

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**2-BOYUTLU HAREKET KESTİRİMİ ve VIDEO
ÇERÇEVE ARADEĞERLEMESİNDE UYGULANMASI**

Metalurji Müh. Selim NASIR

**FBE Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Songül ALBAYRAK (YTÜ)

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT.....	xi
1. GİRİŞ	1
1.1 Hareket Kestirimi ile İlgili Önceki Çalışmalar	2
1.2 Video Çerçeve Aradeğerlemesi ile İlgili Önceki Çalışmalar.....	4
2. VIDEO	6
2.1 Giriş.....	6
2.2 Sayısal Video	6
2.3 Çerçeve Kavramı.....	7
2.4 Videonun Renk Özellikleri	8
2.4.1 Renk Kavramı	9
2.4.2 Video Tekniğinde Kullanılan Renk Modelleri.....	10
2.4.2.1 RGB Renk Uzayı	11
2.4.2.2 YCbCr Uzayı	12
2.4.2.3 YUV Uzayı	13
2.5 Video Alımı	13
2.6 Video Gösterimi.....	14
2.7 Video İletimi	15
2.8 Renk Altörneklemesi	15
2.9 Video Çerçeve Formatları.....	18
3. 2-BOYUTLU HAREKET KESTİRİMİ	19
3.1 Giriş.....	19
3.2 Hareket Kestiriminin Video Sıkıştırmasında Uygulanması.....	20
3.2.1 Video Sıkıştırması.....	21
3.2.2 Hareket Dengeleme (Motion Compensation)	23
3.3 Seyrek Hareket Vektör Kümesi	25
3.4 Piksel Özyinelemeli Yöntemler	26
3.5 Blok Eşleştirme Algoritmaları (BEA)	27
3.5.1 Hareket Kestiriminde Kullanılan Eşleştirme Ölçütleri	29

3.6	Tam Arama (TA)	31
3.7	Hızlı BEA.....	33
3.7.1	Üç Adımda Arama (ÜAA).....	34
3.7.2	Yeni Üç Adımda Arama (YÜAA)	35
3.7.3	Dört Adımda Arama (DAA)	38
3.7.4	Eşkenar Arama (EA).....	40
3.8	Hareket Kestirimi Algoritmalarının Karşılaştırılması.....	43
3.8.1	Hareket Kestirimi Yöntemlerinden Elde Edilen Ortalama PSNR Başarımı.....	46
3.8.2	Karşılaştırma Sonuçlarının Değerlendirilmesi.....	47
4.	HAREKET KESTİRİMİ YÖNTEMLERİNİN GELİŞTİRİLMESİ.....	49
4.1	Hiyerarşik Arama.....	49
4.2	Piksel-altı (Sub-pixel) Hareket Kestirimi	51
4.3	Tezde Önerilen Yöntem	53
4.3.1	Hareket Değişiminin Belirlenmesi.....	54
4.3.2	Hareket Değişimi Matrisinin Hesaplanmasında Kullanılan Eşik Değeri.....	55
4.3.3	Çalışılan Çerçeveadaki Blokların Sınıflandırılması	57
4.3.4	Sınıflandırılan Bloklara Hareket Kestiriminin Uygulanması.....	58
4.3.5	Önerilen Yöntemden Elde Edilen Sonuçlar	58
5.	VİDEO ÇERÇEVE ARADEĞERLEMESİ.....	62
5.1	Hareket Dengelemesiz Çerçeve Aradeğerlemesi.....	63
5.1.1	Çerçeve Yineleme	64
5.1.2	Çerçeve Ortalama.....	64
5.2	Hareket Dengelemeli Çerçeve Aradeğerleme (HDÇA).....	65
5.2.1	HDÇA'nın Uygulanması	66
5.2.2	HDÇA'sinden Elde Edilen Sonuçlar	68
6.	SONUÇ ve ÖNERİLER	71
KAYNAKLAR		74
ÖZGEÇMİŞ		76

SİMGE LİSTESİ

Λ	Çerçeve örnekleme kafesi
A	Üç adımda aramada kullanılan adım boyu
C	Çalışılan çerçeve
p	Arama parametresi
R	Referans çerçeve
T_{BS}	Blok sınıflandırılmasında kullanılan eşik değeri
$T_{FÇ}$	Fark çerçevesinden hareket değişimi matrisinin eldesinde kullanılan eşik değeri

KISALTMA LİSTESİ

BEA	Blok Eşleştirme Algoritmaları
BEAÖ	Büyük Eşkenar Arama Örüntüsü
CCD	Charge-coupled device
CIF	Common Intermediate Format
CMYK	Cyan Magenta Yellow Black
CODEC	COder/DECoder (Video Kodlayıcı-kodçözücüsü)
CRT	Cathode Ray Tube
DAA	Dört Adımda Arama
DVD	Digital Versatile Disc
EA	Eşkenar Arama
HDÇA	Hareket Dengelemeli Çerçeve Aradeğerleme
HSI	Hue Saturation Intensity
KEAÖ	Küçük Eşkenar Arama Örüntüsü
LCD	Liquid Crystal Display
MFT	Mutlak Farklar Toplamı
MHT	Mutlak Hatalar Toplamı
MPEG	Moving Picture Experts Group
NTSC	National Television Standards Committee
OKH	Ortalama Karesel Hata
OMF	Ortalama Mutlak Fark
PAL	Phase Alternating Line
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
RGB	Red Green Blue
SECAM	Séquentiel Couleur à Mémoire
SQCIF	Sub-Quarter Common Intermediate Format
TA	Tam Arama
ÜAA	Üç Adımda Arama
YCbCr	Işıklılık (Y) ve iki renk farkı sinyallerini (Cb ve Cr) içeren renk koordinatı
YÜAA	Yeni Üç Adımda Arama
YUV	Işıklılık (Y) ve iki renk bileşeni (U ve V) içeren renk koordinatı

ŞEKİL LİSTESİ

	Sayfa
Şekil 1.1 Bir hareket kestirimi yönteminde esinlenilen doğadaki örümcek ağı örüntüsü	4
Şekil 2.1 Video oluşturan durağan görüntü dizisi.....	7
Şekil 2.2 Video çerçevesinde tarama çizgileri.....	7
Şekil 2.3 Dalga boyuna bağlı olarak insan gözündeki kırmızı, yeşil ve mavi koni hücrelerinin ışığı soğurması	10
Şekil 2.4 Renkli katot ışını tüpünün kesit resmi	14
Şekil 2.5 Ardışıl ve geçmeli satır satır tarama formatları	15
Şekil 2.6 4:2:0, 4:2:2 ve 4:4:4 örnekleme örüntüleri	17
Şekil 2.7 Çeşitli çözünürlüklerde örneklenen video çerçeveleri.....	18
Şekil 3.1 Bir görüntü dizisindeki konumsal ve zamansal korelasyon	20
Şekil 3.2 Hem tekyönlü hem de çiftyönlü zamansal kestirim kullanılarak video kodlaması.....	21
Şekil 3.3 CODEC.....	22
Şekil 3.4 Video kodlayıcısının blok diyagramı	22
Şekil 3.5 Hareket dengelemesi yapılmadan elde edilen fark çerçevesinde kalan yüksek enerji.....	24
Şekil 3.6 Hareket kestirimi ve dengelemesinden sonra elde edilen fark çerçevesi	24
Şekil 3.7 Hareket kestirimi ve dengelemesi blok diyagramı	25
Şekil 3.8 Seyrek noktalarda hareket vektörünün bulunması.....	26
Şekil 3.9 Blok eşleştirme örneği	28
Şekil 3.10 Blok eşleştirme örneğindeki OKH yüzey grafiği	29
Şekil 3.11 Referans çerçevede arama alanının oluşturulması.....	30
Şekil 3.12 Arama alanında satır düzeninde arama.....	32
Şekil 3.13 Arama sonucunda atanacak hareket vektörleri	32
Şekil 3.14 OMF değerlerinin oluşturduğu hata yüzeyi.....	33
Şekil 3.15 Üç adımda arama	34
Şekil 3.16 Yeni üç adımda arama	36
Şekil 3.17 Dört adımda arama	38
Şekil 3.18 DAA'da kullanılan arama örüntüleri.....	39
Şekil 3.19 Uygun bir arama örüntü desteği	40
Şekil 3.20 EA'da kullanılan arama örüntüleri.....	41
Şekil 3.21 Eşkenar arama.....	41
Şekil 3.22 EA'da kısmen örtüşen denetleme noktaları.....	42

Şekil 3.23 Hareket kestirimi yöntemlerinin karşılaştırılmasında kullanılan video görüntüleri.	43
Şekil 3.24 Hareket kestirimi yöntemlerinin karşılaştırılmasında kullanılan yöntem.....	44
Şekil 3.25 Foreman videosundan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinden elde edilen OKH başarımları	45
Şekil 3.26 Foreman videosundan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinde, bir çerçevedeki ortalama blok karşılaştırma sayıları	45
Şekil 4.1 Hiyerarşik blok eşleştirme algoritması	50
Şekil 4.2 Yarım-piksel aradeğerlemesi	52
Şekil 4.3 Çeyrek-piksel hareket kestirimi	53
Şekil 4.4 Tezde önerilen yöntemin blok diyagramı	54
Şekil 4.5 Fark sinyalinin ilgili pikselin 3x3 komşuluğundaki piksellerin OMF'ı ile hesaplandığı fark çerçevesi	55
Şekil 4.6 Video çerçevelerindeki hareket içeriğine göre fark çerçevesinin ve eşikleme değerinin değişimi	56
Şekil 4.7 Alt blokların hareket kestiriminin TA ile yapıldığı önerilen yöntemin Foreman videosu için TA ve YÜAA ile OKH karşılaştırması	59
Şekil 4.8 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin Foreman videosu için TA ve YÜAA ile OKH karşılaştırması	60
Şekil 5.1 Çerçeve aradeğerlemesi	62
Şekil 5.2 Çerçeve aredeğerleme yöntemleri	63
Şekil 5.3 Çerçeve yinleme	63
Şekil 5.4 Çerçeve ortalaması ile aradeğerlenen çerçevenin piksellerinin belirlenmesi	64
Şekil 5.5 Çerçeve ortalaması ile aradeğerlemede, çerçevelerdeki hareket durumuna göre niteliğin değişmesi	64
Şekil 5.6 Hareket dengelemeli çerçeve aradeğerleme	65
Şekil 5.7 Çerçeve aradeğerlemesinde ileri hareket kestirimi	66
Şekil 5.8 Aradeğerlenen çerçevede hareket vektörünün yarısı kadar ötelenen bloğun blok kafesinde kapladığı bloklar	67
Şekil 5.9 Çerçeve aradeğerlemesinde geri hareket kestirimi	67
Şekil 5.10 HDÇA yöntemlerinin testinde kullanılan yöntem	69
Şekil 5.11 Foreman videosu kullanılarak yapılan aradeğerlemede yöntemlerin OKH başarımları	69

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 Video çerçeve formatları	17
Çizelge 3.1 Blok eşleştirme örneğindeki OKH değerleri	28
Çizelge 3.2 Hareket kestirimi yöntemlerinin çerçeve başına OKH başarımları	46
Çizelge 3.3 Hareket kestirimi yöntemlerinin ortalama PSNR başarımları	47
Çizelge 4.1 Farklı T_{BS} eşik değerlerine göre PSNR başarımları	57
Çizelge 4.2 Farklı T_{BS} eşik değerlerine göre ortalama hareket kestirimi süresi (sn).....	58
Çizelge 4.3 Alt blokların hareket kestiriminin TA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının PSNR başarımları	59
Çizelge 4.4 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının PSNR başarımları	60
Çizelge 4.5 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının ortalama hareket kestirimi süreleri	60
Çizelge 5.1 Çerçeve aradeğerleme yöntemlerinin PSNR başarımları	70
Çizelge 5.2 HDÇA'de tekyönlü hareket kestirimi sonucu PSNR başarımlarının düşmesi	70

ÖNSÖZ

Metalurji Mühendisliği'nden bambaşka bir mesleği yüksek lisans eğitimim boyunca bana kazandıran Bilgisayar Mühendisliği'ndeki tüm hocalarıma ve tez çalışmamda bilgi ve görüşleriyle beni her zaman destekleyen danışmanım Sayın Yrd. Doç. Dr. Songül ALBAYRAK'a teşekkürlerimi sunarım.

Yüksek lisans sırasında birlikte çalışmalar yaptığımız arkadaşlarım Erkan USLU ve İlktan AR'a teşekkür ederim. Sevgili annem, babam ve aileme yalnız bu tez çalışmasında değil, hayatım boyunca gösterdikleri sabır ve destek için teşekkürü bir borç bilirim.

ÖZET

Sayısal video işleme, bilgisayarların işlem gücünün artmasına bağlı olarak çoğulortam uygulamalarının yaygınlaşması ile önem kazanmıştır. Videolar doğal olarak yüksek miktarda veri içermektedir. Bu nedenle videoların verimli bir biçimde saklanması ve aktarımı için pek çok yöntem geliştirilmiştir. Video işleme algoritmalarının amacı videonun en uygun görsel nitelikle gösterilmesidir.

Bantgenişliğinin ya da depolama kapasitesinin kısıtlı olduğu durumlarda zamansal örnekleme yapılarak videonun veri miktarının azaltılması amaçlanır. Zamansal örnekleme sonucunda videonun çerçeve hızının düşürülmesi ile bozulan görsel niteliğin arttırılması için video çerçeve aradeğerleme yöntemleri geliştirilmiştir. Çerçeve aradeğerlemesi için her bir video çerçevesindeki sahnenin değişiminden kaynaklanan hareket olgusundan yararlanılır. 2-boyutlu hareket kestiriminde nesnelerin öteleme hareketi yaptıkları varsayılır ve bu nesnelerin yeni konumları aradeğerlenen çerçevede yaklaşık olarak belirlenir. Çerçeve aradeğerlemesinde, hareket kestirimi sonucunda elde edilen hareket vektörlerinin nesnelerin gerçek hareketlerini göstermesi istenmektedir.

Bu çalışmada, hareket vektörlerinin yüksek doğrulukla hesaplanması problemi ele alınmıştır. Problemin çözümü için önerilen yöntemde hareket kestirimi bilinen yöntemlere göre daha iyi bir başarımla yapılmıştır. Hareket kestiriminin uygulaması olarak video çerçeve aradeğerlemesi seçilmiştir.

Anahtar Kelimeler : Sayısal video işleme, 2-boyutlu hareket kestirimi, hareket dengeleme, blok eşleştirme algoritmaları, video çerçeve aradeğerlemesi.

ABSTRACT

Digital video processing has become an important topic with widespread usage of multimedia applications due to the increasing processing power of computers. A video contains a considerable amount of data. Many methods thus have been developed to store and transmit videos efficiently. The ultimate goal of any video processing algorithm is to display the video with optimum visual quality.

In some cases where bandwidth or storage capacity is limited, temporal sampling can be utilized to reduce the amount of video data. Video frame interpolation schemes have been developed to compensate for distorted visual quality due to the reduction in video frame rate by temporal sampling. Motion phenomenon which is a result of scene change in each video frame is exploited for video frame interpolation. In 2D motion estimation, it is assumed that objects in the scene undergo translational movement and new positions of these objects are determined approximately within the interpolated frame. It is required that motion vectors obtained from motion estimation algorithm for video frame interpolation should point to true object motion.

In this work, the problem of calculation of accurate motion vectors has been dealt with. Known motion estimation methods are outperformed by the proposed method for the solution of the problem. Video frame interpolation has been chosen as an application of motion estimation.

Keywords: Digital video processing, 2D motion estimation, motion compensation, block matching algorithms, video frame interpolation.

1. GİRİŞ

Video çerçeve aradeğerlemesi (video frame interpolation), video-konferans uygulamalarında bantgenişliğinin sınırlı olması ya da gözetim kameralarının daha az görüntü verisiyle kayıt yapma zorunluluğu gibi nedenlerle, video **çerçeve hızının** (frame rate) düşürülmesiyle bozulan görsel niteliğin çerçeve hızının artırılması ile iyileştirilmesini amaçlar. Çerçeve aradeğerlemesi, ardışıl iki video çerçevesinin arasına bir ya da daha fazla sayıda **yeniden oluşturulan çerçevenin** (reconstructed frame) eklenmesiyle gerçekleştirilir.

Video görüntüleri bir sahnenin konumsal ve zamansal örneklenmesiyle elde edilir. Videoyu oluşturan çerçevelerde sahnenin değişimi hareket olgusunu meydana getirir. Çerçeve aradeğerlemesi hareket olgusu göz ardı edilerek **hareket dengelemesiz** ya da hareket bilgisi kullanılarak **hareket dengelemeli** (motion compensated) gerçekleştirilebilir. Hareket dengelemesiz çerçeve aradeğerlemesinde videodaki hareket **kesintili** (jerky) algılanacaktır ya da **bulanık** (blurry) görüntülü çerçeveler elde edilecektir. Özellikle **çok hareketli** (high motion) videolarda sözü geçen etkiler çok daha hissedilir olmaktadır. Hareket dengelemesiz yöntemler her ne kadar yalın ve kolay uygulanabilir olsalar da, başarımları kabul edilebilir değildir.

Video çerçeve aradeğerlemesinin niteliğini arttırabilmek için hareket dengelemeli algoritmalar düşünülmüştür. Hareket dengelemeli yöntemlerde, videonun düşük çerçeve hızı ile saklanması ya da iletilmesinden sonra gösterimi sırasında, videodan atılan çerçeveler başarıyla yeniden oluşturulmaktadır. Bir görüntü dizisindeki çerçeveler arasındaki hareketleri algılayabilmek için **hareket kestirimi** (motion estimation) yöntemleri kullanılmaktadır.

Hareket kestirimi yöntemleri yalnızca video çerçeve aradeğerlemesinde değil, **video sıkıştırması** (video compression) için de önemlidir. Doğal videolarda ardışıl video çerçeveleri birbirlerine oldukça benzerdir. Video sıkıştırmasında benzer çerçeveler birbirlerinden bağımsız bir biçimde ya da bir çerçeve kendisinden önceki ve/veya sonraki çerçevelerdeki farklılıkların belirlenmesi ve yalnızca bu farklılıkların kodlanması ile sıkıştırılabilir. İkinci yöntem diğerine göre çok daha etkin bir video sıkıştırması sağlar. Hareket dengelemeli video sıkıştırmasında iki türlü bilgi söz konusudur: **Çalışılan çerçeve** (current frame) ve sıkıştırılmasında kullanılan **referans çerçeveden** (reference frame) elde edilen **fark çerçevesi** (displaced frame difference, DFD) ve referans çerçeveye yapılan hareket kestirimi sonucu elde edilen **hareket vektörleri** (motion vector). Video sıkıştırmasında her iki bilgi de kodlanmaktadır. Hareket kestiriminin doğru bir biçimde yapılması fark çerçevesinin daha az

bitle kodlanmasını sağlayacaktır, dolayısıyla sıkıştırma verimi artacaktır. Sıkıştırılmış bir videonun kodçözümlü yapılırken bir çerçeve, kodlanmış fark çerçevesi ve hareket vektörleri kullanılarak yeniden oluşturulabilir.

