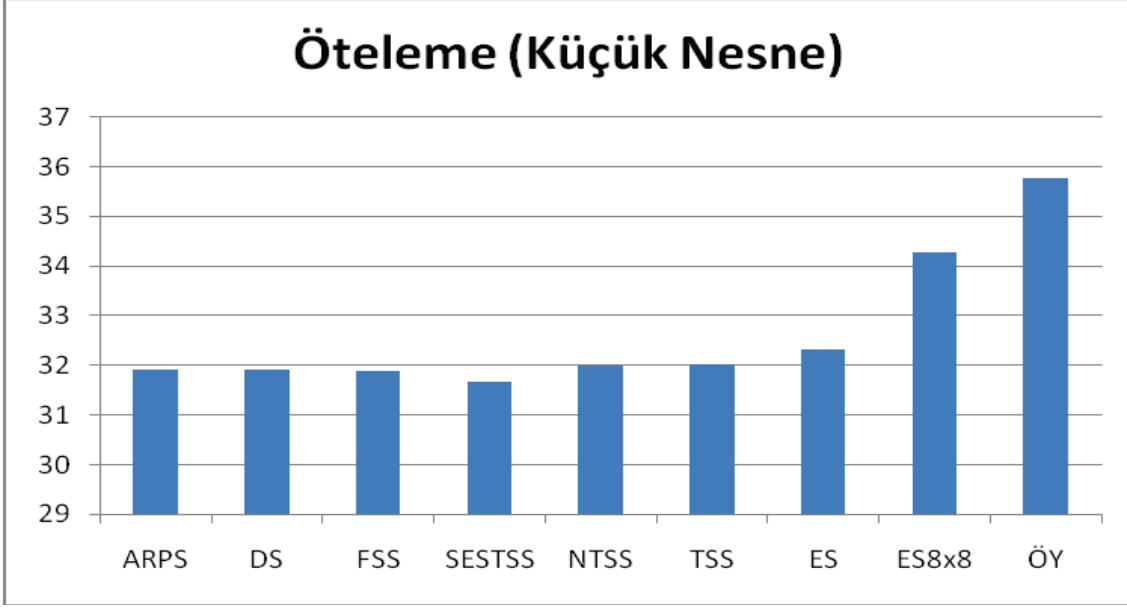
Şekil 6.1.c'de görülen ve tam arama yönteminde 16 x 16 piksellik blok büyüklüğü kullanılan dengelemede PSNR, 30.3963'tür. Şekil 6.1.d'de görülen 8 x 8 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 32.3038'dir. Şekil 6.1.e'de görülen önerilmiş yöntemle gerçekleştirilen dengeleme sonucu ise PSNR, 36.0549 çıkmıştır. Ardışık 30 çerçeve üzerinde uygulanan yöntemler sonucu elde edilen PSNR değerleri Şekil 6.2'de gösterilmektedir.



Şekil 6.2 Çerçeve başına az yer kaplayan nesnenin öteleme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri

Test videosunda araba, ilk 5 çerçevede diğer çerçevelere göre daha yavaş hareket etmektedir. Bu çerçevelerde Şekil 6.2'de de görüldüğü gibi yöntemlerin PSNR değerleri birbirine yakın çıkmaktadır. Hareketin daha fazla olduğu diğer çerçevelerde ise önerilen yöntemin (ÖY) başarımı oldukça iyidir. Bu başarımın en büyük nedeni, önerilen yöntemde hareket kestirimi nesnenin hızından bağımsız olmasıdır. Nesne ne kadar hızlıysa diğer yöntemlere oranla önerilen yöntemde daha fazla başarı elde edilmektedir.

Test videosu üzerinde her bir çerçeve için elde edilen PSNR sonuçlarının ortalaması Şekil 6.3'te verilmektedir.



Şekil 6.3 Çerçeve az yer kaplayan nesnenin öteleme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri

Şekil 6.3'teki ortalama PSNR değerlerine bakıldığında çerçevede az yer kaplayan nesnenin öteleme testinde önerilen yöntem, 16 x 16 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre yaklaşık olarak 3.5 dB, 8 x 8 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre ise yaklaşık 1.5 dB daha iyi sonuç vermektedir.

Çerçevede çok yer kaplayan nesnenin öteleme testini gerçekleştirmek için, çerçevede az yer kaplayan nesnenin öteleme testinde olduğu gibi, kameranın doğrultusuna dik yönde hareket eden bir arabanın görüntüsü 640 x 480 piksel boyutlarında alınmıştır. Test videosuna önerilen yöntem ve bloğa dayalı hareket kestirimi yöntemleri uygulanmış ve PSNR sonuçları ile nesnel video kalitesi elde edilmiştir. Bloğa dayalı hareket kestirimi yöntemleri için 16 x 16 piksellik blok büyüklüğü, +/- 7 x 7 piksellik de arama alanı kullanılmıştır. Bloğa dayalı hareket kestirimi yöntemleri arasında en başarılı sonucu veren tam arama yönteminde ayrıca 8 x 8 piksellik blok büyüklüğü kullanılarak elde edilen sonuçlar gösterilmiştir.

Çerçevede az yer kaplayan nesnenin öteleme testi için hazırlanan videonun 10. ve 11. çerçevesi Şekil 6.4.a ve b'de görülmektedir.



(a)



(b)



(c)



(d)



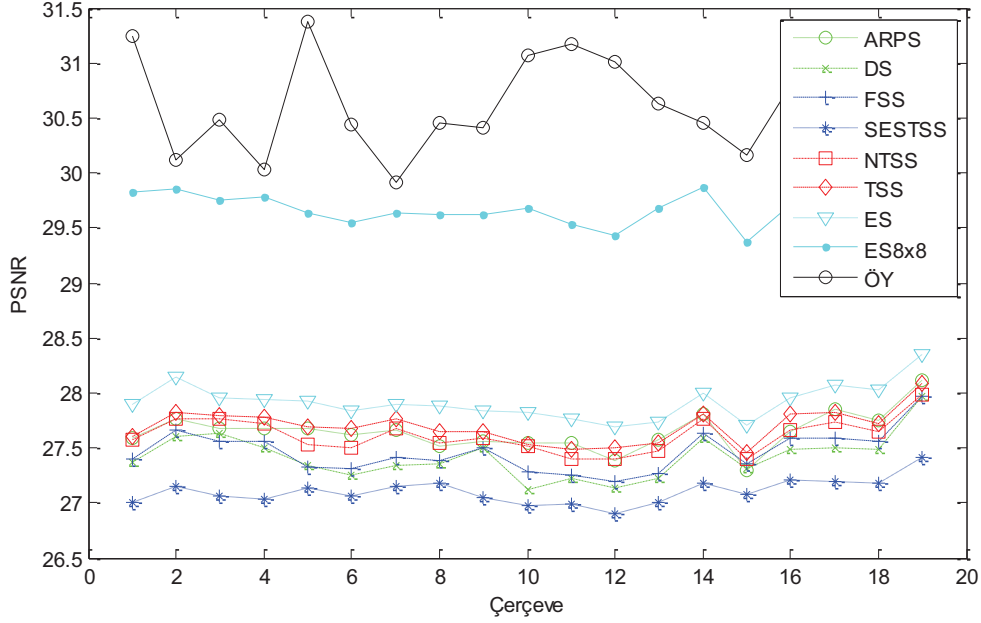
(e)

Şekil 6.4 Çerçeveye çok yer kaplayan nesnenin öteleme testi görüntüleri

a. 10. çerçeve, b. 11. çerçeve, c. 16 x 16 piksellik tam arama yöntemiyle dengelenmiş 11. çerçeve (PSNR=27.8210), d. 8 x 8 piksellik tam arama yöntemiyle dengelenmiş 11. çerçeve (PSNR=29.6773), e. önerilen yöntemle dengelenmiş 11. çerçeve (PSNR=31.0734)

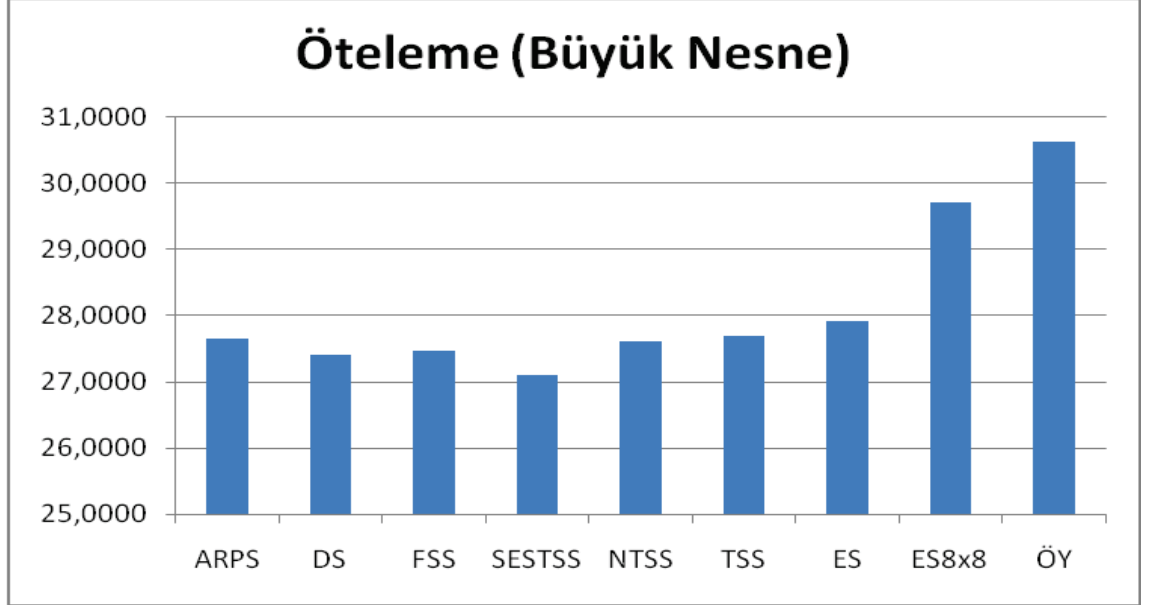
Şekil 6.4.c'de görülen tam arama yöntemiyle gerçekleştirilmiş dengelemenin PSNR'ı 27.8210'dur. Şekil 6.4.d'de görülen 8 x 8 piksellik blok büyüklüğü kullanılarak

gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 29.6773'tür. Şekil 6.4.e'de görülen önerilmiş yöntemle gerçekleştirilen dengeleme sonucu ise PSNR, 31.0734 çıkmıştır. Ardışık 20 çerçeve üzerinde uygulanan yöntemler sonucu elde edilen PSNR değerleri Şekil 6.5'te gösterilmektedir.



Şekil 6.5 Çerçeve de çok yer kaplayan nesnenin öteleme testinde elde edilen PSNR değerleri

Test videosunda arabanın hızı tüm çerçeveler arasında sabittir. Bu sebeple bloğa dayalı hareket kestirimi yöntemlerinin PSNR değerleri, Şekil 6.5'te de görüldüğü gibi, çerçeveler arasında çok fazla değişiklik göstermemektedir. Grafikte önerilen yöntemin diğer yöntemlere göre daha başarılı olduğu görülmektedir. Bu başarımın en büyük nedeni, önerilen yöntemde hareket kestirimi nesnenin hızından bağımsız olmasıdır. Nesnenin çerçeve de iki farklı alan kapladığı öteleme testlerinden elde edilen sonuçlar karşılaştırıldığında nesnenin çerçeve deki büyüklüğü önerilen yöntemin başarımını etkilemediği görülmektedir.



Şekil 6.6 Çerçeveye çok yer kaplayan nesnenin öteleme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri

Şekil 6.6'daki ortalama PSNR değerlerine bakıldığında çerçeveye çok yer kaplayan nesnenin öteleme testinde önerilen yöntem, 16 x 16 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre yaklaşık olarak 3 dB, 8 x 8 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre ise yaklaşık 1 dB daha iyi sonuç vermektedir.

İzdüşüm Testi:

İzdüşüm testini gerçekleştirmek için oyuncak bir otobüsün görüntüsü 640 x 480 piksel boyutlarında alınmıştır. Ardışık 32 çerçeve üzerine, önerilen yöntem ve bloğa dayalı hareket kestirimi yöntemleri uygulanmış ve PSNR sonuçları ile nesnel video kalitesi elde edilmiştir. Bloğa dayalı hareket kestirimi yöntemleri için 16 x 16 piksellik blok büyüklüğü, +/- 7 x 7 piksellik de arama alanı kullanılmıştır. Bloğa dayalı hareket kestirimi yöntemleri arasında en başarılı sonucu veren tam arama yönteminde ayrıca 8 x 8 piksellik blok büyüklüğü kullanılarak elde edilen sonuçlar gösterilmiştir.

İzdüşüm testi için hazırlanan videonun 13. ve 14. çerçevesi Şekil 6.7.a ve b'de görülmektedir.