Video sıkıştırmasında hareket kestiriminin yanlış yapılması, fark çerçevesinin daha fazla bitle kodlanması, dolayısıyla sıkıştırma veriminin azalması gibi bir yan etkiye sahiptir. Hareket vektörü yanlış bir yönü gösteriyor olsa da (daha fazla bitle kodlanmış) fark çerçevesindeki bilgi yanlış hareket kestirimini karşılayacaktır. Ne var ki video çerçeve aradeğerlemesinde bir fark çerçevesi söz konusu değildir. **Özgün videodan** (original video) **atılan çerçevelerin** (skipped frames) yeniden oluşturulması için tek yapılabilecek şey eldeki çerçeveler arasında hareket kestirimi ile hareket vektörlerinin belirlenmesidir. Ancak bu durumda yanlış hareket vektörlerini dengeleyecek herhangi bir bilgi yoktur. Öyleyse video çerçeve aradeğerlemesi için yapılan hareket kestiriminde elde edilen hareket vektörlerinin nesnelerin gerçek hareketlerini belirtmeleri gerekmektedir.

Bu tez çalışmasında gerek video çerçeve aradeğerlemesinin gerekse de video sıkıştırmasının başlıca ögesi olan hareket kestirimi ve video çerçeve aradeğerlemesindeki uygulaması ele alınmıştır. Hareket kestirimi, video sıkıştırmasında fark çerçevesinin kodlanacağı bit sayısının azaltılmasında bir yöntem olarak kullanıldığı için, çerçeve aradeğerlemesinde gerçek hareket kestiriminin yapılmasının bir problem olarak ele alınması tercih edilmiştir.

1.1 Hareket Kestirimi ile İlgili Önceki Çalışmalar

Hareket kestirimi bir çerçeveyi oluşturan tüm pikseller için bir hareket vektörü belirlenmesi ile yapılabilir. Ancak bu işlem oldukça maliyetlidir. Bu nedenle **blok eşleştirme algoritmaları**, **BEA** (block matching algorithms, BMA) önerilmiştir. Bu yöntemlerde genel olarak çalışılan çerçeve birbiriyle örtüşmeyen, uygun boyutlarda bloklara ayrılır ve her bir blok referans çerçeve olarak kullanılan bir önceki ya da bir sonraki çerçevede oluşturulan bir **arama penceresinde** (search window) belirli bir karşılaştırma ölçütüne göre araştırılır. Bu yöntemlerde bir blok içerisindeki tüm piksellerin aynı 2-boyutlu öteleme hareketini yaptıkları varsayılır. Arama penceresinin merkezi çalışılan çerçevedeki blokla aynı konumdaki referans çerçevedeki bloktur. Çalışılan çerçevedeki blok ile referans çerçevedeki arama penceresinde **en iyi eşleşen blok** (best matching block) bulunur ve iki blok arasındaki konumsal fark hareket vektörünü verir. En iyi eşleşen referans blok, kullanılan karşılaştırma ölçütüne göre çalışılan çerçevedeki blokla en az hatayı veren bloktur.

En yalın hareket kestirimi algoritması olan **tam aramada, TA** (full search, FS) arama penceresindeki tüm aday bloklar çalışılan blok ile karşılaştırılır. Ancak bu işlemin zaman maliyeti oldukça fazladır. Video uygulamalarının yaygınlaşmasına koşut olarak TA'nın zaman maliyetini azaltmak için pek çok hızlı hareket kestirimi algoritması önerilmiştir. Tüm hızlı arama algoritmalarında genel olarak belirli bir **arama örüntüsüne** (search pattern) göre arama penceresindeki aday blok sayısının azaltılması amaçlanmaktadır.

Koga vd. (1981) tarafından, belirlenen arama örüntüsüne göre en iyi eşleşen bloğu üç adımda sonlanan bir yöntemle bulan **üç adımda arama, ÜAA** (three step search, TSS) algoritması geliştirilmiştir.

Hızlı hareket kestirimi algoritmalarında arama penceresindeki pek çok blok göz ardı edildiğinden, herhangi bir hızlı algortmada global minimum hatayı veren blok yerine, yerel minimum hatayı veren bir bloğun seçilmesi olanaklıdır. TA'da ise arama penceresindeki tüm aday bloklar dikkate alındığından global minimumun bulunması güvence altındadır. Bu nedenle hızlı arama algoritmalarının başarımları her zaman TA'dan düşüktür.

ÜAA'da kullanılan arama örüntüsünün yerel minimum hatayı veren bir bloğu seçmesi olasılığı yüksek olduğundan, doğal videolardaki hareket özelliklerini dikkate alan hızlı hareket algoritmaları geliştirilmiştir. Doğal videolardaki pek çok bloğun hareket vektörü arama penceresinin merkezi civarındaki referans blokları göstermektedir. Bu nedenle **merkez-yanlı** (center-biased) arama örüntülerinin kullanıldığı, Reoxiang vd. (1994) tarafından **yeni üç adımda arama, YÜAA** (new three step search, NTSS), Po ve Ma (1996) tarafından **dört adımda arama, DAA** (four step search, 4SS), Zhu ve Ma (2000) tarafından **eşkenar arama, EA** (diamond search, DS) algoritmaları önerilmiştir. YÜAA, DAA ve EA yöntemlerine göre biraz daha iyi blok eşleştirme başarısına sahiptir. EA algoritması blok eşleştirme ve zaman karmaşıklığı bakımından iyi bir başarımla ortaya koymaktadır.

Cheung ve Po (2002), **yeni çapraz-eşkenar arama algoritmasında** (a novel cross-diamond search algorithm) doğal videolardaki çapraz-merkez-yanlı hareket vektörü dağılımından yararlanarak EA'yı geliştirmiş ve EA'ya göre %40'a kadar hız kazancı sağlamıştır.

Chen ve Chen (2004) tarafından örümcek ağı arama örüntüsünün kullanıldığı bir hareket kestirimi yöntemi önerilmiştir. Doğadaki örümcek ağı örüntüsünde (Şekil 1.1) içteki dairedaki ipliklerin dıştaki dairedekilerden daha yoğun olmasından yola çıkılmıştır. Örümcek ağının merkezde-toplanan özelliğinden esinlenilerek merkez-yanlı bir arama örüntüsü oluşturulmuştur.



Şekil 1.1 Bir hareket kestirimi yönteminde esinlenilen doğadaki örümcek ağı örüntüsü

Hiyerarşik blok eşleştirme (hierarchical block matching) yöntemlerinde referans ve çalışılan çerçevelerin farklı çözünürlüklerde temsil edildikleri bir çözünürlük piramidi oluşturulur. En düşük çözünürlük seviyesinde hareket kestirimi yapılarak ‘kaba’ hareket vektörleri elde edilir. Hesaplanan hareket vektörleri bir sonraki yüksek çözünürlük seviyesinde **başlangıç tahmini** (initial estimate) olarak kullanılmaktadır.

Hareket kestirimi yöntemlerinin pek çoğunda nesnelerin tamsayılarla ötelendikleri varsayılır. Ancak hareketli nesnenin gerçek öteleme miktarı kesirli bir sayı olabilir. **Piksel-altı** (sub-pixel) hareket kestiriminde, arama penceresindeki kesirli konumlardaki pikseller tamsayı konumlardaki piksellerin **konumsal aradeğerlemesi** (spatial interpolation) ile elde edilir. Zhang vd. (2004) tarafından **çeyrek-piksel** (quarter-pixel) hareket kestirimine dayalı yeni bir yöntem geliştirilmiştir.

Video çerçevelerindeki sahnenin değişimi ya da bir nesnenin sahnedeki başka bir nesne tarafından kapatılması (occlusion) gibi problemleri çözmek üzere **çiftyönlü hareket kestirimi** (bidirectional motion estimation) yöntemleri geliştirilmiştir. Kalra ve Chong (1998) tarafından **vektör yayılımı yoluyla çiftyönlü hareket kestirimi** (bidirectional motion estimation via vector propagation) algoritması önerilmiştir.

1.2 Video Çerçeve Aradeğerlemesi ile İlgili Önceki Çalışmalar

Çerçeve aradeğerlemesinin hareket dengelemesiz yapıldığı yöntemlerden **çerçeve yinlemede** (frame repetition) elde edilen her bir çerçeve, bir sonraki elde edilen çerçeveye kadar istenilen sayıda yinelenir, **çerçeve ortalama** (frame averaging) ise ardışıl iki çerçevenin karşılıklı piksellerinin aritmetik ortalamaları alınarak **aradeğerlenen çerçeve** (interpolated frame) oluşturulur.

Hareket dengelemeli aradeğerleme yöntemlerinde genel olarak, hareket kestirimi ile nesnelerin **gerçek hareket vektörlerinin** (true motion vectors) elde edilmesi amaçlanmaktadır.

Wong ve Au (1995) ile Choi vd. (2000) tarafından çiftyönlü hareket kestirimine dayanan çerçeve aradeğerleme algoritmaları önerilmiştir. Ardışıl iki çerçeveden zamansal sırada önde olan önceki çerçeve ile sonraki çerçeve arasında **ileri hareket kestirimi** (forward motion estimation) ve **geri hareket kestirimi** (backward motion estimation) yapılarak nesnelerin birbirini kapattığı bölgelerde gerçek hareket vektörlerinin bulunmasına çalışılır.

Hesaplanan hareket vektörlerinin doğruluklarının arttırılması ve yanlış hareket vektörlerinin düzeltilmesi için hareket vektörlerinin işlendiği (motion vector processing) yöntemler kullanılmıştır. **Hareket alanı** (motion field) nesne sınırları dışında genellikle düzgündür. Hareket vektörü hatalarının belirlenebilmesi için hareket vektörü modellenmelidir. Dane ve Nguyen (2004) tarafından istatistiksel modele dayanan hareket vektörü işleme algoritması önerilmiştir.

Zhai vd. (2005) tarafından geliştirilen yöntemde kodlayıcının yolladığı hareket vektörleri iki sınıfa ayrılmaktadır: gerçek hareketi temsil ettikleri düşünülen, bu nedenle aradeğerlemede doğrudan kullanılacak hareket vektörleri ve yanlış hareket vektörleri. İkinci sınıf hareket vektörleri için çiftyönlü hareket kestirimi uygulanmaktadır. Diğer pek çok aradeğerleme algoritmasında olduğu gibi hareket alanını düzgünleştirmek için hareket vektörleri sonradan işlenmektedir.

2. VIDEO

2.1 Giriş

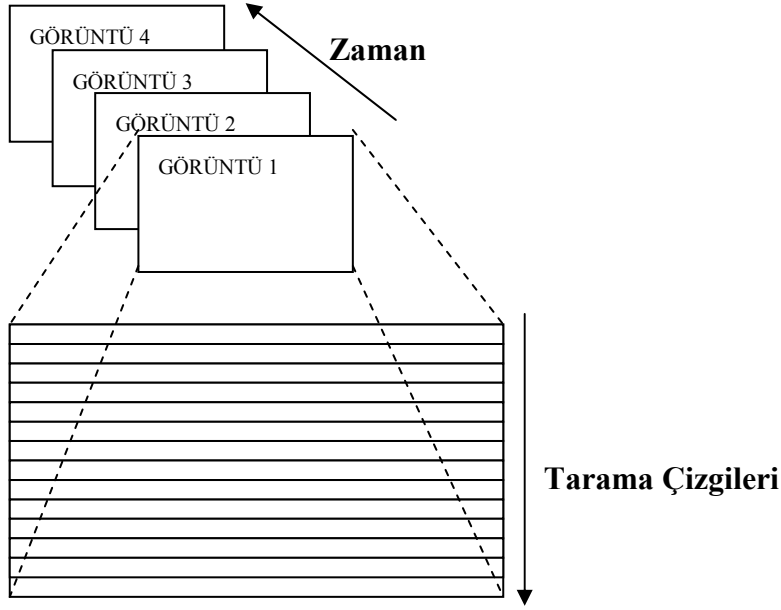
Video işleme algoritmaları videoyu oluşturan bir görüntü dizisi üzerinde çalışır. Tipik bir video işleme algoritmasında, görüntü dizisindeki her bir durağan görüntü diğerlerinden bağımsız olarak işlenebilir ya da bir grup görüntü bir arada ele alınabilir. Bu bölümde video işleme algoritmalarının en temel işlem birimi olan video kavramı üzerinde durulmuştur.

Durağan görüntü dizisinden hareketli video algılamasının nasıl elde edildiği sayısal video tanımı içerisinde verilmiştir. Videoyu oluşturan her bir durağan görüntüye çerçeve adı verilmektedir. Bir videonun niteliğini belirleyen önemli parametrelerden biri videonun çerçeve hızıdır. Çerçeve hızı birim zamanda gösterilen çerçeve sayısını veren bir değerdir. Bir videoyu betimleyen önemli bir kavram da renktir. RGB renk uzayının yanı sıra video standartlarında kullanılan YCbCr ve YUV uzayları üzerinde durulmuştur. Özellikle YUV uzayı PAL, NTSC ve SECAM gibi video standartlarında kullanılmaktadır. Videonun alınması, iletilmesi ve gösterilmesi konuları incelenmiştir. Video sinyalinin daha etkin bir biçimde iletilmesinde kullanılan renk altörnekleme ve video çözünürlüğünü belirleyen video formatları konularına da değinilmiştir.

2.2 Sayısal Video

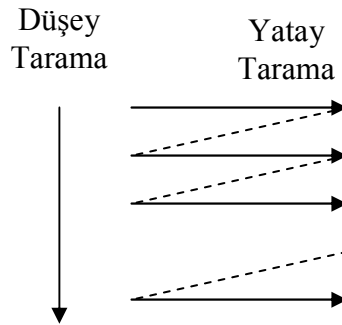
Hareketli görüntülerdeki hareket olgusu aslında, bir durağan görüntü dizisinin insanın görme sisteminde süreklilik algısı oluşturacak biçimde hızlı değişiminden ortaya çıkar. Sözü geçen durağan görüntü dizisi doğal bir sahnenin konumsal ve zamansal örneklenmesi ile elde edilir. Örnekleme hareketli video sinyalini üretmek için (örneğin 1/25 ya da 1/30 saniyelik) aralıklarda yinelenir.

Doğal bir sahne konumsal ve zamansal olarak süreklidir. Böyle bir sahnenin sayısal biçime dönüştürülmesi (genellikle video görüntü düzlemindeki dikdörtgen bir kafeste) konumsal ve (durağan görüntü dizisi ya da zamanda belirli aralıklarda örneklenen görüntülerin bileşenleri olarak) zamansal örnekleme ile gerçekleştirilir. Sayısal video örneklenen görsel sahnenin sayısal biçimde betimlenmesidir. Her bir uzam-zamansal örnek (imge ögesi ya da piksel) parlaklığını ya da rengini veren bir sayı ya da sayı kümesi olarak temsil edilir.



Şekil 2.1 Video oluşturan durağan görüntü dizisi. Her bir görüntü ise ayrı ayrı veri satırlarından oluşur (Jack, 2005)

Doğal bir video sahnesinin video işleme ve sıkıştırma ile ilgili ayırt edici özellikleri sahne içerisindeki doku değişimi, nesnelerin sayısı ve şekilleri, renk vb. konumsal ayrıtlar ve nesne hareketi, aydınlatmadaki değişiklikler, kameranın hareketi ya da bakış açısı, vb. zamansal ayrıtlardır.



Şekil 2.2 Video çerçevesinde tarama çizgileri (Jack, 2005)

2.3 Çerçeve Kavramı

Videoyu oluşturan her bir durağan görüntü çerçeve olarak adlandırılır. Çerçeveler yukarıdan aşağıya doğru birbirini ardışıl olarak izleyen tarama çizgilerinden oluşmaktadır (Şekil 2.1). Videonun görüntülenmesinde kullanılan iki ayrı zamanlama söz konusudur. Düşey

eşzamanlama yeni bir çerçevenin, yatay eşzamanlama ise yeni bir tarama çizgisinin başlangıcını belirtir. Videonun ekranda gösterilmesi, ekranın sol üst köşesinden başlayarak her bir tarama çizgisi için ekranın solundan sağ yanına doğru ilerlenmesi ve bir tarama çizgisi tamamlandığında bir sonraki tarama çizgisine geçilerek soldan sağa tarama işleminin yinelenmesi biçimindedir (Şekil 2.2). Düşey ve yatay eşzamanlama bilgisi genellikle şu üç yoldan birisi ile aktarılır (Jack, 2005).

- Ayrı, yatay ve düşey eşzamanlama sinyalleri
- Ayrı, bileşik eşzamanlama sinyali
- Video sinyaline gömülü bileşik eşzamanlama sinyali

Bileşik eşzamanlama sinyali düşey ve yatay eşzamanlamanın bir birleşimidir. Sayısal video için, yaygın olarak birinci yöntem kullanılır ya da zamanlama kod sözcükleri sayısal video bilgisinde gömülüdür.

Herhangi bir görüntü biriminde 1 saniyede oynatılan çerçeve sayısına çerçeve hızı denir. Çerçeve hızları tipik olarak saniyede 24, 25 ya da 30 çerçeve olabilir. Saniye başına gösterilen çerçeve sayısının artırılması hareketli görüntüdeki değişikliklerin görünümse olarak daha düzgün olarak algılanmasını sağlayacaktır. Ancak bu durumda daha fazla örneğin yakalanması ve aktarılması gerekmektedir. Saniyede 10 çerçevenin altındaki çerçeve hızları genellikle (veri miktarı görece az olduğundan) çok düşük bit hızlı video iletişimi için kullanılır ancak hareket bu hızda açıkça kesintilidir ve doğal değildir. Saniyede 10 ve 20 çerçeve arasındaki hızlarda görüntü daha pürüzsüzdür ancak görüntü dizisinin hızlı hareket eden bölümlerinde kesintili hareket gözlemlenebilir. Saniyede 25 ya da 30 çerçeve ile örnekleme televizyon görüntüleri için standarttır. Saniyede 50 ya da 60 çerçevelik hızlarda oldukça yüksek veri hızına karşın belirli, düzgün bir hareket algısı oluşur (Richardson, 2003).