(a)



(b)



(c)



(d)

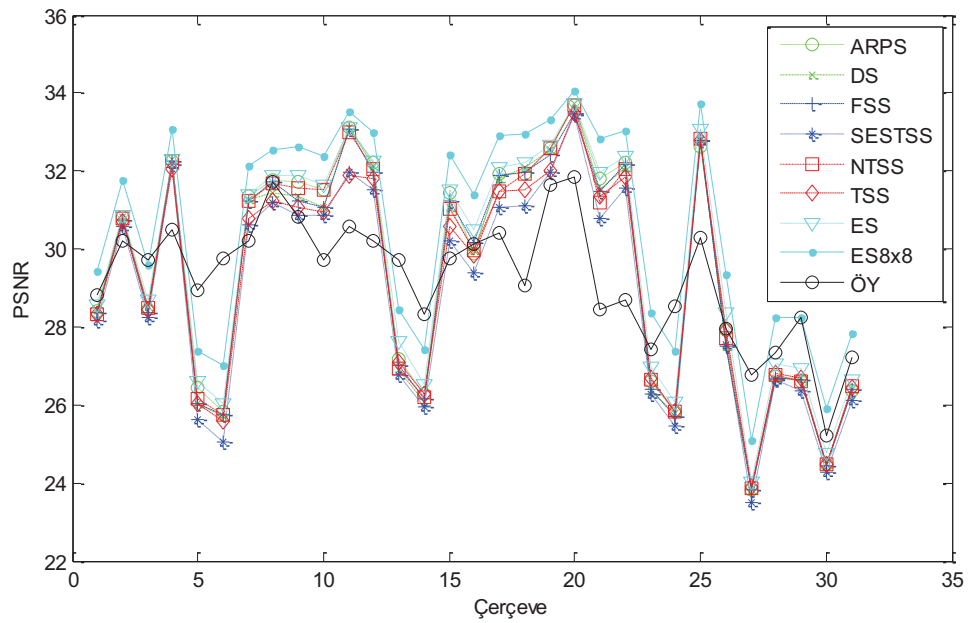


(e)

Şekil 6.7 İzdüşüm testi görüntüleri

a. 13. çerçeve, b. 14. çerçeve, c. 16 x 16 piksellik tam arama yöntemiyle dengelenmiş 14. çerçeve (PSNR=27.6187), d. 8 x 8 piksellik tam arama yöntemiyle dengelenmiş 14. çerçeve (PSNR=28.4514), e. önerilen yöntemle dengelenmiş 14. çerçeve (PSNR=29.7088)

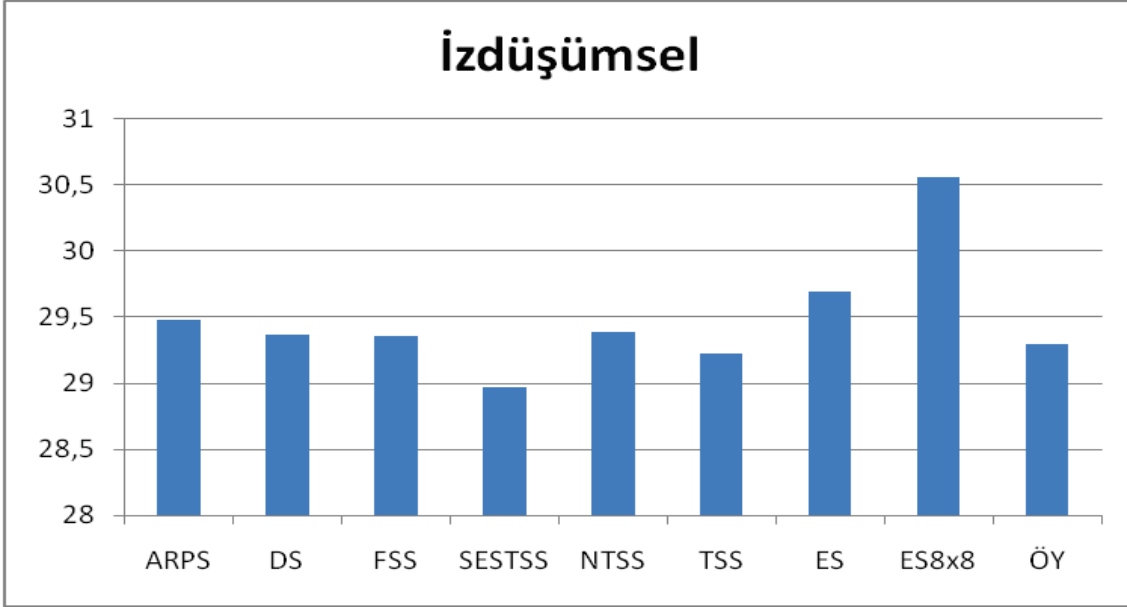
Şekil 6.7.c'de görülen 16 x 16 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 27.6187'tür. Şekil 6.7.d'de görülen 8 x 8 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 28.4514'dir. Şekil 6.7.e'de görülen önerilmiş yöntemle gerçekleştirilen dengeleme sonucu ise PSNR, 29.7088 çıkmıştır. Ardışık 32 çerçeve üzerinde uygulanan yöntemler sonucu elde edilen PSNR değerleri Şekil 6.8'de gösterilmektedir.



Şekil 6.8 İzdüşüm testinde ardışık 32 çerçeve üzerinde elde edilen PSNR değerleri

Şekil 6.8'de de görülen izdüşüm testinde elde edilen sonuçlara bakıldığında önerilen yöntemin bazı çerçevelerde diğerlerinden daha kötü sonuç verdiği, bazı çerçevelerde ise daha iyi sonuç verdiği görülmektedir. Test videosunda aydınlanma, çerçeveden çerçeveye farklılık göstermektedir. Bu sebeple arka plan modelindeki aydınlanma ile bazı çerçevelerdeki aydınlanma birbirinden farklı olmaktadır. Bunun sonucunda da dengelenen çerçeve ile orijinal çerçevenin piksel değerleri arasında farklar oluşmaktadır. Bu durum da önerilen yöntemin başarımını fazlaca etkilemektedir.

Test videosu üzerinde her bir çerçeve için elde edilen PSNR sonuçlarının ortalaması Şekil 6.9’da verilmektedir.



Şekil 6.9 İzdüşüm testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri

İzdüşüm testinde Şekil 6.9’deki ortalama PSNR değerlerine bakıldığında önerilen yöntem diğer yöntemlerle hemen hemen aynı sonuçlar vermiş olmasına karşın 16 x 16 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre yaklaşık olarak 0.4 dB, 8 x 8 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre ise yaklaşık 1 dB daha kötü sonuç vermektedir.

Dönme Testleri:

Yüksek çözünürlükte dönme testini gerçekleştirmek için bir yanlısama görüntüsü 800 x 592 piksel boyutlarda hazırlanmıştır. Yanlısamada, Şekil 6.10’da da görüldüğü gibi görüntü 180° döndürüldüğünde görünen nesne değişmektedir. Birinde sincap görünürken, diğerinde eşek görünmektedir (Jusim 1999). Hazırlanan videoda ardışık 30 çerçeve üzerine, önerilen yöntem ve bloğa dayalı hareket kestirimi yöntemleri uygulanmış ve PSNR sonuçları ile nesnel video kalitesi elde edilmiştir. Bloğa dayalı

hareket kestirimi yöntemleri için 16 x 16 piksellik blok büyüklüğü, +/-7 x 7 piksellik de arama alanı kullanılmıştır. Bloğa dayalı hareket kestirimi yöntemleri arasında en başarılı sonucu veren tam arama yönteminde ayrıca 8 x 8 piksellik blok büyüklüğü kullanılarak elde edilen sonuçlar gösterilmiştir.