2.4 Videonun Renk Özellikleri

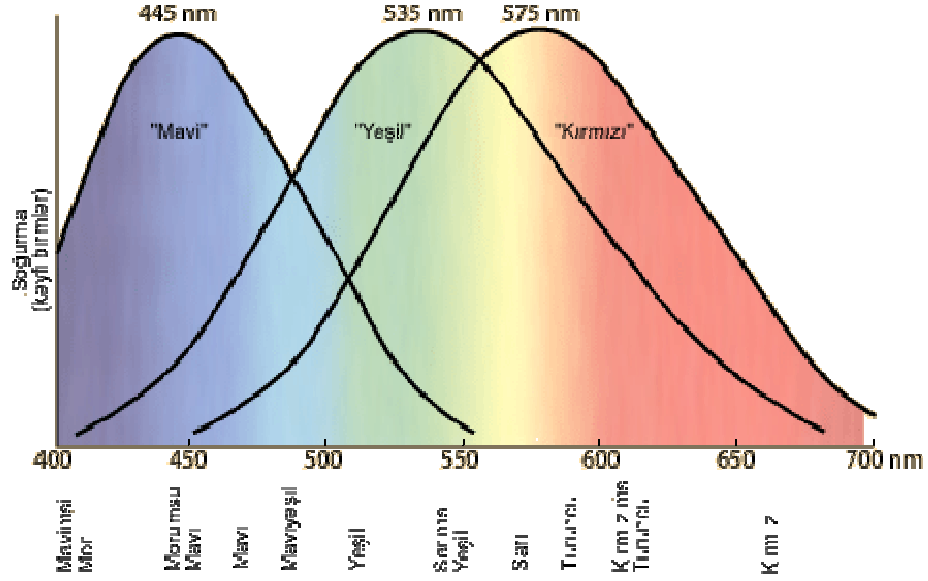
Video 3-boyutlu dinamik bir sahnenin video kamerasının görüntü düzlemi üzerine izdüşümünden elde edilen 2-boyutlu görüntü dizisidir. Video sinyali iki konumsal boyuta (yatay ve düşey) ve bir zamansal boyuta sahip 3-boyutlu bir sinyaldir. Bir video çerçevesindeki herhangi bir noktanın renk değeri gözlemlenen sahnedeki belirli bir 3-boyutlu noktadan yayılan ya da yansıyan ışığı belirtir. Renk değerinin fiziksel olarak ne anlama geldiğini anlayabilmek için, ışığı ve renk bilgisini tanımlayan özellikler ele alınmalıdır.

2.4.1 Renk Kavramı

Bir kromatik ışık kaynağının niteliğini betimleyebilmek için üç temel nicelik kullanılır: ışınlılık (radiance), ışıklılık (luminance) ve parlaklık (brightness). Işınlılık ışık kaynağından akan toplam enerji miktarıdır ve genellikle watt cinsinden ölçülür. Lümen cinsinden ölçülen ışıklılık gözlemcinin bir ışık kaynağından algıladığı enerji miktarının bir ölçüsünü verir. Örneğin, spektrumun kızılötesi bölgesinin uzağında çalışan bir kaynaktan yayılan ışık önemli miktarda enerjiye, ışınlılığa, sahip olabilir ancak bir gözlemci ışığı neredeyse hiç algılamayabilir; ışıklılık neredeyse sıfır olabilir. Parlaklık ise ölçülmesi olanaksız olan öznel bir betimleyicidir. Parlaklık yeğinliğin (intensity) kromatik olmayan özelliğini belirtir ve renk algılamasının betimlenmesindeki anahtar etmenlerden birisidir (Gonzales ve Woods, 2002).

Genel olarak bir rengi diğerinden ayırt etmekte kullanılan nitelikler renk tonu (hue), doygunluk (saturation) ve parlaklıktır. Renk tonu bir ışık dalgaları karışımındaki baskın dalga boyu ile ilişkilendirilen bir özniteliktir. Bu nedenle, bir nesne kırmızı, turuncu ya da sarı olarak adlandırıldığında renk tonu belirtilmektedir. Doymunluk görelî saflığı ya da bir renk tonu ile karıştırılmış beyaz ışığın miktarını belirtmektedir. Saf spektrum renkleri tam doymundur. Pembe (kırmızı ve beyaz) ve eflatun (mor ve beyaz) gibi renkler eklenen beyaz ışık miktarıyla ters orantılı olan doymunluk derecesiyle az doymundur. Renk tonu ve doymunluk bir arada renklilik (chromaticity) olarak adlandırılır. Bu nedenle bir renk parlaklığı ve renkliliği ile nitelendirilebilir (Gonzales ve Woods, 2002).

Renk bir yüzeyin özniteliği değildir, daha çok renk algısının ışık işaretlerinin işlenmesiyle oluşturulduğu beyinsel bir etkinliktir. Işık optik algılayıcıları uyarır ve hem retinadaki hem de beynin bazı bölümlerindeki sinyal işlemeden sonra, nesne yüzeyinin rengini belirleyen renk algısı oluşur (Bigun, 2006).



Şekil 2.3 Dalga boyuna bağlı olarak insan gözündeki kırmızı, yeşil ve mavi koni hücrelerinin ışığı soğurması (Gonzales ve Woods, 2002)

Retina üzerinde iki tür optik algılayıcı bulunur; çubuk ve koni hücreleri. Koni hücreleri renkli görmeden sorumludur. İnsan gözündeki 6 ila 7 milyon koni hücresi kabaca kırmızı, yeşil ve maviye karşılık gelen başlıca üç algılama sınıfına ayrılabilir. Tüm koni hücrelerinin yaklaşık %65'i kırmızı ışığa, %33'ü yeşil ışığa ve yalnızca %2'si maviye duyarlıdır. Ancak mavi koni hücreleri en duyarlı olanlarıdır. Şekil 2.3 gözdeki kırmızı, yeşil ve mavi koni hücrelerinin ışığı soğurmalarını ayrıntılandıran ortalama deneysel eğrileri göstermektedir. İnsan gözünün soğurma niteliğine bağlı olarak, renkler ana renkler olarak adlandırılan kırmızı, yeşil ve mavinin değişken birleşimleri olarak ele alınmaktadır (Gonzales ve Woods, 2002).

İnsanın görme sistemi parlaklığa göre renge daha az hassastır. İnsanın görme sisteminin bu özelliğinden video çerçevelerinin saklandığı ya da aktarıldığı renk modellerinin belirlenmesinde yararlanılmıştır.

2.4.2 Video Tekniğinde Kullanılan Renk Modelleri

Renk uzayı bir renk kümesinin matematiksel gösterimidir. Kısaca, bir renk modeli her rengin tek bir nokta ile temsil edildiği bir koordinat sisteminin belirtimi ve bu sistem içindeki bir altuzaydır. Bugün kullanımdaki renk modellerinin çoğu ya donanıma (renkli monitörler ve yazıcılar gibi) ya da renk işlemenin bir amaç olduğu uygulamalara (canlandırma için renkli grafiklerin yaratılmasında olduğu gibi) yöneliktir. En çok kullanılan üç renk modeli bilgisayar grafiklerinde kullanılan RGB, video sistemlerinde kullanılan YIQ, YUV ya da YCbCr ve

renkli baskıda kullanılan CMYK modelleridir. Ancak bu renk uzaylarının hiç biri renk tonu, doygunluk ve parlaklık kavramlarıyla doğrudan ilişkili değildir. Bu nedenle programlamayı, işlemeyi ve son kullanıcının kullanımını yalınlaştırmak için HSI ve HSV gibi modeller de geliştirilmiştir. Renk uzaylarının tümü kamera ve tarayıcı gibi çevre birimlerinden elde edilen RGB bilgisinden çıkarılabilir.

2.4.2.1 RGB Renk Uzayı

RGB uzayı bir toplanır renk sistemidir çünkü renkler bileşenlerin siyaha (0, 0, 0) eklenmesiyle oluşturulur. Renkli katot ışıını tüpleri ve sıvı kristal ekranlar bir RGB görüntüsünü her bir pikselin kırmızı, yeşil ve mavi bileşenlerini her bir bileşenin yeğlinliğine göre ayrı ayrı aydınlatarak gösterirler. Bir piksele ilişkin fosforun üç komşu ögesine yoğunlukları sırasıyla c_1 , c_2 ve c_3 olan üç elektron ışıını düşürölür. CRT ekranının küçük bir bölgesinde üç bileşenden meydana gelen ışık fiziksel olarak toplanır ve karıştırılır. İnsanın görme sistemi rengi algılamak için (c_1 , c_2 , c_3) parlaklıklarını tümleştirir. Bir RGB görüntüsünün elde edilmesi sahnenin kırmızı, yeşil ve mavi bileşenlerinin süzölmesini ve her birinin ayrı bir algılayıcı dizilimi ile eldesini içermektedir (Shapiro ve Stockman, 2001).

RGB uzayındaki her bir pikseli temsil etmek için kullanılan bit sayısı piksel derinliği olarak adlandırılır. Örneğın her bir kırmızı, yeşil ve mavi görüntüsünün 8 bitlik görüntü olduğı bir RGB görüntüsü 24 bit derinliğindedir. Tam renkli görüntü kavramı 24 bit RGB renkli görüntüsünü belirtmek için sıklıkla kullanılmaktadır

RGB gerçek görüntülerde çok etkin değildir. Üç RGB bileşeni de RGB renk küpündeki herhangi bir rengi oluşturabilmek için eşit bantgenişliğine gereksinim duyar. Üç renk sinyalinin üç ayrı sinyalle betimlendiğı video formatına bileşenli video denir. Her bir RGB bileşeni için aynı piksel derinliğine ve ekran çözünürlüğüne sahip bir video arabelleğine gereksinim duyulur. Ayrıca, bir görüntünün RGB renk uzayında işlenmesi her zaman en etkin yöntem değildir. Örneğın, bir pikselin yeğlinliğini ya da rengini değıştirmek için video arabelleğinden üç RGB değeri okunmalı, yeğinlik ya da renk hesaplanmalı, istenilen değışiklikler gerçekleştirilmeli ve yeni RGB değeri hesaplanarak video arabelleğine geri yazılmalıdır. Bu ve diğeri nedenlerle, pek çok video standardı parlaklık ve iki renk farkı sinyallerini kullanmaktadır. En yaygın olanları YCbCr, YUV ve YIQ renk uzaylarıdır. Hepsi birbiriyle ilişkili olmasına karşın, bazı ayrımları vardır. YUV ve YIQ sayısal görüntülerin ve videoların sıkıştırılmasında diğeri renk kodlama yöntemlerine göre daha iyi bir potansiyele sahiptir, çünkü RGB’de olanaklı olmayan biçimde parlaklık ve renk bilgileri farklı sayıda bit

kullanılarak kodlanabilir.

2.4.2.2 YCbCr Uzayı

İnsanın görme sistemi parlaklığa daha duyarlı olduğundan renk bilgisinin RGB uzayına göre daha etkin kodlandığı YCbCr uzayı geliştirilmiştir. Y parlaklık bileşenidir ve kırmızı (R), yeşil (G) ve mavinin (B) ağırlıklı ortalaması olarak hesaplanabilir (2.1).

$$Y = k_r R + k_g G + k_b B \quad (2.1)$$

Renk bilgisi her renk bileşenin kırmızı, yeşil ya da mavi ile parlaklık arasındaki fark olduğu renk farkı bileşenleri olarak temsil edilebilir (2.2)

$$\begin{aligned} C_r &= R - Y \\ C_g &= G - Y \\ C_b &= B - Y \end{aligned} \quad (2.2)$$

Bir renkli görüntünün tam bir betimlemesi Y (parlaklık bileşeni) ve her bir pikselin ortalama parlaklığı ve renk yeğlilikleri arasındaki farkları temsil eden Cr, Cg ve Cb ile yapılabilir. Bu biçimde gösterimin açık bir yararı yoktur çünkü RGB'deki üç bileşen yerine dört bileşen söz konusudur. Ancak $C_r + C_g + C_b$ toplamı sabittir ve üç renk bileşeninden yalnızca ikisi saklanabilir ya da iletebilir. Çünkü üçüncü bileşen her zaman diğer ikisinden hesaplanabilir. YCbCr renk uzayında, yalnızca parlaklık (Y) ve mavi ve kırmızı (Cb, Cr) renkler iletilir. YCbCr uzayının RGB'ye göre, insanın görme sisteminin özelliğinden yararlanılarak Cb ve Cr bileşenlerinin Y'ye göre daha düşük çözünürlükle temsil edilebilmesi gibi bir avantajı vardır. Böylece görsel nitelikte açık bir etki olmaksızın renk bileşenlerinin temsil edilmesi için gerekli veri miktarı azalmaktadır. Aslında renk bilgisinin bu biçimde parlaklığa göre daha düşük çözünürlükte temsil edilmesi görüntü sıkıştırmanın basit ancak etkin bir biçimidir (Richardson, 2003).

Bir RGB görüntüsü elde edildikten sonra saklama ve/veya iletim gereksinimlerini azaltmak için YCbCr renk uzayına dönüştürülebilir. Genellikle videonun görüntülenmesinden önce yeniden RGB'ye dönüşüm yapılması gerekli olmaktadır. YCbCr uzayından RGB uzayına ve RGB uzayından YCbCr uzayına dönüşüm eşitlikleri sırasıyla eşitlik (2.3) ve (2.4)'de belirtilmiştir.

$$\begin{aligned}
R &= Y + 1.402Cr \\
G &= Y - 0.344Cb - 0.714Cr \\
B &= Y + 1.772Cb
\end{aligned} \tag{2.3}$$

$$\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B \\
Cb &= 0.564(B - Y) \\
Cr &= 0.713(R - Y)
\end{aligned} \tag{2.4}$$

2.4.2.3 YUV Uzayı

YUV renk uzayı PAL, NTSC ve SECAM gibi video standartlarınca kullanılmaktadır. Siyah ve beyaz sistem yalnızca parlaklık bilgisini kullanır. Renkli video sinyaline renk bilgisi (U ve V) öyle bir biçimde eklenmiştir ki siyah ve beyaz alıcı renkli video görüntüsünü normal bir siyah ve beyaz resim olarak gösterecektir. Renkli alıcılar renkli bir resmi görüntüleyebilmek için ek renk bilgisini çözerler (Jack, 2005). Bileşik video formatlarında üç renk sinyali böylece tek bir sinyale bindirilir. RGB'nin YUV'a çevrilmesi (2.5)'te verilmiştir.

$$\begin{aligned}
Y &= 0.30R + 0.59G + 0.11B \\
U &= 0.493 * (B - Y) \\
V &= 0.877 * (R - Y)
\end{aligned} \tag{2.5}$$

2.5 Video Alımı

Tüm analog kameralar videoyu çerçeveler biçiminde, çerçeveler arasında belirli bir zaman aralığı bırakarak elde eder. Video alımında ve gösteriminde insanın görme sisteminin zamanda ve uzayda çok yüksek frekans değişimlerini algılayamamasından yararlanılmaktadır. Kameranin merceği sahnenin görüntüsünü kamera görüntüleyicisinin optik sinyalleri elektrik sinyallerine dönüştüren ışığa duyarlı yüzeyine odaklar. Analog kameraların ışığa duyarlı görüntüleyici yüzeyi bir elektron demeti ya da başka bir elektronik yöntem ile satır satır taranır (raster scan) ve her bir çerçevedeki taranmış satırlar ışık yenginliğindeki değişimleri elektriksel gerilimdeki değişimler olarak temsil eden bir elektrik sinyaline dönüştürülür. Böylelikle farklı satırlar biraz farklı zamanlarda sürekli bir biçimde alınır (Wang vd., 2002).

Sayısal video analog taramayı örnekleyerek ya da doğrudan sayısal video kamerası kullanılarak elde edilebilir. Analog kameralarda olduğu gibi, sayısal kamera görüntülenen sahneyi ayrık çerçeveler halinde örnekler. Bir CCD kamerasında ışığa duyarlı yüzey her biri bir piksele karşılık gelen algılayıcılar matrisinden oluşur ve her bir algılayıcıya ulaşan optik sinyal elektrik sinyaline dönüştürülür. Her bir çerçeve alımında yakalanan algılayıcı değerleri

ilk önce bir arabellekte tutulur ve sonra satır satır ardışıl olarak arabellekten okunur. Analog kameraların aksine, aynı çerçevede okunan tüm değerler aynı zamanda yakalanır.

Rengi yakalamak için genellikle üç tür ışığa duyarlı yüzey ya da CCD algılayıcıları vardır. Ancak çoğu CCD kamerasında renkli görüntüleme için tek bir CCD yongası kullanılır. Bunun için algılayıcı yüzeyindeki pikseller her biri farklı bir ana renge duyarlı olan üç ya da dört alt alana bölünür. Yakalanan üç renk sinyali bir parlaklık ve iki renk sinyaline dönüştürülerek bileşenli renkli video olarak gönderilir ya da bileşik video sinyaline dönüştürülür (Wang vd., 2002).



Şekil 2.4 Renkli katot ışını tüpünün kesit resmi (Resimdeki sayılar şu bölümleri belirtmektedir: 1. Elektron tabancası 2. Elektron ışını 3. Görüntülenen resmin kırmızı, yeşil ve mavi bileşenleri için ışınları ayırıcı maske 4. Kırmızı, yeşil ve mavi bölgelere sahip fosfor tabaka 5. Ekranın fosfor kaplı iç kısmının yakından görünümü)

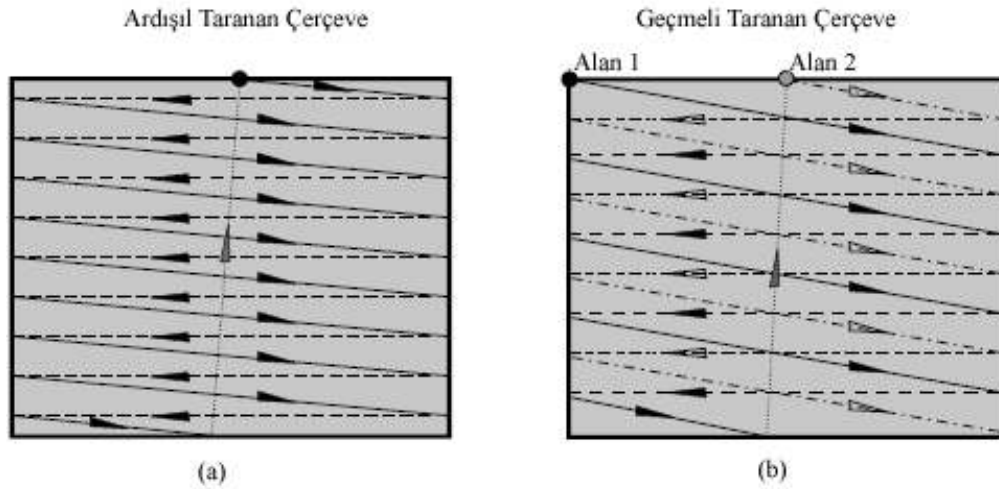
2.6 Video Gösterimi

CRT ile video gösteriminde, bir elektron tabancası ekran boyunca satır satır tarama yaparak ilgili konumlarda video sinyalinin yeğlinliğiyle orantılı yeğlilikle ekran yüzeyindeki fosforlu maddeyi uyaran bir elektron demeti yayar (Şekil 2.4). LCD’lerde ise uygulanan elektrik alanı ile sıvı kristalin optik özellikleri ve sonuç olarak parlaklığı/rengi değiştirilir.