(a)



(b)

Şekil 6.10 Yanılsama görüntüsü

(a) Sincap, (b) eşek (180° döndürülmüş)

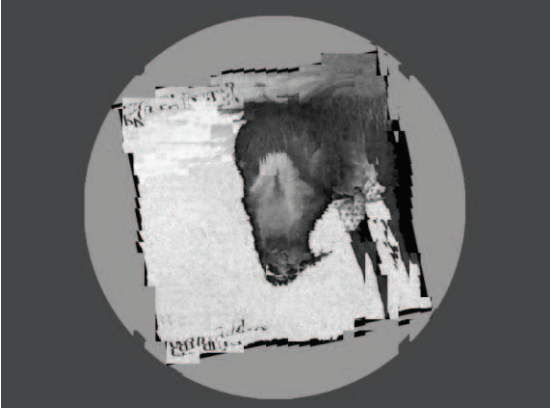
Yüksek çözünürlükte dönme testi için hazırlanan videonun 15. ve 16. çerçevesi Şekil 6.11.a ve b'de görülmektedir.



(a)



(b)



(c)



(d)

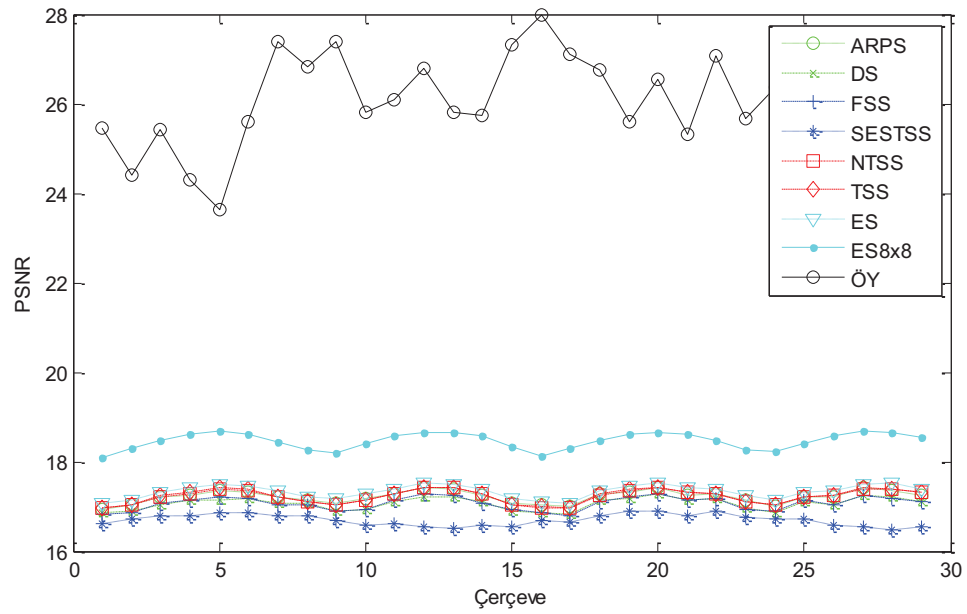


(e)

Şekil 6.11 Yüksek çözünürlükte dönme testi görüntüleri

a. 15. çerçeve, b. 16. çerçeve, c. 16 x 16 piksellik tam arama yöntemiyle dengelenmiş 16. çerçeve (PSNR=17.0930), d. 8 x 8 piksellik tam arama yöntemiyle dengelenmiş 16. çerçeve (PSNR=18.1092), e. önerilen yöntemle dengelenmiş 16. çerçeve (PSNR=27.9926)

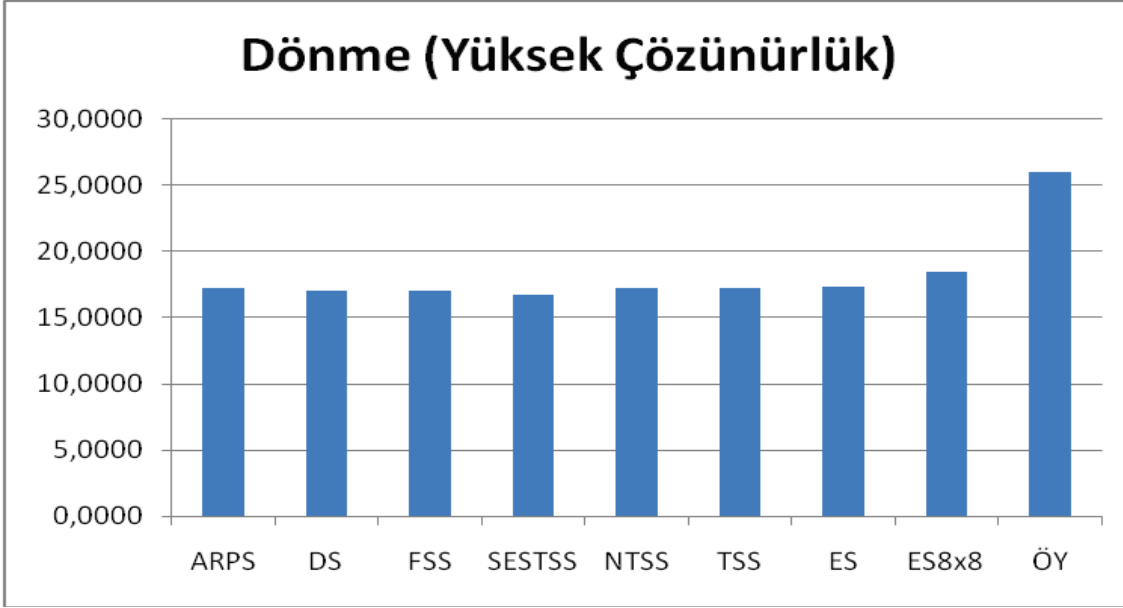
Şekil 6.11.c'de görülen 16 x 16 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 17.0774'tür. Şekil 6.11.d'de görülen 8 x 8 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 18.1092'dir. Şekil 6.11.e'de görülen önerilmiş yöntemle gerçekleştirilen dengeleme sonucu ise PSNR, 27.9926 çıkmıştır. Ardışık 30 çerçeve üzerinde uygulanan yöntemler sonucu elde edilen PSNR değerleri Şekil 6.12'de gösterilmektedir.



Şekil 6.12 Yüksek çözünürlükte dönme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri

Şekil 6.12'de görülen yüksek çözünürlükte dönme testinde elde edilen sonuçlara bakıldığında önerilen yöntem diğer yöntemlere oranla çok daha fazla başarı gösterdiği görülmektedir. Bu başarıya sebep olan en büyük etken önerilen yöntemde nesne hareketi ilgin dönüşümlerle modelleniyor olmasıdır. Öte yandan diğer yöntemlerde uygulanan hareket kestiriminde ise sadece öteleme hareket modeli hesaplanmaktadır.

Test videosu üzerinde her bir çerçeve için elde edilen PSNR sonuçlarının ortalaması Şekil 6.13'te verilmektedir.



Şekil 6.13 Yüksek çözünürlükte dönme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri

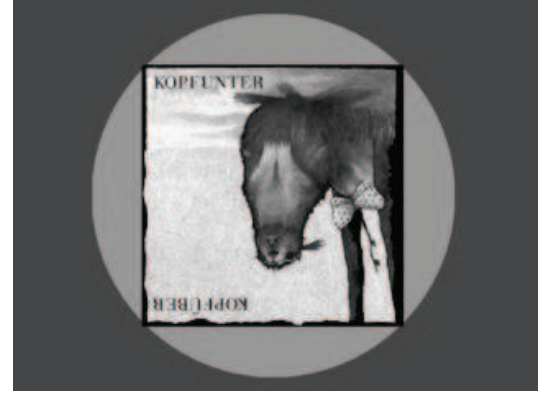
Şekil 6.13'teki ortalama PSNR değerlerine bakıldığında yüksek çözünürlükte dönme testinde önerilen yöntem, 16 x 16 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre yaklaşık olarak 8.5 dB, 8 x 8 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre ise yaklaşık 7.5 dB daha iyi sonuç vermektedir.

Düşük çözünürlükte dönme testini gerçekleştirmek için yüksek çözünürlükte dönme testinde kullanılan 800 x 592 piksel çözünürlükteki yanısama görüntüsü 320 x 240 çözünürlüğe indirilmiştir. Bu test ile çözünürlüğün yöntemler üzerindeki etkisi gösterilmiş ve blok büyüklüklerinin daha iyi görülmesi sağlanmıştır. Hazırlanan videoda ardışık 30 çerçeve üzerine, önerilen yöntem ve bloğa dayalı hareket kestirimi yöntemleri uygulanmış ve PSNR sonuçları ile nesnel video kalitesi elde edilmiştir. Bloğa dayalı hareket kestirimi yöntemleri için 16 x 16 piksellik blok büyüklüğü, +/-7 x 7 piksellik de arama alanı kullanılmıştır. Bloğa dayalı hareket kestirimi yöntemleri arasında en başarılı sonucu veren tam arama yönteminde ayrıca 8 x 8 piksellik blok büyüklüğü kullanılarak elde edilen sonuçlar gösterilmiştir.

Düşük çözünürlükte dönme testi için hazırlanan videonun 15. ve 16. çerçevesi Şekil 6.14.a ve b'de görülmektedir.



(a)



(b)



(c)



(d)

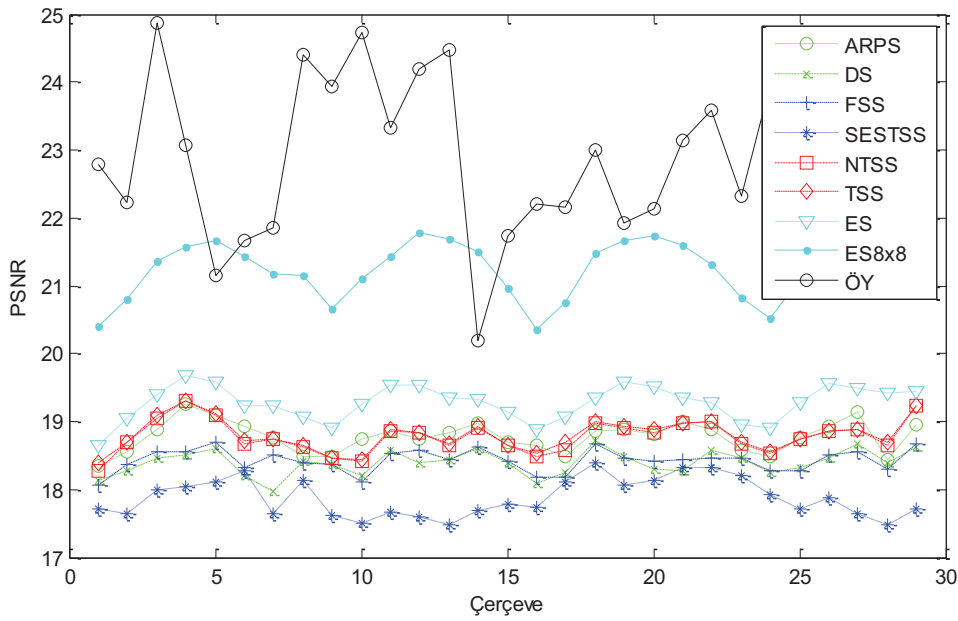


(e)

Şekil 6.14 Düşük çözünürlükte dönme testi görüntüleri

a. 15. çerçeve, b. 16. çerçeve, c. 16 x 16 piksellik tam arama yöntemiyle dengelenmiş 16. çerçeve (PSNR=18.8828), d. 8 x 8 piksellik tam arama yöntemiyle dengelenmiş 16. çerçeve (PSNR=20.3461), e. önerilen yöntemle dengelenmiş 16. çerçeve (PSNR=22.2148)