Bileşik video sinyalini renkli bir monitörde görüntülemek için, modüle edilen renk sinyallerini ve parlaklık sinyalini birbirinden ayırmak için bir filtre kullanılır ve elde edilen renk ve parlaklık bileşenleri kırmızı, yeşil ve mavi renk bileşenlerine dönüştürülür. Tek renkli bir monitörde yalnızca parlaklık sinyali elde edilir ve doğrudan gösterilir (Wang vd., 2002).

2.7 Video İletimi

Televizyonun ilk zamanlarında, her bir çerçeve için gönderilen bilgi miktarını azaltmak için geçmeli tarama denilen bir teknik kullanılmıştır. Bir video dizisinde bir çerçevedeki tek sayılı satırları, bir sonraki çerçeve ise çift sayılı satırları aktararak, her bir çerçeve için gönderilen bilgi miktarı yarıya indirilir. Bir çerçevedeki tek sayılı ya da çift sayılı satırlar alan olarak adlandırılır. Her bir alan çerçevedeki satır sayısının yarısı kadar satıra sahiptir. İki alan arasındaki zaman aralığı çerçeve zaman aralığının yarısı kadardır. Ardışıl taramada ise bir çerçevedeki tüm satırlar peşi sıra taranır. PAL formatında çerçeve hızı saniyede 25 çerçevedir. Ancak geçmeli tarama ile saniyede gerçekte 50 çerçeve oynatılır, böylece saniyede 25 çerçeve içeren eşdeğer ardışıl video dizisine göre hareket daha düzgündür. Geçmeli taramanın sözü geçen getirisine karşın geçmeli taramada her bir tarama satırı ardışıl taramanın yarısı kadar bir sıklıkla tazelenir. Çok düşük tazeleme hızına bağlı olarak keskin kenarlardaki satır titremesini önlemek için görüntü düşey olarak alçak geçiren filtre ile işleme sokularak satırdan satıra değişimler kısıtlanır (Richardson, 2003).



Şekil 2.5 Ardışıl (a) ve geçmeli (b) satır satır tarama formatları (Wang vd., 2002)

2.8 Renk Altörneklemesi

Renk altörneklemesi bir görüntüdeki renk bilgisi için parlaklık bilgisine göre daha düşük bir çözünürlük kullanılmasıdır. İnsan gözü renge parlaklıktan daha az duyarlı olduğu için, bir görüntünün renk bileşenlerinin parlaklık bileşeni kadar iyi tanımlanmasına gerek yoktur. Bu ise resim niteliğinde görünür bir kayıp olmaksızın video sinyalinin net bantgenişliğini

azaltmaktadır. Kayıp değerler o kanal için önceki örnekten yararlanarak aradeğerleme ya da yineleme ile elde edilecektir.

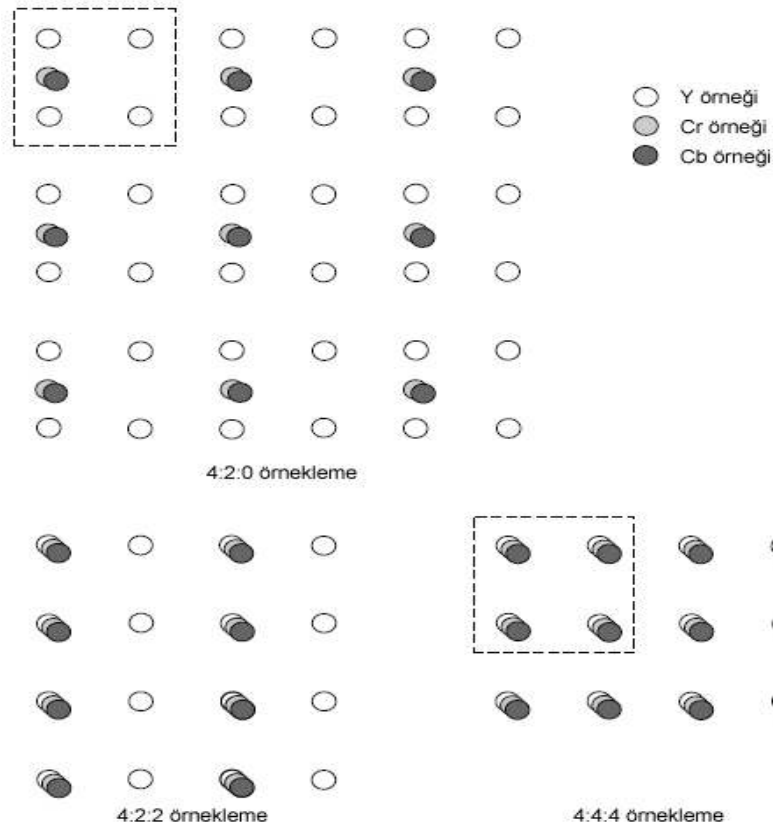
4:4:4 örnekleme üç bileşenin aynı çözünürlüğe sahip olduğu ve bu nedenle her piksel konumunda her bir bileşenin birer örneği olduğu anlamına gelmektedir. Sayılar yatay doğrultuda her bir bileşenin görelî örnekleme oranını belirtmektedir. Buna göre 4:4:4 örneklemede her dört parlaklık örneğine karşılık dört Cb ve dört Cr örneği vardır. 4:2:2 örneklemede renk bileşenleri parlaklıkla aynı düşey çözünürlüğe sahiptir ancak yatay çözünürlükleri parlaklığın yarısı kadardır. Buna göre yatay doğrultudaki her dört parlaklık örneği için iki Cb ve iki Cr örneği vardır.

Yaygın olarak kullanılan 4:2:0 örnekleme Cb ya da Cr bilgisinin saklanmadığı anlamına gelmemektedir. Cb ve Cr'nin her ikisi de Y'nin yatay ve düşey çözünürlüğünün yarısına sahiptir.

Örneğin çözünürlüğü 720x576 piksel olan bir görüntüde Y'nin çözünürlüğü her biri 8 bitle temsil edilen 720x576 örnek olacaktır. 4:4:4 örneklemede Cb, Cr çözünürlüğü benzer biçimde her biri 8 bit olan 720x576 örnek olacaktır. Bu durumda kullanılacak toplam bit sayısı (2.6) ile verilmiştir. 4:2:0 örneklemede Cb, Cr çözünürlüğü her biri 8 bit olan 360x288 örnektir. Buna göre kullanılacak toplam bit sayısı (2.7)'de belirtilmiştir.

$$720 \times 576 \times 8 \times 3 = 9953280 \quad (2.6)$$

$$(720 \times 576 \times 8) + (360 \times 288 \times 8 \times 2) = 4976640 \quad (2.7)$$



Şekil 2.6 4:2:0, 4:2:2 ve 4:4:4 örnekleme örüntüleri (Richardson, 2003)

4:2:0 örnekleme bazen piksel başına 12 bit olarak tanımlanır. 4:4:4 örnekleme kullanıldığında, Y, Cb ve Cr bileşenlerinin her biri için dört pikselde toplam 12 örneğe gereksinim duyulur. Böylece toplam $12 \times 8 = 96$ bite gereksinim vardır ve ortalama piksel başına $96/4 = 24$ bittir. 4:2:0 örneklemede ise, Y'nin dört ve Cb, Cr'nin her birinin birer örnekle temsil edildiği yalnızca 6 örneğe gereksinim duyulur. Böylece toplam $6 \times 8 = 48$ bite gereksinim vardır ve ortalama piksel başına $48/4 = 12$ bittir (Richardson, 2003).

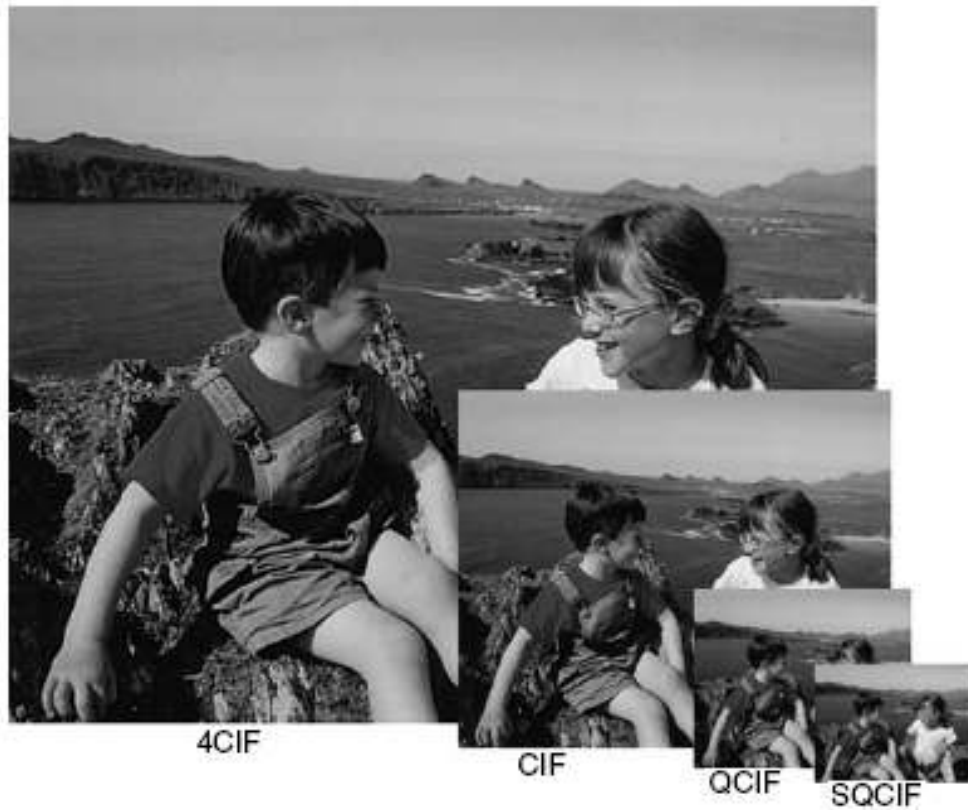
Çizelge 2.1 Video çerçeve formatları

Format	Parlaklık Çözünürlüğü (yatay x dikey)	Çerçeve Başına Bit Sayısı (4:2:0, örnek başına 8 bit)
Sub-QCIF (SQCIF)	128 x 96	147456
Quarter CIF (QCIF)	176 x 144	304128
CIF	352 x 288	1216512
4CIF	704 x 576	4866048

2.9 Video Çerçeve Formatları

Çerçeve çözünürlüğünün seçimi uygulamaya ve eldeki saklama ya da aktarma kapasitesine bağlıdır. CIF Çizelge 2.1’de gösterilen yaygın çerçeve formatları için bir temel oluşturmaktadır.

4CIF standart tanımlı televizyon ve DVD-video için uygundur; CIF ve QCIF video konferans uygulamalarında yaygındır; QCIF ya da SQCIF ekran çözünürlüğünün ve bit hızının sınırlı olduğu mobil çoğulortam uygulamaları için uygundur.



Şekil 2.7 Çeşitli çözünürlüklerde örneklenen video çerçeveleri (Richardson, 2003)

3. 2-BOYUTLU HAREKET KESTİRİMİ

3.1 Giriş

Hareket eden nesnelerin seziminde ya da gidim-izlerinin (trajectory) hesaplanmasında bir görüntü dizisindeki değişikliklerden yararlanılabilir. Hareket, bir nesnenin biçimini ve hız ya da işlevi gibi ayrıtlarını ortaya çıkarır. Genel olarak hareket şu dört genel durumda incelenebilir (Shapiro ve Stockman, 2001).

- Durağan kamera, tek hareketli nesne, değişmez arkaplan
- Durağan kamera, pek çok hareketli nesne, değişmez arkaplan
- Hareketli kamera, görece değişmez arkaplan
- Hareketli kamera, pek çok hareketli nesne

En yalın durum durağan kameranın görece değişmez bir arkaplana sahip sahneyi çekmesidir. Böyle bir arkaplan boyunca hareket eden nesneler nesneyle ilişkili görüntü piksellerinde değişikliklere neden olacaktır. Bu piksellerin sezimi nesnenin biçimini, hızını ve gidim-izini belirtir. En zor hareket sorunları herhangi bir değişmez arkaplanın olmadığı, hareketli kameranın ve pek çok hareket eden nesne içeren sahnenin bulunduğu durumlarda ortaya çıkar.

Bir görüntü dizisindeki harekete bağlı olarak değişikliklerin sezimi ya da nesnelerin ya da hareketlerinin çözümlenmesi video işleme sistemlerinde önemli bir yer tutar. Hareket kestiriminde bir ya da daha fazla sayıda referans çerçevedeki varolan veriyi temel alarak çalışılan çerçeve için bir hareket modeli oluşturulur. Hareket kestirimi, hareketli görüntünün bir çerçevesindeki nesnelere ve arkaplana ilişkin örüntülerin çerçeve içerisinde hareket ederek sonraki çerçevedeki ilişkili nesneleri oluşturması fikrine dayanmaktadır. Hareket kestiriminin uygulamaları arasında video sıkıştırması, video aradeğerlemesi ve video filtreleme sayılabilir.

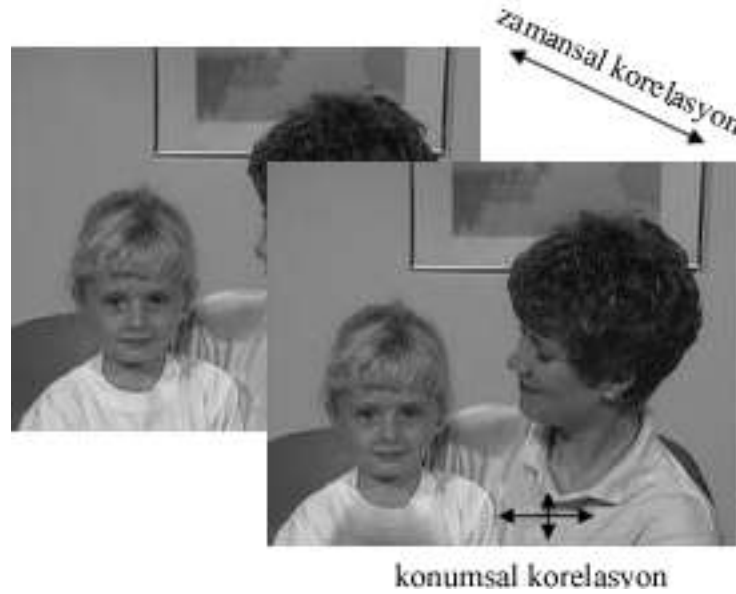
Hareket kestiriminde temel sorun hareket alanının tanımlanmasıdır. Görüntülenen sahnede farklı biçimde hareket eden pek çok hareketli nesne vardır. Bu nedenle tüm çerçeve için geçerli olan genel bir hareket modeli yeterli değildir. Çerçeve içindeki her bir piksel için bir hareket vektörü hesaplanabilir. Böylelikle çerçevenin hareket modeli piksel seviyesinde belirlenmiş olur. Yalnızca kamera hareket ediyorsa ya da görüntülenen sahne tek bir hareketli nesne içeriyorsa tüm hareket alanını nitelendirmek için genel bir hareket betimi kullanılabilir.

Pek çok hareketli nesne içeren sahneler için, çerçeveler her bir bölge içindeki hareketin parametrik bir modelle iyi tanımlandığı pek çok bölgeye ayrılabilir. Bölge seviyesinde hareket kestiriminde bir bölge bölütleme bilgisi ve her bir bölge için pek çok hareket parametresi söz

konusudur. Bu yaklaşımın zorluğu hangi piksellerin benzer hareketi yaptıklarının önceden bilinmemesidir. Bu nedenle, bölütleme ve kestirimin yinelemeli bir biçimde yapılması gerekir ancak bu, uygulamada pek olanaklı olmayabilir. Bölge seviyesindeki hareket kestiriminin karmaşıklığını azaltmanın bir yolu çerçevenin pek çok küçük blok halinde değişmez bölümlere ayrılmasıdır. Her blok yeteri kadar küçük olduğu sürece, her bir blok içerisindeki hareket değişimi yalın bir modelle iyi bir biçimde tanımlanabilir ve hareket parametreleri her bir blok için bağımsız olarak kestirilebilir (Wang vd., 2002).

3.2 Hareket Kestiriminin Video Sıkıştırmasında Uygulanması

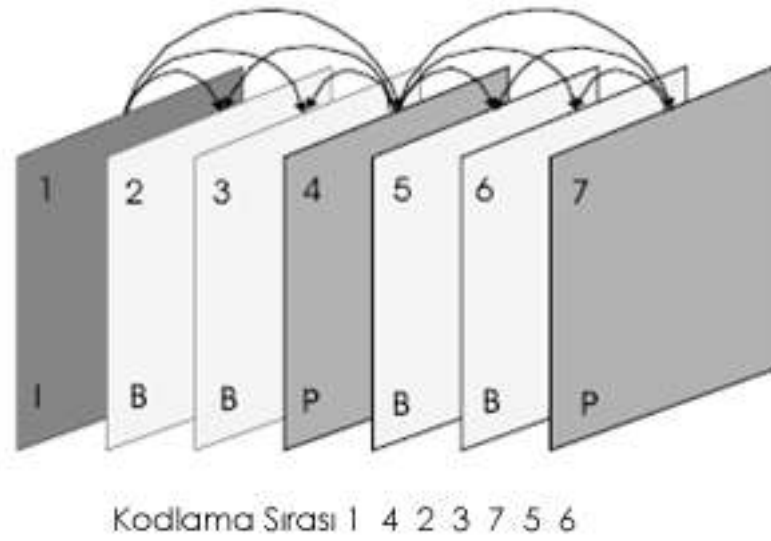
Bir çerçevedeki herhangi bir piksel seçildiğinde, yüksek olasılıkla yakın komşuları aynı renge ya da oldukça benzer renklere sahiptir. Bu nedenle görüntü sıkıştırma komşu piksellerin komşularıyla ilişkili olmaları olgusuna dayanmaktadır. Bu ilişki aynı zamanda konumsal artıklık olarak da adlandırılır. Video sıkıştırmasının dayandığı iki ilkedен biri durağan görüntülerdeki konumsal artıklıktır.



Şekil 3.1 Bir görüntü dizisindeki konumsal ve zamansal korelasyon

Bir videonun çerçeve hızının artırılması ardışıl iki çerçeve arasındaki örnekleme zamanını azaltacaktır. Çerçeve hızı arttıkça ardışıl çerçeveler arasındaki benzerlik daha fazla olacaktır. Video sıkıştırmasının dayandığı diğer bir ilke olan bu özelliğe zamansal artıklık denir. Öyleyse yüksek çerçeve hızına sahip bir hareketli görüntüde birbirine komşu çerçeveler arasındaki farklılıklar belirlenebilir. Hareket kestirimi bir durağan görüntü dizisinde

çerçeveler arasındaki benzerlikten yararlanarak hareket değişimlerinin hesaplanmasında kullanılabilir.

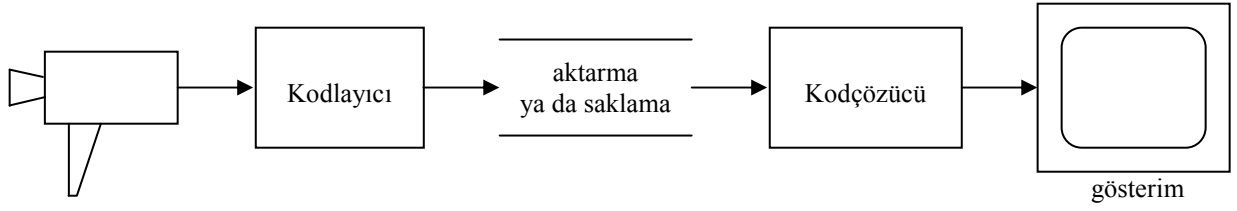


Şekil 3.2 Hem tekyönlü hem de çiftyönlü zamansal kestirim kullanılarak video kodlaması. Oklar kodlanan çerçevenin kestirilmesinde kullanılan referans çerçeveleri belirtmektedir. I, P ve B harfleri ile etiketlenen çerçeveler sırasıyla bağımsız olarak, tekyönlü kestirim ile ve çiftyönlü kestirim ile kodlanmıştır (Wang vd., 2002)

3.2.1 Video Sıkıştırması

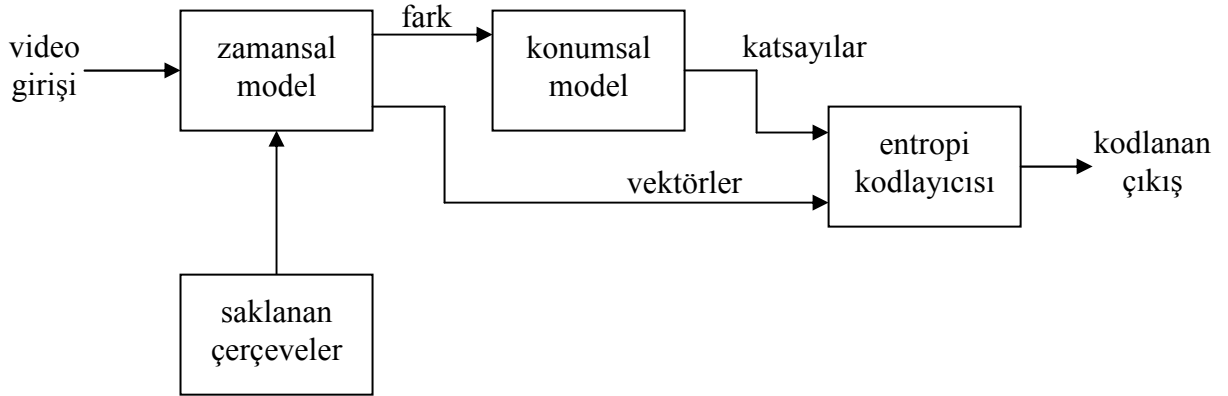
Videonun sıkıştırılması ilk çerçevesinin herhangi bir durağan görüntü sıkıştırma yöntemi kullanılarak kodlanmasıyla başlar. Sonraki her bir çerçevenin kendisi ve öncülü arasındaki farklar belirlenir ve çalışılan çerçeve bu farklar ile kodlanır. Eğer bir çerçeve öncülünden oldukça farklıysa, diğer çerçevelerden bağımsız bir şekilde kodlanmalıdır. Video sıkıştırma alanında, öncülü kullanılarak kodlanan çerçeveye ara çerçeve, bağımsız olarak kodlanan çerçeveye ise ana çerçeve denilmektedir. Şekil 3.2’de ana çerçeveler I harfi ile, ara çerçeveler ise P harfi ile etiketlenmiştir. Bir ara çerçeve öncüllerinden birisi kullanılarak kodlanabileceği gibi ardıllarından birisi de kullanılarak kodlanabilir. Hem önceki hem de sonraki çerçeveler kullanılarak kodlanan çerçevelere çiftyönlü çerçeveler denir ve B harfi ile etiketlenir (Salomon, 2002).

Bir çerçevenin ardılı kullanılarak kestirilmesi nesnenin hareketinin arkaplana ilişkin bir alanı aşamalı olarak açığa çıkardığı durumlarda anlamlıdır. Arkaplana ilişkin alan çalışılan çerçevede kısmen biliniyorken sonraki çerçevede daha iyi bilinebilir. Bu nedenle, sonraki çerçeve çalışılan çerçevedeki bu alanın kestirilmesi için doğal bir adaydır.



Şekil 3.3 CODEC (Richardson, 2003)

Şekil 3.3’de gösterilen video CODEC, kaynak görüntüsünü ya da video dizisini sıkıştırılmış biçimde kodlar ve bunu kaynak dizisinin aynısını ya da bir yaklaşımını elde etmek için çözer. Video sıkıştırmasında özgün video dizisi bir modelle (video verisinin bir yaklaşımının yeniden kurulmasında kullanılabilecek verimli kodlanmış tanımlama) nitelenir. Modelin video dizisini olanaklı en az bitle ve en iyi doğrulukla tanımlaması arzu edilir. Bu iki amaç (sıkıştırma verimliliği ve yüksek nitelik) genellikle çatışır, çünkü düşük bit hızıyla kodlama kodçözücüsünün düşük görüntü niteliğiyle video dizisini çözmesine neden olur (Richardson, 2003).



Şekil 3.4 Video kodlayıcısının blok diyagramı (Richardson, 2003)

Video CODEC üç ana işlevsel birimden oluşur: zamansal model, konumsal model ve entropi kodlayıcısı (Şekil 3.4). Zamansal modelin girişi sıkıştırılmamış video dizisidir.

Zamansal model çalışılan video çerçevesinin kestirimi ile, komşu video çerçeveleri arasındaki benzerliklerden yararlanarak zamansal artıklığı azaltmayı amaçlar. Hareket kestirimi videonun zamansal modelinin çıkarımında uygulanır. Zamansal kestirimin en yalın yöntemi çalışılan çerçevenin kestiriminde önceki çerçevenin kullanılmasıdır. Zamansal artıklıktan dolayı çalışılan çerçeveden önceki çerçeve çıkarıldığında benzer alanlar birbirini sönmüleyecektir. Görüntü çıkarma değişmez bir arkaplan boyunca hareket eden bir nesnenin

sezimi için de kullanılabilir. Görüntü çıkarma işlemi sonucunda elde edilen çerçeveye fark çerçevesi ya da yer değiştirmeli çerçeve farkı denir. Fark çerçevesinin kodlanması, çalışılan çerçevenin bağımsız olarak kodlanmasından çok daha etkin olarak gerçekleştirilir. Zamansal modelin çıkışı (kestirimin çalışılan çerçeveden çıkartılmasıyla oluşturulan) fark çerçevesi ve hareketin nasıl dengelendiğini belirten hareket vektörü kümesidir (Richardson, 2003).

Konumsal artıklığı azaltmak için fark çerçevesindeki komşu örnekler arasındaki benzerlikleri kullanan konumsal modelin girişini fark çerçeveleri oluşturur. MPEG-4 Görsel ve H.264 standartlarında kalan örneklerle bir dönüşüm uygulanır ve sonuçlar nicemlenir. Örnekler dönüşüm sonucunda örneklerin dönüşüm katsayıları ile temsil edildikleri başka bir alana geçirilir. Katsayılar fark çerçevesinin daha yoğun bir betimini sağlayan az sayıdaki anlamlı katsayıların elde edilmesi için nicemlenir. Konumsal modelin çıkışı nicemlenmiş dönüşüm katsayıları kümesidir (Richardson, 2003).

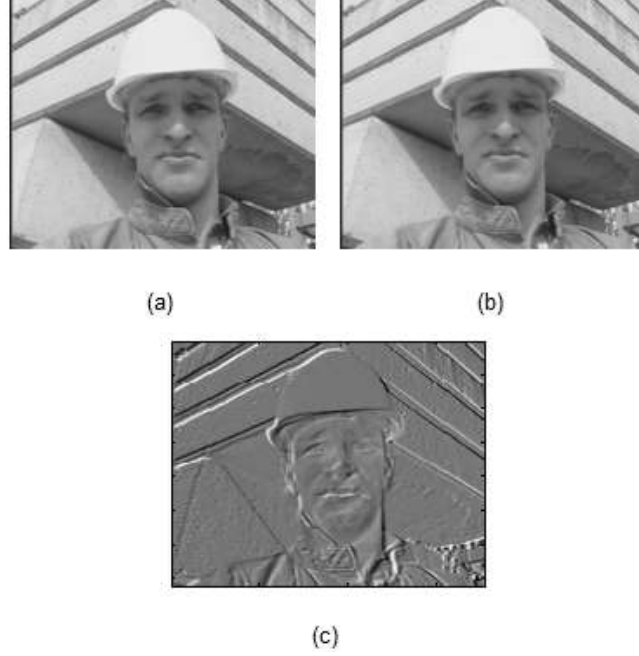
Zamansal modelin parametreleri (hareket vektörleri) ve konumsal model (katsayılar) entropi kodlayıcısı tarafından sıkıştırılırlar. Sıkıştırılmış video kodlanmış vektör parametrelerini, kodlanmış fark çerçevesi parametrelerini ve başlık bilgisini içerir (Richardson, 2003).

Video kodçözücüsü video çerçevelerini sıkıştırılmış bit dizisinden yeniden oluşturur. Katsayılar ve hareket vektörleri entropi kodçözücü tarafından çözüldükten sonra fark çerçevesini yeniden oluşturmak için konumsal model çözülür. Kodçözücü çalışılan çerçevenin bir kestirimini oluşturmak için bir ya da daha fazla sayıdaki önceden çözülmüş çerçevelerle hareket vektörü parametrelerini kullanır ve fark çerçevesi bu kestirime eklenerek çalışılan çerçeve yeniden oluşturulur (Richardson, 2003).

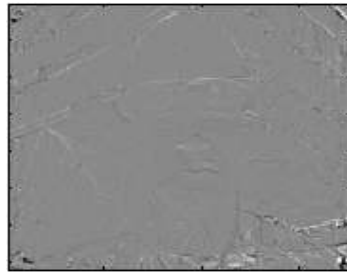
3.2.2 Hareket Dengeleme (Motion Compensation)

Video sıkıştırma uygulamalarında hareket kestirimi kullanılmaksızın kodlama yapılabilir. Ancak Şekil 3.5’de görüldüğü gibi fark çerçevesi hareket dengelemesi yapılmadan elde edildiğinde hareket eden nesnelerin sınırlarında önemli miktarda enerji kalır. Bu enerjinin hareket kestirimi ve dengelemesi kullanılarak azaltılması olanaklıdır. Çalışılan ve önceki çerçevelerde hareket halindeki bölgelerdeki pikseller arasında neredeyse hiç bir benzerlik yoktur, bu nedenle hareketli bölgelerde fark çerçevesindeki sinyal önemli seviyede artar. Sinyal gücü arttıkça veri hızı da artacaktır. En iyi sıkıştırma başarısı kodlanmış fark çerçevesinin ve kodlanmış ek bilginin boyutlarının en aza indirilmesiyle elde edilir. Kodlanmış fark çerçevesinin boyutu hareket dengelemesinden sonra fark çerçevesinde kalan

enerji ile bağlantılıdır. MPEG-4 Görsel ve H.264 standartlarında, kestirim bir ya da daha fazla sayıda öncül ya da ardıl çerçeveden elde edilir ve çerçeveler arasındaki farkların dengelenmesiyle iyileştirilir.



Şekil 3.5 Hareket dengelemesi yapılmadan elde edilen fark çerçevesinde kalan yüksek enerji. (a) Önceki çerçeve (b) Çalışılan çerçeve (c) Fark çerçevesinde, orta yeğlilikteki gri, farkın 0 olduğu, açık ya da koyu gri ise pozitif ve negatif farkları göstermektedir. Hareket dengelemesi olmaksızın elde edilen fark çerçevesinde kalan yüksek enerji açık ve koyu alanlarda görülmektedir. Bu ise zamansal kestirimden sonra hala sıkıştırılması gereken önemli miktarda bilgi olacağını söylemektedir.

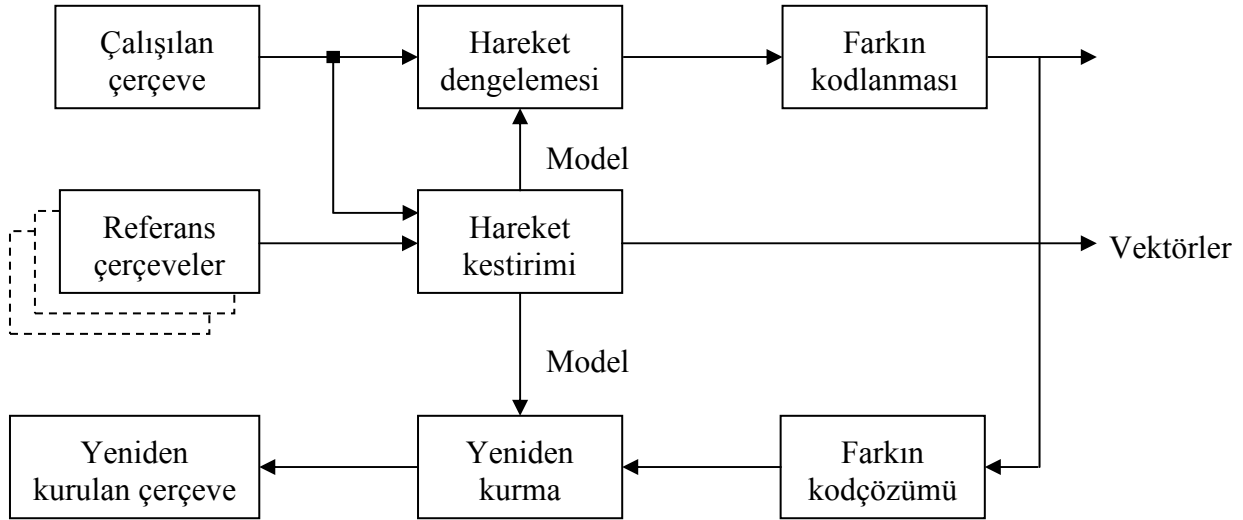


Şekil 3.6 Hareket kestirimi ve dengelemesinden sonra elde edilen fark çerçevesi

Video sıkıştırması için hareket kestirimi algoritması geliştirilirken çalışılan çerçevenin kabul edilebilir hesaplama karmaşıklığıyla en olanaklı doğrulukla modellenmesi (böylelikle daha iyi sıkıştırma başarımı elde edilir) hedeflenir.

Hareket kestirimi ve dengelemesinin etkisi Şekil 3.6'da gösterilmiştir. Hareket kestirimi ve

dengelemesinden sonra, referans çerçeve çalışılan çerçeveyle daha yakın eşleştirilebilmek için yeniden düzenlenmiştir. Şekil 3.6'da gösterilen hareket dengelemeli fark çerçevesi Şekil 3.5 (c)'deki hareket dengelemesiz fark çerçevesine göre daha az enerji içermektedir. Bu nedenle hareket dengelemeli fark çerçevesinin kodlanması sonucunda daha küçük boyutta kodlanmış çerçeve elde edilecektir.



Şekil 3.7 Hareket kestirimi ve dengelemesi blok diyagramı (Richardson, 2003)

Zamansal modelin hareket kestirimi ve dengelemesi ile hesaplandığı bir video sıkıştırma uygulamasının blok diyagramı Şekil 3.7'de gösterilmiştir.

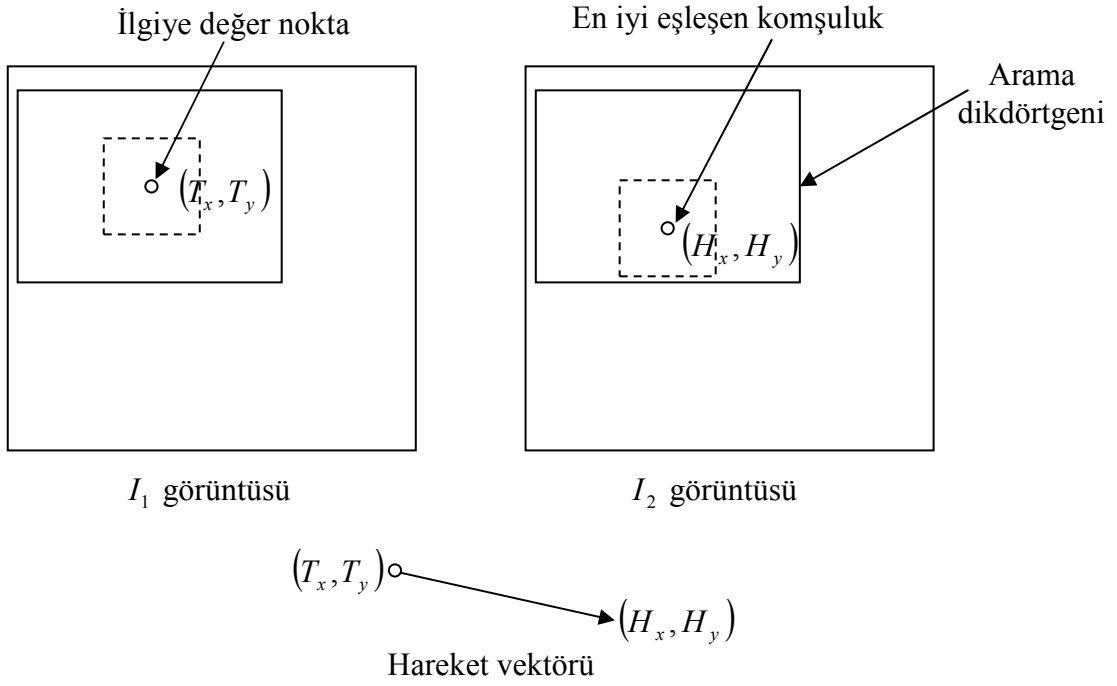
3.3 Seyrek Hareket Vektör Kümesi

3-boyutlu yapının ve hareket parametrelerinin çıkarılması için 2-boyutlu hareket vektörlerinin kullanıldığı bilgisayarla görme uygulamalarında, duyarlı özellik noktalarında hesaplanan seyrek 2-boyutlu hareket vektörü kümesi yeterli olabilir.