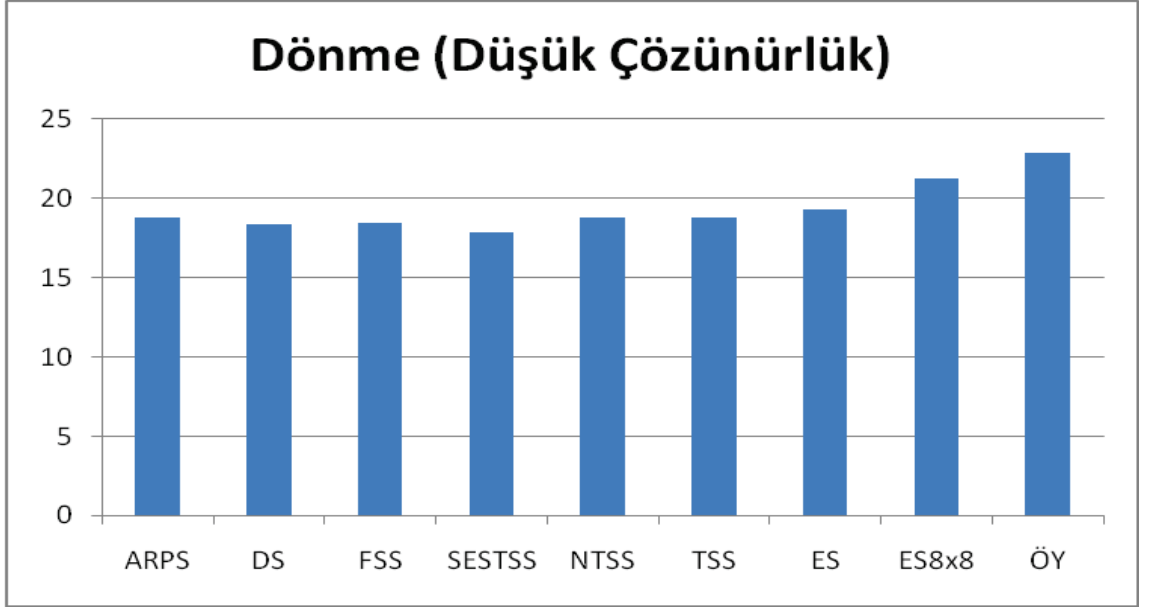
Şekil 6.14.c'de görülen 16 x 16 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 18.8828'dir. Şekil 6.14.d'de görülen 8 x 8 piksellik blok büyüklüğü kullanılarak gerçekleştirilmiş tam arama yöntemiyle elde edilen dengelemenin PSNR'ı 20.3461'dir. Şekil 6.14.e'de görülen önerilmiş yöntemle gerçekleştirilen dengeleme sonucu ise PSNR, 22.2148 çıkmıştır. Ardışık 30 çerçeve üzerinde uygulanan yöntemler sonucu elde edilen PSNR değerleri Şekil 6.12'de gösterilmektedir.



Şekil 6.15 Düşük çözünürlükte dönme testinde ardışık 30 çerçeve üzerinde elde edilen PSNR değerleri

Şekil 6.15'te görülen düşük çözünürlükte dönme testinde elde edilen sonuçlara bakıldığında önerilen yöntem diğer yöntemlere oranla daha başarılı olduğu görülmektedir. Bu başarıya sebep olan en büyük etken önerilen yöntemde nesne hareketi ilgin dönüşümlerle modelleniyor olmasıdır. Öte yandan diğer yöntemlerde uygulanan hareket kestiriminde ise sadece öteleme hareket modeli hesaplanmaktadır. Fakat yüksek çözünürlükte elde edilen sonuçlarla kıyaslandığında düşük çözünürlükte, önerilen yöntemin başarımının olumsuz etkilendiği görülmektedir.

Test videosu üzerinde her bir çerçeve için elde edilen PSNR sonuçlarının ortalaması Şekil 6.16'da verilmektedir.



Şekil 6.16 Düşük çözünürlükte dönme testinde hareket kestirimi yöntemleriyle elde edilen ortalama PSNR değerleri

Şekil 6.16'daki ortalama PSNR değerlerine bakıldığında düşük çözünürlükte dönme testinde önerilen yöntem, 16 x 16 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre yaklaşık olarak 3.5 dB, 8 x 8 piksellik blok büyüklüğü kullanılan tam arama yöntemine göre ise yaklaşık 1.6 dB daha iyi sonuç vermektedir.

7. SONUÇLAR

Tezde, nesne tabanlı bir hareket kestirimi yöntemi önerilmektedir. Önerilen yöntemde hareket kestirimi nesnenin silueti üzerinden gerçekleştirilmektedir ve nesnenin hareketi öteleme hareketinin yanı sıra, ölçekleme, dönme ve izdüşüm hareketi ile modellenmektedir.

Video kodlama standartları, hareketli nesne tespiti yöntemleri ve nesne takibi gibi önerilen yöntemle alakalı konulardan tez içerisinde bahsedilmektedir. Önerilen yöntemin sonuçları bloğa dayalı hareket kestirimi yöntemleriyle karşılaştırılarak yöntemin başarımı ölçülmektedir. Test kapsamında öteleme, izdüşüm ve dönme testi yapılmaktadır. Ayrıca çözünürlüğün ve çerçevede yer alan nesnenin kapladığı alanın yöntemlere etkisi de incelenmektedir.

Testler sonucunda nesne hareketinin fazla olduğu çerçevelerin dengelenmesinde önerilen yöntemin diğer yöntemlere göre çok daha fazla başarılı olduğu görülmektedir. Çözünürlük arttıkça önerilen yöntemin başarımı, bloğa dayalı hareket kestirimi yöntemlerine oranla artmaktadır. Çerçeve içerisinde nesnenin büyük ya da küçük görünüyor olması ise önerilen yöntemin başarımını etkilememektedir.

Öteleme testinde arabanın diğer çerçevelere göre daha yavaş hareket ettiği çerçevelerde yöntemlerin PSNR değerleri birbirine yakın çıkmaktadır. Hareketin daha fazla olduğu diğer çerçevelerde ise önerilen yöntemin başarımı oldukça etkileyicidir. Önerilen yöntemde nesnenin hızından bağımsız olarak hareket kestirimi yapılıyor olması, öteleme testinde yöntemin başarımının yüksek olmasına en büyük etkendir. Nesnenin hareketi ne kadar fazlaysa diğer yöntemlere oranla önerilen yöntemde daha çok başarı elde edilmektedir.

Önerilen yöntemde nesnenin hareketi ilgin hareket modeli ile temsil edilmektedir. Buna karşın diğer yöntemlerde ise hareket, sadece öteleme hareket ile modellenmektedir. Bu sebeple, dönme testinde elde edilen sonuçlarda da görüleceği gibi, nesnenin dönme

hareketi yaptığı çerçevelerde önerilen yöntemin diğer yöntemlere oranla çok üstün bir başarı gösterdiği görülmektedir.

İzdüşüm testinde aydınlanma, çerçeveden çerçeveye farklılık gösterdiği için arka plan modelindeki aydınlanma ile bazı çerçevelerdeki aydınlanma birbirinden farklı olmaktadır. Bunun sonucunda da dengelenen çerçeve ile orijinal çerçevenin piksel değerleri arasında aydınlanma değişikliğinden kaynaklanan farklar oluşmaktadır. Bu durum da önerilen yöntemin başarımını fazlaca düşürmektedir.

Gelecekte ise birden fazla harekete sahip nesnelerin hareketlerinin farklı olduğu bölgelerin daha iyi tespit edilmesi üzerine çalışılabilir. Böylelikle nesne üzerindeki her bir hareket bölgesinin hareket modeli daha doğru çıkartılabilir. Böylece önerilen yöntemin başarımını sınırlayacak etkenler daha da azaltılabilir.