Seyrek hareket alanı t_1 ve $t_1 + \Delta$ zamanlarında alınan iki görüntüde karşılık gelen nokta çiftlerinin özdeşleştirilmesiyle hesaplanabilir. Kullanılan noktalar her iki görüntüde tanımlanmaları ve bulunabilmeleri için bir şekilde bilinmelidir. Köşe noktaların ya da ilgiye değer noktaların sezimi hem renkli hem de tek renkli görüntülerde geçerlidir.

İlgiliye değer noktalar kümesi $\{P_j\}$ t anında alınan I_1 görüntüsünde bulunduktan sonra,

$t_1 + \Delta$ anında alınan I_2 görüntüsündeki karşılıklı noktalarla özdeşleştirilmelidir. Noktaların I_2 görüntüsünden aynı biçimde araştırılması yerine I_2 görüntüsü I_1 görüntüsünden gelen her bir noktanın yeni konumunu bulmak için araştırılabilir. Bu araştırma karşılıklı korelasyon yöntemiyle gerçekleştirilebilir. I_1 görüntüsünden ilgiye değer bir P_j noktası alındığında, noktanın I_1 görüntüsündeki komşuluğu alınır ve hareketin miktarının kısıtlı olduğu varsayımıyla I_2 görüntüsündeki en çok ilişkili komşuluk bulunur. Şekil 3.8’de I_1 görüntüsündeki ilgiye değer (T_x, T_y) noktasının I_2 görüntüsündeki (H_x, H_y) merkezli komşulukla en iyi eşleştiği gösterilmiştir. (T_x, T_y) noktasından (H_x, H_y) noktasına düzlemsel öteleme miktarı hareket vektörünü verecektir (Shapiro ve Stockman, 2001).



Şekil 3.8 Seyrek noktalarda hareket vektörünün bulunması

3.4 Piksel Özyinelemeli Yöntemler

Piksel tabanlı hareket kestiriminde her bir piksel için bir hareket vektörünün belirlenmesi oldukça maliyetlidir. Video sıkıştırma uygulamaları için geliştirilen piksel özyinelemeli hareket kestirimi yöntemlerinde, hareket vektörleri özyinelemeli olarak elde edilir. İşlem yapılan bir pikselin hareket vektörü, daha önce kodlanan komşu piksellerin hareket vektörleri kullanılarak güncellenir. Kodçözücü aynı güncelleme kuralına uyarak aynı hareket vektörünü çıkarabilir, böylece hareket vektörlerinin kodlanmasına gerek kalmaz.

Piksel özyinelemeli yöntemler yalın olmalarına karşın, hareket kestirimi doğrulukları oldukça düşüktür; kestirim hatası büyüktür ve kodlama için önemli miktarda bit gerekir. Bu algoritmalar yalın oldukları gerekçesiyle eski kuşak CODEC'lerde kullanılmıştır. Günümüzdeki CODEC'ler, hareket vektörlerini belirtmek için kullanılan bit sayısı ve fark çerçevesi arasında daha iyi ödünleşim (tradeoff) sağlayan ayrıntılı hareket kestirimi yöntemlerini kullanmaktadır (Wang vd., 2002).

3.5 Blok Eşleştirme Algoritmaları (BEA)

BEA temelde çalışılan çerçevenin birbiriyle örtüşmeyen bloklara ayrılarak, her bir bloğun bir referans çerçevede belirli bir eşleştirme ölçütüne göre aranmasına dayanmaktadır. Kuramsal olarak bloklar herhangi bir çokgen şekle sahip olabilir ancak, uygulamada özellikle karesel bloklar kullanılmaktadır. BEA bir blok içerisindeki tüm piksellerin aynı düzlemsel öteleme hareketini yaptıkları varsayımına dayanmaktadır. Böylelikle hareket kestirimi sorunu her bir blok için bir hareket vektörünün bulunmasına dönüşür. Bloklar yeteri kadar küçükse, BEA doğru sonuçlar vermektedir. B_m m. blok, M blok sayısı ve $\beta = \{1,2,...M\}$ olmak üzere çalışılan çerçevenin bloklara ayrılması (3.1)'deki koşulu sağlamalıdır (Wang vd., 2002).

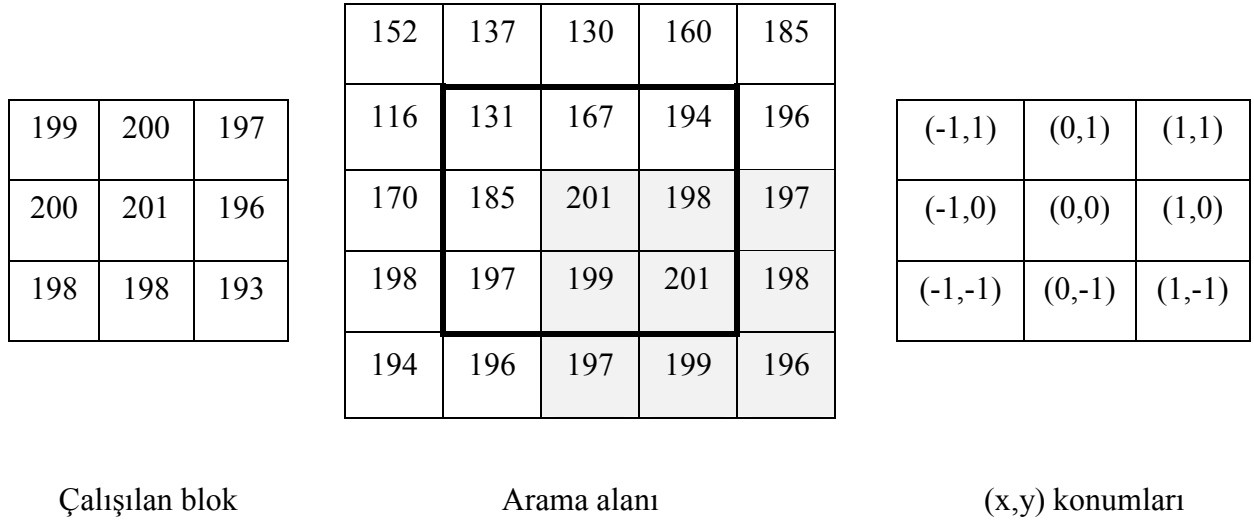
$$\bigcup_{m \in \beta} B_m = \Lambda, \text{ ve } B_m \cap B_n = \emptyset, m \neq n \quad (3.1)$$

BEA doğruluk ve karmaşıklık arasında iyi bir uzlaşma sağlar ve uygulamada çok büyük bir başarı kazanmıştır. Yaygın video kodlama standartlarında (H.261, H.263, MPEG-1, MPEG-2 ve MPEG-4) hareket kestirimi ve dengelemesi çalışılan çerçevedeki 8x8 ya da 16x16'lık bloklarla gerçekleştirilmektedir.

Blok seviyesindeki yaklaşımın sorunlarından biri komşu bloklar arasındaki hareket geçişlerine herhangi bir kısıtlama olmamasıdır. Bloktan bloğa gerçek hareket alanı düzgün değişse bile sonuçtaki hareket blok sınırları boyunca süresizdir. Bu sorun çalışılan çerçevenin birbirleriyle örtüşmeyen çokgenlere ayrıldığı bir yaklaşımla çözülebilir (Wang vd., 2002).

Çalışılan bloğun boyutları 16x16 olduğu varsayılırsa, hareket kestirimi algoritması referans çerçevedeki bir komşuluk bölgesinde çalışılan blokla eşleşen 16x16'lık bloğu bulmaya çalışır. En iyi eşleşme çalışılan blok ve eşleşen arasındaki fark enerjisini en aza indiren eşleşmedir. Referans çerçevedeki en iyi eşleşen bölgeyi bulmak için kuramsal olarak çalışılan bloğun referans çerçevede bulunan tüm bölgelerle karşılaştırılması gerekmektedir. Bu ise

karşılaştırma sayısının çok fazla olması nedeniyle uygulanabilir değildir. Bu nedenle eşleşen bloğun bulunması için yapılan arama çalışılan blok ile, referans çerçevedeki aynı konumu merkez alan bir komşuluk bölgesiyle sınırlı tutulur. Bunun yanı sıra çerçeveler arasındaki yüksek korelasyon nedeniyle eşleşmenin çalışılan bloğun yakın komşuluğunda bulunması daha yüksek olasılıklıdır.



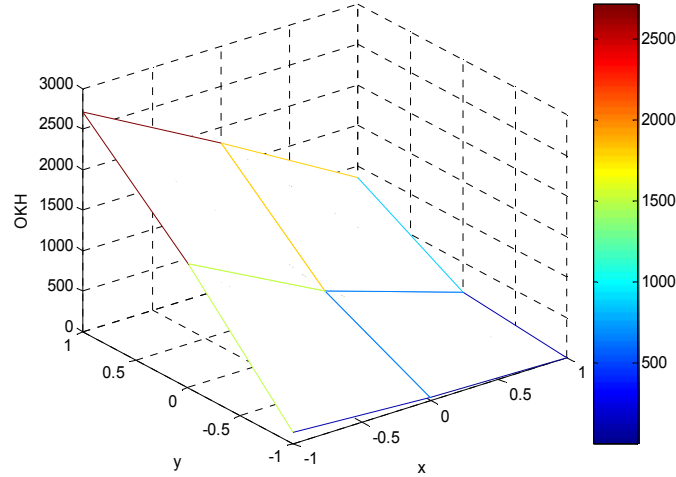
Şekil 3.9 Blok eşleştirme örneği

Blok eşleştirme işlemi Şekil 3.9’da gösterilmiştir. 3x3’lük çalışılan çerçeve referans çerçevede (merkezdeki kalın çizgi ile gösterilen) aynı konumdaki blokla ve (her bir yöndeki ± 1 piksel) konumundaki yakın komşu bloklarla karşılaştırılmaktadır. Çalışılan blok ve referans çerçevede aynı konumdaki ((0,0) konumundaki) blok arasındaki ortalama karesel hata (OKH) hesabı (3.2)’deki gibi yapılabilir.

$$\left\{ \begin{aligned} &(199 - 131)^2 + (200 - 167)^2 + (197 - 194)^2 + (200 - 185)^2 + \\ &((201 - 201)^2 + (196 - 198)^2 + (198 - 197)^2 + (198 - 199)^2 + (193 - 201)^2) \end{aligned} \right\} / 9 = 668.5 \quad (3.2)$$

Çizelge 3.1 Blok eşleştirme örneğindeki OKH değerleri

Blok konumu	(-1,1)	(0,1)	(1,1)	(-1,0)	(0,0)	(1,0)	(-1,-1)	(0,-1)	(1,-1)
OKH	2720.1	1804.1	852	1529.8	668.5	123	127.4	30.7	2.6



Şekil 3.10 Blok eşleştirme örneğindeki OKH yüzey grafiği

Her bir arama konumundan elde edilen OKH değerleri Çizelge 3.1’de listelenmiş ve Şekil 3.10’daki grafikte gösterilmiştir. Dokuz aday konum arasından (1,-1) en küçük OKH’yı vermektedir ve bu nedenle en iyi eşleşmedir. Bu örnekte, çalışılan blok için en iyi model (bir başka deyişle en iyi kestirim) (1,-1) konumundaki 3x3’lük bölgedir (Şekil 3.9’daki arama alanında açık gri tonlama ile gösterilen bölge). A çalışılan bloğun sol üst köşe koordinatlarını, B referans çerçevedeki eşleşen bloğun sol üst köşe koordinatlarını göstermek üzere çalışılan bloğa atanacak hareket vektörü (3.3)’deki gibi hesaplanır (Salomon, 2002).

$$(A_x - B_x, A_y - B_y) = (\Delta x, \Delta y) \quad (3.3)$$

Piksel seviyesinde değerlendirilirse çalışılan bloktaki tüm piksellere atanan hareket vektörü blok için belirlenen $(\Delta x, \Delta y)$ vektörü olacaktır. BEA nesnelerin ölçeklendirildiği ya da döndürüldüğü durumlarda değil, ötelendiği durumlarda etkilidir. Ayrıca çerçeveden çerçeveye, ışıklandırmadaki büyük değişiklikler yöntemlerin başarısını azaltır (Salomon, 2002).

3.5.1 Hareket Kestiriminde Kullanılan Eşleştirme Ölçütleri

$N \times N$ ’lik blok için OKH (3.4)’deki gibi hesaplanabilir:

$$OKH = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (3.4)$$

(3.4)'de C_{ij} çalışılan bloktaki bir pikseli, R_{ij} ise arama alanındaki bir pikseli göstermektedir. Buna göre C_{00}, R_{00} pikselleri sırasıyla çalışılan ve referans bloklardaki sol-üst köşedeki piksellerdir.

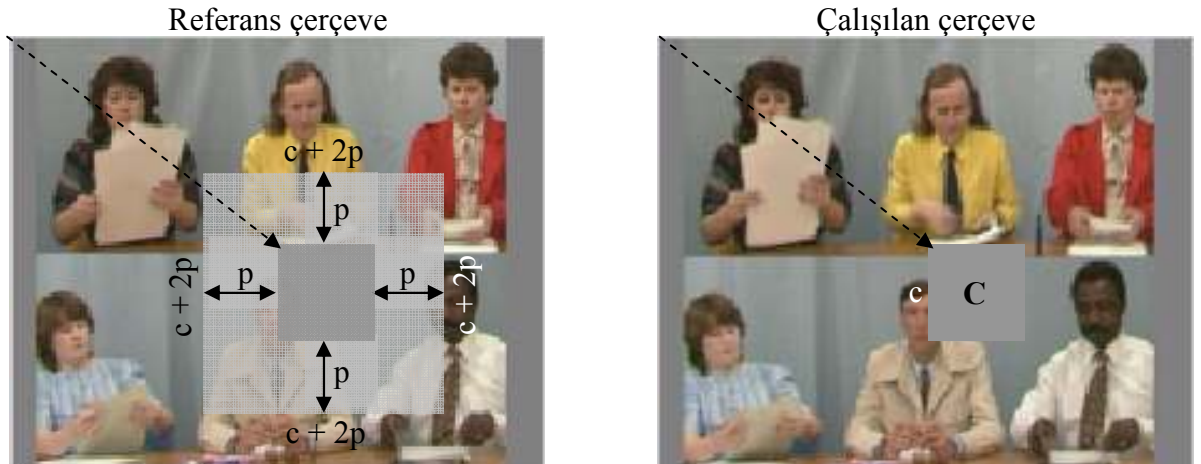
Ortalama mutlak fark (OMF) her piksel çifti için kare hesabı yerine büyüklük hesaplaması ile gerçekleştirildiği için OKH'ya göre daha kolay hesaplanır (3.5).

$$OMF = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (3.5)$$

Bloklar arasındaki karşılaştırma OMF eşitliğindeki $1/N^2$ terimi göz ardı edilerek ve mutlak farklar toplamı (MFT) ya da bir başka deyişle mutlak hatalar toplamı (MHT) hesaplanarak daha da yalınlaştırılabilir (3.6).

$$MFT = MHT = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (3.6)$$

MFT'nda bloklar arasındaki fark kabul edilebilir bir yaklaşıklıkla hesaplanabildiği için blok seviyesindeki hareket kestiriminde eşleştirme ölçütü olarak uygulamada yaygın olarak kullanılmaktadır.



Şekil 3.11 Referans çerçevede arama alanının oluşturulması

3.6 Tam Arama (TA)

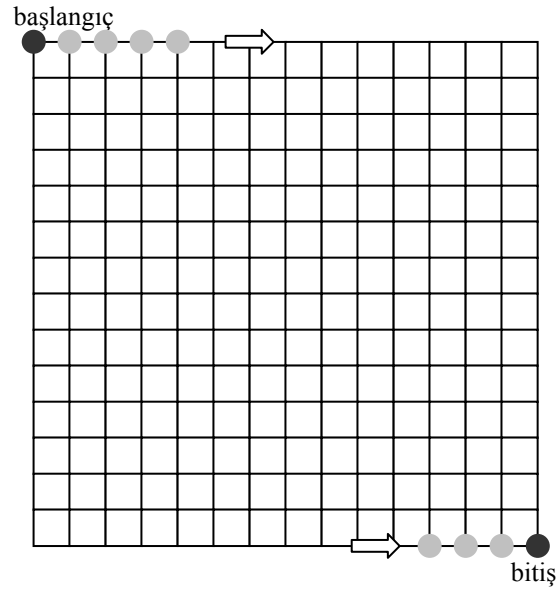
TA'da çalışılan blok referans çerçevede oluşturulan arama alanı içerisinde, karşılık gelen merkezdeki blok ve tüm komşu bloklarla karşılaştırılır. TA yalın, ancak hesaplama maliyeti yüksek bir algoritmadır. Arama alanı, çalışılan bloğa referans çerçevede konumsal olarak karşılık gelen bloğun dört kenarından da p piksel değeri kadar genişletilmesi ile elde edilir (Şekil 3.11). Hareket kestiriminde p piksel değerine arama parametresi (search parameter) denir.

Arama alanının en uygun boyutu pek çok etmene bağlıdır (Richardson, 2002): Her bir çerçevenin çözünürlüğü (daha yüksek çözünürlükler için daha büyük bir arama alanı uygundur), sahnenin yapısı (çok hareketli sahnelerde daha büyük bir arama alanının kullanılması az hareketli sahnelere göre daha uygundur) ve eldeki işlem gücü (daha büyük bir arama alanı daha fazla karşılaştırma ve bu nedenle daha fazla işlem gerektirir).

Çalışılan C bloğu bir ayrıtı c uzunluğunda kare olarak alınırsa arama penceresi $(c + 2p)^2$ pikselden oluşacaktır. TA'da çalışılan blok arama penceresindeki $(2p + 1)^2$ tane ayrı, birbiriyle kısmen örtüşen aday bloklarla karşılaştırılır ve belirlenen eşleştirme ölçütüne göre en iyi eşleşen referans blok seçilir.

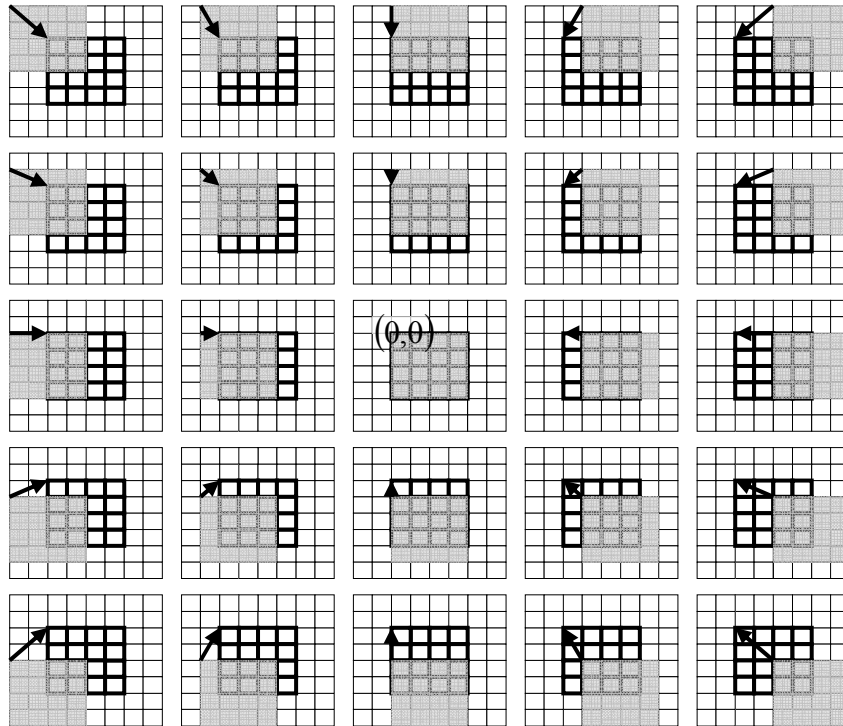
Çalışılan blok ve referans çerçevede arama yapılan bloktaki karşılıklı pikseller başına bir işlemin bir çıkarma, bir mutlak değer hesabı ve bir toplamadan oluştuğu düşünülürse MFT için her bir aday blok başına c^2 tane işlem yapılacaktır. Bu durumda çalışılan çerçevedeki bir bloğa hareket vektörü atanması için gerekli işlem sayısı $(2p + 1)^2 c^2$ 'dir. $M \times M$ boyutlarındaki bir çerçevede (M 'nin, blok boyutunun tam katı olduğu varsayımıyla) $(M / c)^2$ blok vardır. Sonuç olarak çalışılan çerçevedeki tüm blokların hareket kestirimi için gerekli işlem sayısı $M^2 (2p + 1)^2$ 'dir. Burada toplam hesaplama karmaşıklığının c blok boyutundan bağımsız olduğuna dikkat edilmelidir (Wang vd., 2002). Örneğin, $M = 256$, $c = 8$ ve $p = 7$ alınırsa çerçeve başına toplam işlem sayısı 1.47×10^7 'dir. Çerçeve hızı 30 olan bir video için, saniye başına işlem sayısı ise 4.42×10^8 'dir. Bu örnek TA'nın ne denli yoğun bir hesaplama yükü getirdiğini göstermektedir. Bu sorun nedeniyle, düşük hesaplama karmaşıklığına karşılık kestirim doğruluğundan ödün veren ve Bölüm 3.7'de ele alınan hızlı BEA geliştirilmiştir.

Kestirimin doğruluğu yatay ya da düşey doğrultuda iki referans bloğu arasındaki uzaklık olan arama adım boyuyla belirlenir. Genel olarak, yatayda ve düşeyde aynı adım boyu kullanılır.



Şekil 3.12 Arama alanında satır düzeninde arama

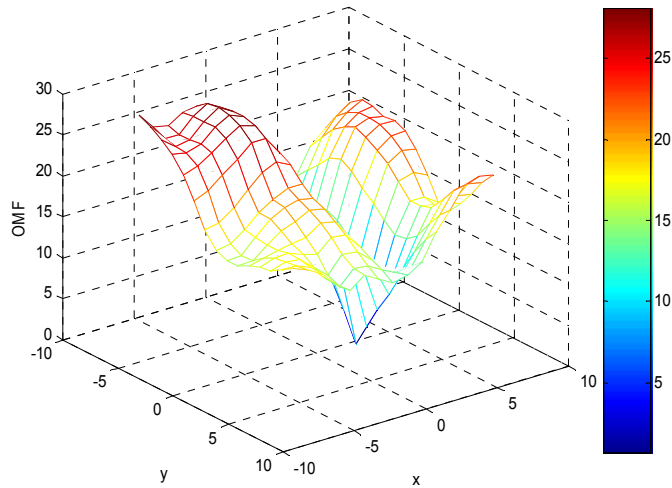
Uygulamada adım boyu bir piksel alınabilir. TA satır satır yapıldığında, arama alanının sol üst köşesindeki blokla başlanır ve arama işlemi yatayda ve düşeyde arama adım boyu kadarlık yer değiştirmelerle sürdürülür. Şekil 3.12'deki kafesin kesim noktaları arama alanındaki blok konumlarını göstermektedir ve arama alanındaki satır araması oklarla gösterilen biçimdedir.



Şekil 3.13 Arama sonucunda atanacak hareket vektörleri

Çalışılan blokla referans çerçevede eşleşen blok belirlendikten sonra çalışılan bloğa atanacak hareket vektörü arama penceresindeki tüm blok konumları için Şekil 3.13’de gösterilmiştir. Şekil 3.13’de çalışılan blok boyutları 4×4 olarak alınmıştır. Arama parametresi $p = 2$ alındığından arama alanı $(4 + 2 \times 2)^2 = 64$ pikselden oluşmaktadır ve arama alanında $(2 \times 2 + 1)^2 = 25$ tane birbiriyle kısmen örtüşen aday blok vardır.

BEA gerçekleştirilirken her bir çerçeve görüntünün yüksekliği ve genişliği ile sırasıyla aynı satır ve sütun sayısına sahip bir matrisle temsil edilir. Çalışılan çerçevenin birbiriyle örtüşmeyen bloklara ayrılmasıyla görüntü blok boyutlarındaki alt matrislerden oluşur. Çalışılan blok referans çerçevedeki karşılık gelen blokla eşleştirildiğinde hareket vektörünün belirlenebilmesi için her iki bloğun sol-üst köşelerinin satır ve sütun değerleri birbirinden çıkarılır. Bu nedenle Şekil 3.13’de çalışılan bloğa atanacak hareket vektörleri gösterilirken vektörler blokların merkezlerinden değil sol-üst köşelerinden çizilmiştir.



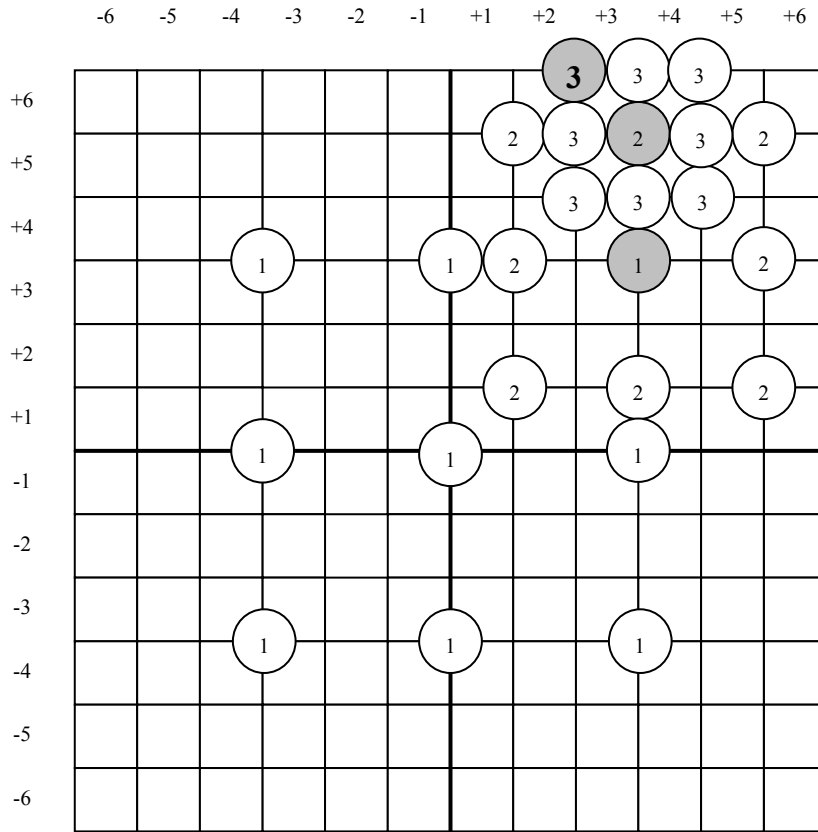
Şekil 3.14 OMF değerlerinin oluşturduğu hata yüzeyi

3.7 Hızlı BEA

Tüm hızlı BEA’nda temel amaç TA’nın hesaplama yükünü arama alanındaki aday blok sayısını azaltarak hafifletmektir. Hızlı bir blok eşleştirme algoritmasında, arama alanında az sayıdaki aday blokla karşılaştırma yapılarak minimum OMF bulunmaya çalışılır.

Çalışılan bloğun referans çerçevedeki aday bloklarla karşılaştırılmasından elde edilecek OMF değerleri bir hata yüzeyi meydana getirir. $p = 7$ olarak alınırsa elde edilecek tipik bir hata

yüzeyi Şekil 3.14’de verilmiştir. TA’da arama alanındaki tüm aday bloklar incelendiği için hata yüzeyindeki global minimum mutlaka bulunur. Ne var ki, hızlı BEA’nda olduğu gibi arama alanındaki yalnızca bazı aday blokların incelendiği yöntemlerde global minimumun yerine yerel minimumun bulunması gibi bir hataya düşülebilir. Bu durumun, örneğin, video sıkıştırmasına ilişkin şöyle bir sakıncası vardır: Hızlı BEA’nda, çalışılan blok ve onunla en iyi eşleşen referans blok arasındaki fark enerjisini en aza indiren global minimum yerine yerel minimumun bulunduğu durumda fark enerjisi en küçük değeri almayacak ve böylece video kodlayıcısının fark bloğunu kodlamak için üreteceği kodlanmış bit sayısı daha fazla olacaktır. Bu nedenle, hızlı BEA kullanıldığında genellikle TA’ya göre daha yetersiz bir sıkıştırma başarımına ulaşılır.



Şekil 3.15 Üç adımda arama. Blok örüntüsündeki numaralar algoritmanın adımlarını göstermektedir (Koga vd., 1981)

3.7.1 Üç Adımda Arama (ÜAA)

ÜAA p arama parametresinden hareketle hesaplanan adım boyuyla başlar. Başlangıçtaki adım boyutu A_{ilk} (3.7)’deki gibi hesaplanabilir.

$$A_{ilk} = 2 \times (\lfloor \log(p+1) / \log 2 \rfloor) - 1 \quad (3.7)$$

$p = 7$ alındığında adım boyu $A = 4$ olarak belirlenir ve arama merkezden başlatılır. İlk yapılır. İlk adımda aranılan dokuz konumdaki en az hatayı veren blok seçilerek yeni arama merkezi olarak belirlenir. Adım sayısı $A = A/2$ biçiminde güncellenerek $A = 1$ olana dek benzer biçimde arama işlemi yinelenir. $A = 1$ olduğu durumda en küçük hatayı veren konum bulunur ve bu konumdaki blok en iyi eşleşen blok olarak belirlenir. Her adımda belirlenen merkez etrafındaki sekiz arama konumu, arama alanı içerisindeki dört köşe noktası ve bu köşeleri birleştiren ayrıtların orta noktalarıdır. Buna göre arama parametresi ± 7 olan arama alanı içerisinde TA'daki 225 aday blok yerine yalnızca 25 ($9+8+8$) aday blok aranır.

Koga vd. (1981) arama parametresini ± 6 piksel ve başlangıç adım boyunu 3 alarak aramaya başlamış ve her adımdan sonra adım boyunu 1 eksiltmiştir. Böylece algoritma 3 adım sonra durmaktadır. Görüldüğü gibi adım sayısının her adımdan sonra nasıl belirleneceği çeşitli biçimlerde yapılabilir. Şekil 3.15'de, ilk adımda $(0,0)$ konumundaki blok ve orijin çevresindeki ± 3 piksel uzaklıktaki 8 blok aranmaktadır. İkinci adımda ise ilk adımda bulunan en iyi eşleşen blok çevresindeki ± 3 piksel uzaklıktaki 8 blok ve üçüncü adımda, ikinci adımda bulunan en iyi eşleşen blok çevresindeki ± 1 piksel uzaklıktaki 8 blok karşılaştırılmaktadır. Her üç adımda bulunan en iyi eşleşen bloklar gri tonlama ile gösterilmiştir. Üçüncü adımdaki en iyi eşleşme arama algoritmasının sonucu olarak seçilir.

ÜAA algoritmasında sabit, kestirilebilir sayıda arama adımı ve arama noktası söz konusudur. Ayrıca daha düzenli bir yapıya sahiptir. ÜAA yönteminin bu özellikleri onu donanımsal gerçekleştirmeler için diğer bazı hızlı hareket kestirimi yöntemlerine göre daha uygun hale getirmektedir.

3.7.2 Yeni Üç Adımda Arama (YÜAA)

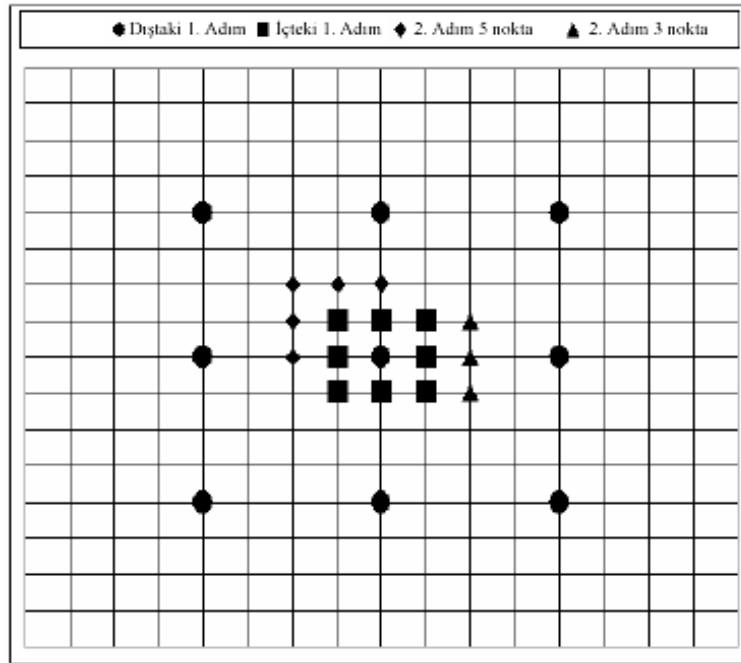
Hızlı çalışan BEA'nın ana hedefi TA algoritmasının gerektirdiği sabit arama noktaları sayısının büyük ölçüde azaltıldığı en uygun ya da en uyguna yakın bir çözümün bulunmasıdır. Arama alanı içerisinde çok sayıda yerel minimum noktası bulunduğundan küçük bir arama örüntüsü ile arama yapılması yerel minimum noktasına yakalanmaya neden olacak ve bu nedenle özellikle büyük hareket içeriğinin olduğu görüntü dizilerinde başarıyı düşürecektir.

Öte yandan, ÜAA'da olduğu gibi dokuz seyrek karşılaştırma noktasının bulunduğu büyük bir arama örüntüsünün kullanılması arama yolunu olasılıkla yanlış bir yöne kaydıracak ve bu

nedenle en uygun nokta kaçırılacaktır. Gerçek dünyadaki videolarda global minimum noktalarının dağılımı orijinde yoğunlaştığından, YÜAA merkez-yanlı bir ÜAA algoritması olarak geliştirilmiştir (Reoxiang vd., 1994).

ÜAA ilk adımda, durağan ya da neredeyse-durağan bloklarda gözlemlenen küçük hareketleri belirlemekte pek etkin olmayan eşbiçimli bölüştürülmüş bir arama örüntüsü kullanır. Bu sorunu düzeltmek ve aramayı hareket ölçeğine ve belirsizliğine daha uyarlanabilir yapmak için pek çok uyarlamalı teknik önerilmiştir. YÜAA'da arama örüntüsü her adımda sabittir ve bu algoritmada eşikleme işlemleri kullanılmaz. Genel olarak bakıldığında YÜAA yöntemi özgün ÜAA yönteminin yalınlığını ve genelliğini korumakta, hareket dengeleme hatası ve sağlamlığı bakımından ÜAA'dan daha iyi çalışmaktadır ve hesaplama karmaşıklığı bakımından ÜAA ile oldukça uyumludur (Reoxiang vd., 1994).

Pek çok hızlı blok eşleştirme algoritması şu varsayıma dayanmaktadır: Karşılaştırma noktaları global minimumdan uzaklaştıkça hata fonksiyonu tekdüze artar. Bu varsayıma göre, arama alanına ilişkin hata yüzeyi tek bir tepe noktasına sahip olmalıdır. Ne yazık ki bu varsayım dokusal yerel görüntü içeriği, hareketli nesne ve arkaplanın tutarsız blok bölütlenmesi, çerçeveler arasında ışıklılığın değişmesi, vb. nedenlerden dolayı doğru değildir. Sonuç olarak, arama kolaylıkla bir yerel minimumla sonuçlanacaktır (Reoxiang vd., 1994).



Şekil 3.16 Yeni üç adımda arama (Reoxiang vd., 1994)

ÜAA gibi çok adımlı bir arama yordamında ilk adımda denetim noktaları arama alanında

eşbiçimli bölüştürülmüştür. Böyle bir düzenleme küçük hareketlerin gözlemlendiği bazı bloklar için çok uygun olmayabilir ve bu nedenle ilk adım için karşılaştırma noktası örüntüsünün en uygun nasıl tasarlandığı önemlidir. Gerçek dünyadaki bir görüntü dizisinin blok hareket alanının düzgün olduğu ve yavaş değiştiği bilinen bir gerçektir. Global minimum dağılımı genel olarak düzgün dağılımlı değil, merkez-yanlıdır.

YÜAA iki bakımdan ÜAA'dan ayrılmaktadır:

- İlk adımında merkez-yanlı bir karşılaştırma örüntüsü kullanır
- Durağan ya da neredeyse-durağan bloklar için yarıyolda-durma yöntemi uygular

Şekil 3.16'da gösterilen YÜAA algoritması aşağıdaki gibidir (Reoxiang vd., 1994).

- İlk adımda, ÜAA'da kullanılan özgün denetim noktalarına ek olarak arama alanı merkezinin sekiz komşusu olan fazladan sekiz nokta eklenmiştir.
- Durağan ya da neredeyse-durağan blokları çabucak belirlemek ve kestirim yapmak için yarıyolda-durma yöntemi kullanılır:

Eğer ilk adımda minimum hata arama alanının merkezinde bulunursa arama durdurulur (Bu ilk-adımda-durma olarak adlandırılır).