KAYNAKLAR

- Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H. and Rosenberger, C., 2008, "Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms".
- Black, M., and Jepson, A., 1996, "The Robust Estimation of Multiple Motions: Parametric And Piecewise-Smooth Flow Fields", *Comput. Vision Image Understand.* 63, 1, 75-104.
- Bovik, A., 1999, "Handbook of Image and Video Processing", Academic Pres, Inc. Orlando, USA, pp. 95-99.
- Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T., Wedi T. and Ostermann, J., 2004, "Video Coding with H.264/AVC: Tools, Performance, and Complexity", *IEEE Circuits and Systems Magazine*, 4(1):7–28.
- Canny, J., 1986, "A Computational Approach to Edge Detection", *IEEE Trans. Patt. Analy. Mach. Intel.* 8, 6, 679-698.
- Cao, F., Lisani, J. L., Morel, J. M., Muse, P. and Sur, F., 2008, "A Theory of Shape Identification", Springer-Verlag, New York.
- Chiariglione, L., 1995, "The Development of An Integrated Audiovisual Coding Standard: MPEG", *Proc. IEEE*, vol. 83, pp. 151–157.
- Dubrofsky, E., 2009, "Homography Estimation", A Master's Essay, The University of British Columbia.
- Ekinci, M. and Gedikli, E., 2005, "Sihouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance", *Turk J. Elec. Engin.*, Vol. 13, No. 2, Tübitak.
- Elgammal, A. M., Harwood, D. and Davis, L. S., 2000, "Non-Parametric Model for Background Subtraction", *ECCV*, pp. 751-767.
- F'evrier, L., 2007, "A Wide-baseline Matching Library for Zeno", Internship report.
- Friedman, N. and Russell, S., 1997, "Image Segmentation in Video Sequences: a Probabilistic Approach" in *Proc. 13th Conf. Uncertainty in Artificial Intelligence*, pp. 1-3.
- Gall, D. J. L., 1992, "The MPEG video compression algorithm," *Signal Processing: Image Commun.*, vol. 4, no. 4, pp. 129–140.
- Ghanbari, M. , 2003, "Image Compression to Advanced Video Coding", UK, Hertz.

- Grimson, W. E. L. and Stauffer, C., 1999, "Adaptive Background Mixture Models for Real-Time Tracking", CVPR.
- Haralick, R. M. and Shapiro, L. G., 1992, "Computer and Robot Vision, Volume I", Addison-Wesley, pp. 28-48.
- Harris, C. and Stephens, M., 1988, "A combined corner and edge detector", in Proceedings of the Fourth Alvey Vision Conference, Vol. 15, pp.317-325.
- Hartley, R. and Zisserman, A., 2003, "Multiple View Geometry in Computer Vision", Cambridge University Press.
- Hestenes, M. R., 1958, "Inversion of Matrices by Biorthogonalization and Related Results". Journal of the Society for Industrial and Applied Mathematics, 51–90.
- Horn, B. and Schunk, B., 1981, "Determining optical flow", *Artific. Intell.* 17, 185-203.
- Huffman, D. A., 1952, "A Method for the Construction of Minimum-Redundancy Codes", *Proceedings of IRE*, 40(9), 1098-1101.
- Huynh-Thu, Q. and Ghanbari, M., 2008, "Scope of Validity of PSNR in Image/Video Quality Assessment", *Electronics Letters* 44: 800-801.
- ISO/IEC 11172, 1993, Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s.
- ISO/IEC-11172-2, 1991, Video, coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s. ISO/IEC.
- ISO/IEC 13818, 1995, Information technology: generic coding of moving pictures and associated audio information.
- ISO/IEC-13818-2, 1995, MPEG-2 Video Coding Standard, Information Technology - Generic coding of moving pictures and associated audio information: Video. ISO/IEC.
- ISO/IEC FDIS 14496-2 Visual, 1998, ISO/IEC JTC1/ SC29/ WG11 N2502, Vancouver, Canada.
- ISO/IEC 14496-2, 2004, Coding of audio-visual objects – Part 2: Visual.
- ISO/IEC 14496-10 and ITU-T Rec. H.264, 2003, Advanced Video Coding.
- ISO/IEC-14406-10, 2005, Advanced Video coding for generic audiovisual services.
- ITU-T Recommendation H.261, 1993, Video CODEC for audiovisual services at px64 kbit/s.

- ITU-T Recommendation H.263, 1998, Video coding for low bit rate communication, Version 2.
- ITU, 2006, 50 Years of Excellence.
- Jain, R., Kasturi, R. and Schunck, B. G., 1995, "Machine Vision".
- Jusim, J. and Pressler, M., 1999, "Kopfunter Kopfüber", Carl Hanser.
- Kadir, T., Zisserman, A. and Brady, M., 2004, "An affine invariant salient region detector", in Proceedings of the 8th European Conference on Computer Vision, pp. 228–241.
- Koenen, R., 2002, "Overview of the MPEG-4 Standard", ISO/IEC JTC 1/SC 29/WG 11/N4668.
- Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B. and Russel, S., 1994, "Toward Robust Automatic Traffic Scene Analysis in Real-time", in Proc. Int. Conf. Pattern Recognition, Israel, pp. 126-131.
- Kriegman, D. J., 2007, "Computer Vision Lecture Notes", University of California.
- Langer, M., 2009, "Fundamentals of Computer Vision Lecture Notes", McGill, Canada.
- Lee, M. C. et al., 1997, "A Layered Video Object Coding System Using Sprite and Affine Motion Model", IEEE Transaction on Circuits and Systems for Video Technology, Vol. 7, No. 1.
- Li, R., Zeng, B. and Liou, M. L., 1994, "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 4., no. 4, pp. 438-442.
- Lloyd, S. P., 1982, "Least squares quantization in PCM". IEEE Transactions on Information Theory, 129–137.
- Lowe, D., 2004, "Distinctive Image Features from Scale-Invariant Key Points", IJCV, 60(2): 91-110.
- Lu, J. and Liou, M. L., 1997, "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 7, no. 2, pp. 429-433.
- Lucas, B. D. and Kanade., T., 1981, "An iterative image registration technique with an application to stereo vision", International Joint Conference on Artificial Intelligence.

- Massey, M. and Bender, W., 1996, “Salient Stills: Process and Practice”, IBM Systems Journal.
- Matas, J., Chum, O., Urban, M. and Pajdla, T., 2004, “Robust wide-baseline stereo from maximally stable extremal regions”, *Image Vision Comput.*, 22, pp. 761–767.
- McFarlane, N. and Schofield, C., 1995, “Segmentation and Tracking of Piglets in Images”, *MVA*, 8:187-193.
- McIvor, A. M., 2000, “Background Subtraction Techniques”, In Proc. Of Image and Vision Computing, Auckland, New Zealand.
- Meier T. and Ngan, K. N., 1998, “Automatic Segmentation of Moving Objects for Video Object Plane Generation”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5.
- Mikolajczyk, K. and Schmid, C., 2001, “Indexing based on scale invariant interest points”, in *Proceedings of the 8th International Conference on Computer Vision (ICCV)*, pp. 525–531.
- Mikolajczyk, K. and Schmid, C., 2002, “An affine invariant interest point detector”, in *Proceedings of the Seventh European Conference on Computer Vision (ECCV)*, Springer-Verlag, London, pp. 128–142.
- Mikolajczyk, K. and Schmid, C., 2004, “Scale and affine invariant interest point detectors”, *Int. J. Comput. Vis.*, 60, pp. 63–86.
- Mikolajczyk K. and Schmid, C., 2005, “A performance evaluation of local descriptors”, *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1615–1630.
- Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Gool, L., 2005, “A comparison of affine region detectors”, *Int. J. Comput. Vis.*, 65, pp. 43–72.
- Mittal, A. and Paragios, N., 2004, “Motion-Based Background Subtraction using Adaptive Kernel Density Estimation”, *CVPR*, pp. 302-309.
- Muse, P., Sur, F., Cao, F., Gousseau, Y. and Morel, J., 2006, “An a contrario decision method for shape element recognition”, *Int. J. Comput. Vis.*, 69, pp. 295–315.
- Nie, Y. and Ma, K. K., 2002, “Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation”, *IEEE Trans. Image Processing*, vol 11, no. 12, pp. 1442-1448.
- Norriega, P., Bascle, B. and Bernier, O., 2006, “Local Kernel Histograms for Background Subtraction”, *VISAPP*, Volume 1, pp. 213-219.

- Norriega, P. and Bernier, O., 2006, "Real Time Illumination Invariant Background Subtraction Using Local Kernel Histograms", BMVC, Volume 3, pp. 979-988.
- Nunes, P., Correia, P.L. and Pereira, F., 1997, "Coding Video Objects with the Emerging MPEG-4 Standard", Proc Conf. on Telecommunications - ConfTele, Aveiro, Portugal, Vol. I, pp. 425 - 428.
- Panahi, S., Sheikhi, S, Hadadan, S. and Gheissari, N., 2008, "Evaluation of Background Subtraction Methods", IEEE Digital Image Computing: Techniques and Applications.
- Parks, D. H. and Fels, S. S., 2008, "Evaluation of Background Subtraction Algorithms with Post-processing", IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance.
- Phadtare, M., 2007, "Motion Estimation Techniques in Video Processing", Electronic Engineering Times India.
- Piccardi, M., 2004, "Background Subtraction Techniques: A Review", IEEE International Conference on Systems, Man and Cybernetics, pp. 3099-3104.
- Po, L. M., and Ma, W. C., 1996, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 6, no. 3, pp. 313-317.
- Prati, A., Mikic, I., Trivedi, M. and Cucchiara, R., 2003, "Detecting of Moving shadows: Algorithms and Evaluation", IEEE Trans. On Patt. Anal. And Machine Intelligence, vol. 25, pp. 918-923.
- Pratt, W. K., 2007, "Digital Image Processing 4th Edition," John Wiley & Sons, Inc., Los Altos, California.
- Richardson, I. E. G., 2003, "H.264 and MPEG-4 Video Compression ", Wiley.
- Ridler, T. W. and Calvard, S., 1978, "Picture Thresholding Using an Iterative Selection Method.", IEEE Trans. System, Man and Cybernetics, SMC-8, 630-632.
- Rogers, D. F. and Adams, J. A., 1976, "Mathematical Elements for Computer Graphics", McGraw-Hill.
- Salvador, E., Cavallaro, A. and Ebrahimi, T., 2001, "Shadow Identification and Classification Using Invariant Color Model", ICASSP.
- Santa-Cruz, D., Rbahimi, T., Larrson, J. A. M. and Christopoulos, C., 2000, "JPEG2000 Still Image Coding versus Other Standards" Proceedings of the SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII. San Diego, California.

- Schafer, R. and Sikora, T., 1995, "Digital video coding standards and their role in video communications," Proc. IEEE, vol. 83, pp. 907–924.
- Sikora, T., 1997, "The MPEG-4 video standard verification model. Circuits and Systems for Video Technology", IEEE Transactions on, 7, 19-31.
- Smith, M., 2000, "Lossless Realtime Video Compression On The Pentium III", Imperial College Computing Dept., MEng Final Year Project.
- Su, D. and Willis, P., 2004, "Image interpolation by pixel level data-dependent triangulation", Computer Graphics Forum, 23, 189.
- Sullivan, G., J., 2005, "Overview of International Video Coding Standards (preceding H.264/AVC)", ITU-T VICA Workshop.
- Sun, J., Zhang, W., Tang, X. and Shum, H., 2006, "Background Cut", ECCV, pp. 628-641.
- Szeliski, R. and Coughlan, J., 1997, "Spline-based image registration", Int. J. Comput. Vision 16, 1-3, 185-203.
- Turkowski, K., 1990, "Filters for common resampling tasks", Graphics gems, Academic Press Professional.
- Tuytelaars, T. and Gool, L. V., 1999, "Content-based image retrieval based on local affinity invariant regions", in Proceedings of the Third International Conference on Visual Information and Information Systems, pp. 493–500.
- Tuytelaars T. and Gool, L. V., 2000, "Wide baseline stereo matching based on local, affinity invariant regions", in Proceedings of the British Machine Vision Conference, pp. 412–425.
- Yilmaz, A., Li, X. and Shah, M., 2004, "Contour Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras", IEEE Trans. Patt. Analy. Mach. Intel. 26, 11, 1531-1536.
- Yilmaz, A., Javed, O. and Shah, M., 2006, "Object Tracking: A survey", ACM Comput. Surv. 38, Article 13.
- Yu, Y. and Doermann, D., 2005, "Model of Object-Based Coding for Surveillance Video", ICASSP.
- Yu, G. and Morel, J.-M., 2009, "A Fully Affine Invariant Comparison Method", ICASSP.
- Zhu S., and Ma, K. K., 2000, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 9, no. 2, pp. 287-290.

ÖZGEÇMİŞ

Adı Soyadı : SEYİT TUNÇ

Doğum Yeri : ANKARA

Doğum Tarihi : 26/11/1984

Medeni Hali : Bekar

Yabancı Dili : İngilizce, Almanca

Eğitim Durumu

Lise : Gazi Anadolu Lisesi (1998 – 2002)

Lisans : Ankara Üniversitesi, Elektronik Mühendisliği (2002 - 2006)

Yüksek Lisans: Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Mühendisliği Anabilim Dalı (Eylül 2007 – Ağustos 2010)

Çalıştığı Kurumlar ve Yıl

Siemens A.Ş. (2005)

Man Türkiye A.Ş. (2005)

Kuartet Otomasyon Ltd. Şti. (2007 - 2009)

Roketsan A.Ş. (2010 - ...)

Yayınlar

Sünnetçioğlu, Ö. F., Tunç, S., Oğur, U., 2007, “İnsansız Model Araç”, SIU, Eskişehir.