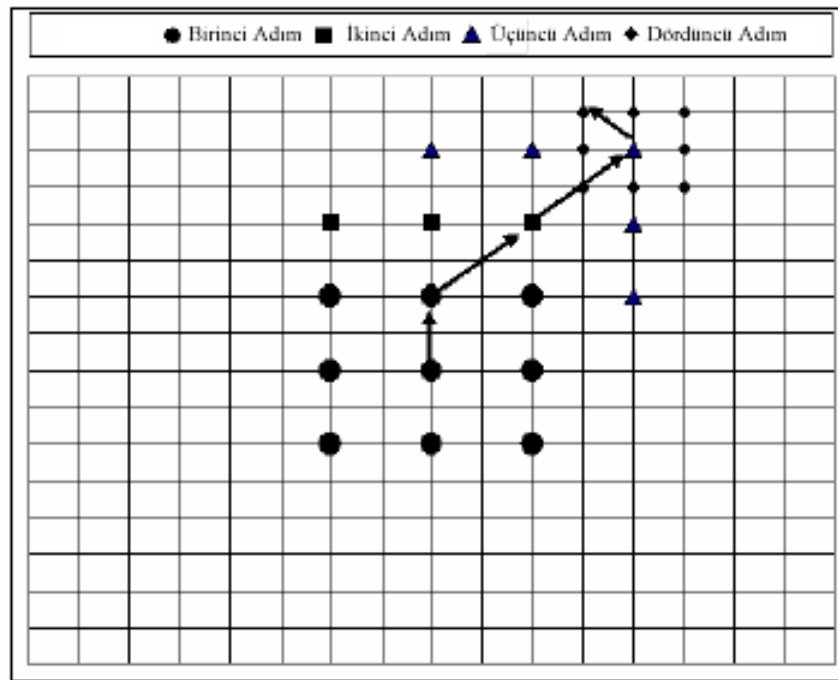
Eğer ilk adımda minimum hata noktası arama penceresinin merkezinin sekiz komşusundan birisi ise, arama örüntüsüne yeni arama noktaları eklenir. Merkezin sekiz komşuluğundaki minimum hataya sahip nokta köşedeysse 5 tane ek karşılaştırma noktası, arama örüntüsünün yatay ya da dikey eksenin ortasında konumlanmışsa 3 tane ek karşılaştırma noktası kullanılır (Şekil 3.16). İkinci adımdaki arama yalnızca minimumun komşusu olan sekiz noktada yapılır (bu sekiz noktanın bazılarının ilk adımda zaten denetlendiğine dikkat edilmelidir) ve arama durdurulur (Bu ise ikinci-adımda-durma olarak adlandırılır).

Minimum hata noktası ne arama alanının merkezi ne de merkezin sekiz komşusundan birisi ise çalışılan blok için hareket vektörünün belirlenmesi için tam bir ÜAA yordamı uygulanır.

Bu yöntemde her blok için en az 17 nokta ya da en kötü durumda 33 noktanın denetlenmesi gerekebilir. İlk-adımda-durma gerçekleşirse 17 karşılaştırma yapılmış olur ve hareket vektörü olarak (0,0) atanır. İkinci-adımda-durma gerçekleşirse arama alanı merkezinin sekiz komşusundan hangisinin minimum hatayı verdiğine bağlı olarak toplamda 20 ya da 22 nokta denetlenir. Eğer algoritmanın akışı ÜAA'da olduğu gibi sürerse en kötü durum olan 33 nokta denetlenmelidir.

3.7.3 Dört Adımda Arama (DAA)

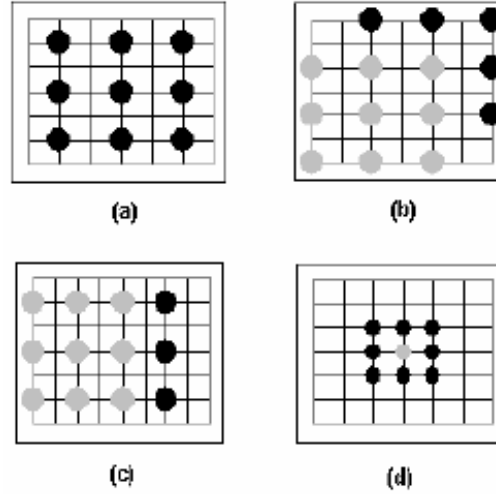
$p = 7$ için ÜAA algoritması yalnızca 25 blok karşılaştırmasına gereksinim duyarken, YÜAA algoritması en kötü durumda 33 blok karşılaştırması gerektirmektedir. Büyük hareketlerin çok olduğu bazı videolar için, YÜAA'nın hesaplama karmaşıklığı ÜAA'dan yüksek olabilir. Buna ek olarak, hareket kestiriminin gerçek-zamanlı ya da donanım gerçekleştirmesinde, ortalama hesaplama yerine en kötü durumdaki hesaplama gereksinimi göz önüne alınmalıdır. DAA, ÜAA'dan daha iyi başarımlar gösterirken YÜAA ile karşılaştırıldığında benzer bir başarıma sahiptir. DAA'nın hesaplama gereksinimi en kötü durumda ÜAA ile karşılaştırıldığında yalnızca iki blok daha fazla olan 27 blok eşleştirmesidir (Po ve Ma, 1996).



Şekil 3.17 Dört adımda arama (Po ve Ma, 1996)

DAA ilk adımda merkez blok ve çevresindeki ± 2 piksel uzaklıktaki sekiz konumdaki toplam 9 karşılaştırma noktasının olduğu merkez-yanlı bir arama örüntüsü kullanır. Arama alanının merkezi daha sonra minimum hatayı veren noktaya kaydırılır. Sonraki iki adımın arama örüntüsü minimum hatayı veren noktaların konumlarına bağlıdır. Minimum hatayı veren nokta arama örüntüsünün merkeziyse, arama 3x3'lük bir arama penceresinin oluşturulduğu son adıma (adım 4) gidecektir. Aksi durumda, adım 2 ve adım 3 için arama merkezinden ± 2 piksel uzaklıktaki arama örüntüsü korunur. Son adımda, bir önceki adımda seçilen merkez bloğun sekiz komşusunun oluşturduğu arama örüntüsünde işlem sonlanır. Şekil 3.17'de gösterilen DAA algoritması aşağıdaki gibi özetlenebilir (Po ve Ma, 1996).

Adım 1: Arama alanının merkezindeki blok ve ± 2 piksel uzaklıktaki bloklardan oluşan dokuz denetleme noktasından minimum hatayı veren nokta hesaplanır. Minimum hatayı veren nokta arama örüntüsünün merkezindeyse adım 4'e geçilir, yoksa adım 2'den devam edilir.



Şekil 3.18 DAA'da kullanılan arama örüntüleri (Po ve Ma, 1996). (a) İlk adım (b) İkinci/üçüncü adım (c) İkinci/üçüncü adım (d) Dördüncü adım

Adım 2: Merkez blok çevresindeki ± 2 piksel uzaklıktaki blok konumları korunur. Ancak arama örüntüsü minimum hatayı veren önceki noktanın konumuna bağlıdır.

- Minimum hatayı veren nokta önceki arama örüntüsünün köşesinde ise, Şekil 3.18(b)'de gösterildiği gibi 5 tane ek karşılaştırma noktası kullanılır.
- Minimum hatayı veren nokta önceki arama örüntüsünün yatay ya da düşey ekseninin ortasında konumlanmışsa Şekil 3.18(c)'de gösterildiği gibi 3 tane ek karşılaştırma noktası kullanılır.

Minimum hatayı veren nokta arama örüntüsünün merkezindeyse adım 4'e geçilir, yoksa adım 3'ten devam edilir.

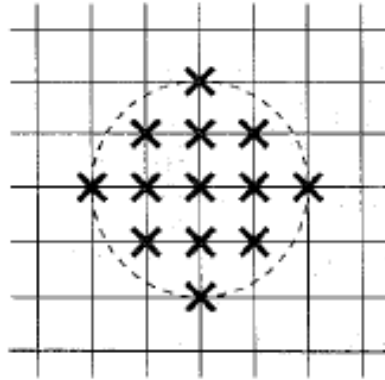
Adım 3: İkinci adımda olduğu gibi gerçekleştirilir, ancak en sonunda adım 4'e geçilir.

Adım 4: Şekil 3.18(d)'de gösterildiği gibi arama merkezinin çevresinde ± 1 piksellik bir arama örüntüsü oluşturulur ve sonuç olarak hareket vektörünün yönü bu dokuz arama noktasından minimum hataya sahip nokta olarak belirlenir.

Algoritmadan, herhangi bir adımda, minimum hataya sahip nokta arama örüntüsünün merkezinde konumlanmışsa DAA'nın ara adımlarının atlanabileceği ve ± 1 piksellik arama örüntüsünün kullanıldığı son adıma geçilebileceği sonucuna varılabilir. ± 2 piksellik arama örüntüsünde birbiriyle örtüşen karşılaştırma noktaları bulunmaktadır, bu nedenle karşılaştırılan noktaların toplam sayısı $9 + 8 = 17$ ile $(9 + 5 + 5 + 8) = 27$ arasında değişmektedir. DAA'nın en kötü durumdaki hesaplama gereksinimi 27 blok karşılaştırmasıdır ve bu durum büyük hareketlerin kestiriminde karşımıza çıkar. Buna göre DAA'nın hesaplama karmaşıklığı en kötü durumda ÜAA'dan yalnızca 2 blok karşılaştırması fazla, YÜAA'dan 6 blok karşılaştırması azdır (Po ve Ma, 1996).

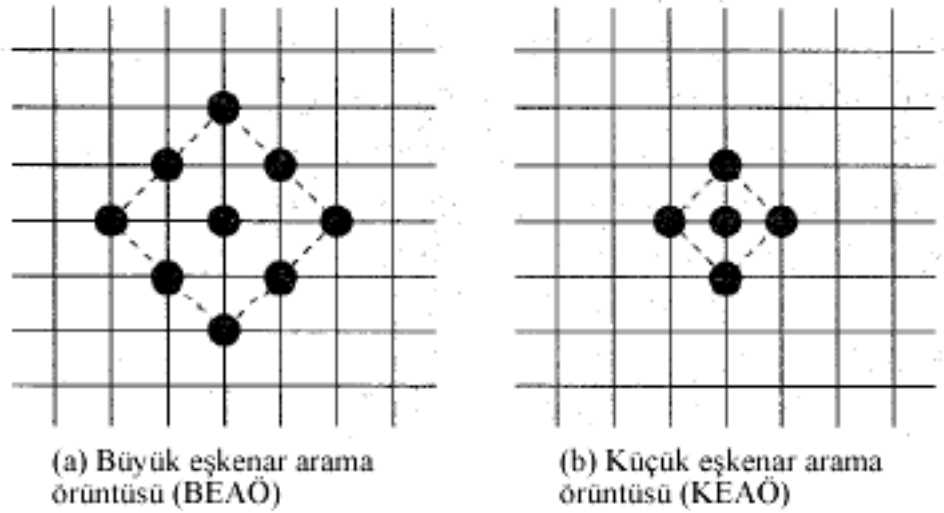
3.7.4 Eşkenar Arama (EA)

± 2 piksellik sabit boyutlu uygun bir arama örüntüsü kullanarak, DAA YÜAA'ya benzer bir başarımlı göstermektedir ve ÜAA'dan üstündür. Ancak DAA hala durağan bir blok için 17 denetim noktasına bakılmasını gerektirmektedir (Zhu ve Ma, 2000).



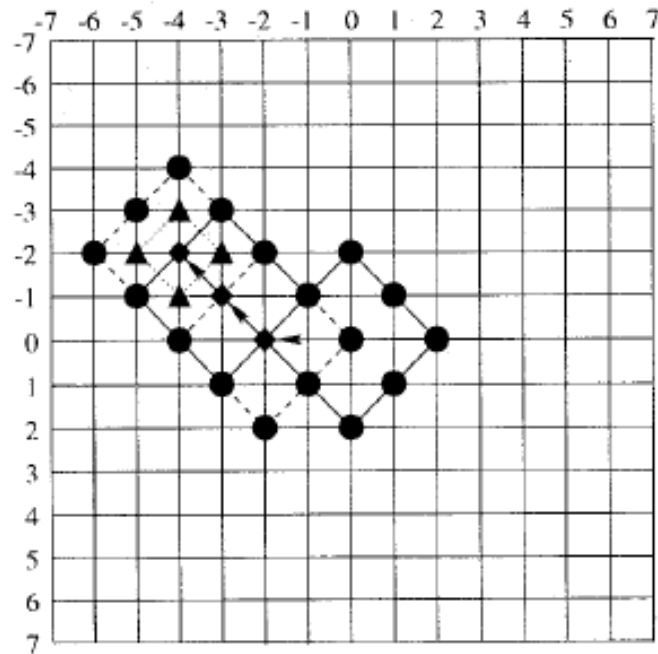
Şekil 3.19 Uygun bir arama örüntü desteği (Zhu ve Ma, 2000). – 2 piksel yarıçaplı dairesel alan. 13 tane “x” daire içerisindeki tüm olanaklı denetim noktalarının konumlarını göstermektedir

Herhangi bir doğrultuda blok yer değiştirmesinin gerçek videolarda eşit dağılımlı olduğu varsayılabilir. Bu gözleme dayanarak, 2 piksel yarıçaplı bir daire içerisindeki arama noktalarının konumlarının (Şekil 3.19'da “x” ile gösterilmiştir) arama örüntüsü oluşturmak için en uygun oldukları söylenebilir (Zhu ve Ma, 2000).



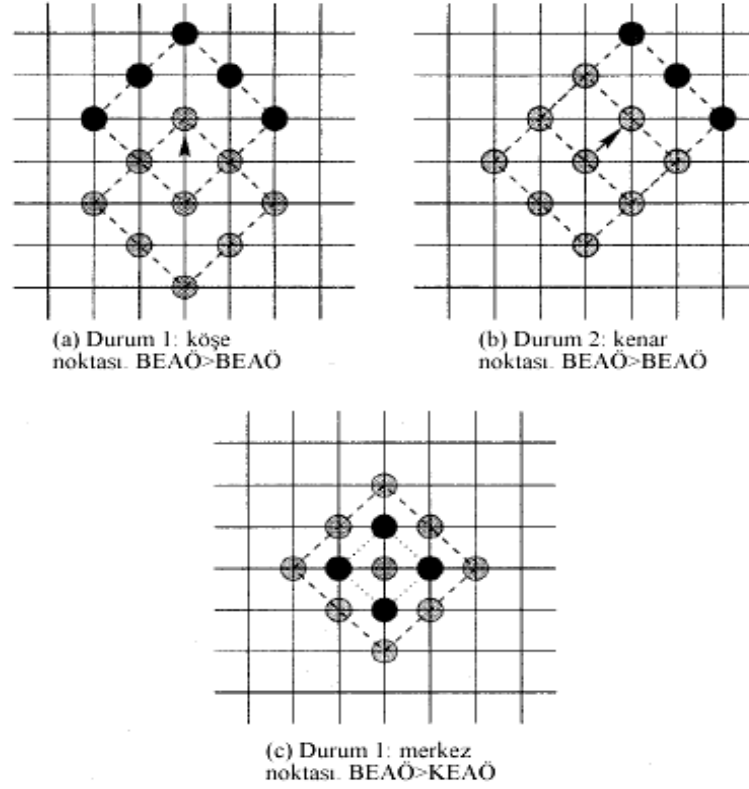
Şekil 3.20 EA’da kullanılan arama örüntüleri (Zhu ve Ma, 2000)

EA algoritmasında Şekil 3.20’de gösterildiği gibi Şekil 3.19’deki “x” konumlarından türetilen iki arama örüntüsü kullanılmaktadır. Büyük eşkenar arama örüntüsü (BEAÖ) olarak adlandırılan ilk örüntü, eşkenar dörtgen biçimli, merkezdeki bir noktanın sekiz nokta ile çevrelendiği 9 denetleme noktasından oluşmaktadır. Küçük eşkenar arama örüntüsü (KEAÖ) olarak adlandırılan ikinci örüntü ise, daha küçük bir eşkenar dörtgen şekli oluşturan 5 denetleme noktasına sahiptir.



Şekil 3.21 Eşkenar arama (Zhu ve Ma, 2000). İşlem sonucunda (-4,-2) hareket vektörü elde ediliyor. Ardışıl olarak, her adımda 9, 5, 3, 3, ve 4 arama noktasından oluşan toplam 24 arama noktası karşılaştırılıyor.

Şekil 3.21’de gösterildiği gibi, ilk olarak BEAÖ merkezdeki noktada minimum hata elde edilene kadar yinelemeli olarak uygulanır. Son arama aşamasına ulaşıldığında arama örüntüsü olarak BEAÖ yerine KEAÖ kullanılır. KEAÖ’deki beş denetleme noktasından en küçük hatayı veren konumdaki blok çalışılan blokla en iyi eşleşen blok olarak belirlenir.



Şekil 3.22 EA’da kısmen örtüşen denetleme noktaları (Zhu ve Ma, 2000). Önceki arama adımında bulunan (gri noktalar) minimum hataya sahip nokta (a) köşe noktalardan birinde, (b) kenar noktalardan birinde ve (c) merkez noktada konumlandığında BEAÖ’ndeki denetleme noktalarının örtüşmesi gösterilmektedir. Siyah noktalar o anki adım için blok eşleştirme ölçütünün hesaplanmasının gerektiği yeni denetim noktalarıdır.

Özellikle BEAÖ sürekli olarak uygulanırken, ardışıl iki adım arasında kısmen örtüşen denetleme noktalarının bulunduğu dikkat edilmelidir. Denetleme noktalarının örtüştüğü üç durum Şekil 3.22’de gösterilmiştir. Önceki minimum nokta Şekil 3.22(a) ve 3.22(b)’de gösterildiği gibi BEAÖ’nün kenarlarından ya da köşe noktalarından birinde konumlanmışsa, yalnızca 5 ya da 3 yeni denetleme noktasının sınanması gerekmektedir. BEAÖ’nün merkez noktası minimum hatayı verirse arama örüntüsü olarak BEAÖ yerine KEAÖ kullanılır ve Şekil 3.22(c)’de gösterildiği gibi yalnızca 4 yeni noktanın sınanması gerekir (Zhu ve Ma, 2000).

EA algoritması daha küçük hareket kestirimi hatası ile hem az hem de çok hareketli videolarda kullanılabilir. Ayrıca YÜAA, DAA ve ÜAA ile karşılaştırıldığında daha az sayıda arama noktası gerektirmektedir.

3.8 Hareket Kestirimi Algoritmalarının Karşılaştırılması

Hareket kestirimi algoritmaları genellikle, yaygın olarak kullanılan video görüntüleri ile değerlendirilmektedir. Bu video görüntüleri birbirlerinden sabit, yavaş ya da hızlı hareket eden nesnelerin bulunması, kameranın farklı eksenlerdeki hareketleri gibi ayırt edici özellikler ile ayrılmaktadır.



Şekil 3.23 Hareket kestirimi yöntemlerinin karşılaştırılmasında kullanılan video görüntüleri. (a) Tüm yönlerde çok hareketin ve dikkate değer kamera hareketinin olduğu görüntü dizisi (b) Hareketin yalnızca genel olarak değişmez arkaplandaki dans gösterisinde ve önplandaki sunucuların yüzlerinde sınırlandığı, görece az hareketli görüntü dizisi (c) Arkaplanı karmaşık olan ve görece az hareketin konuşmacıdan kaynaklandığı video dizisi.

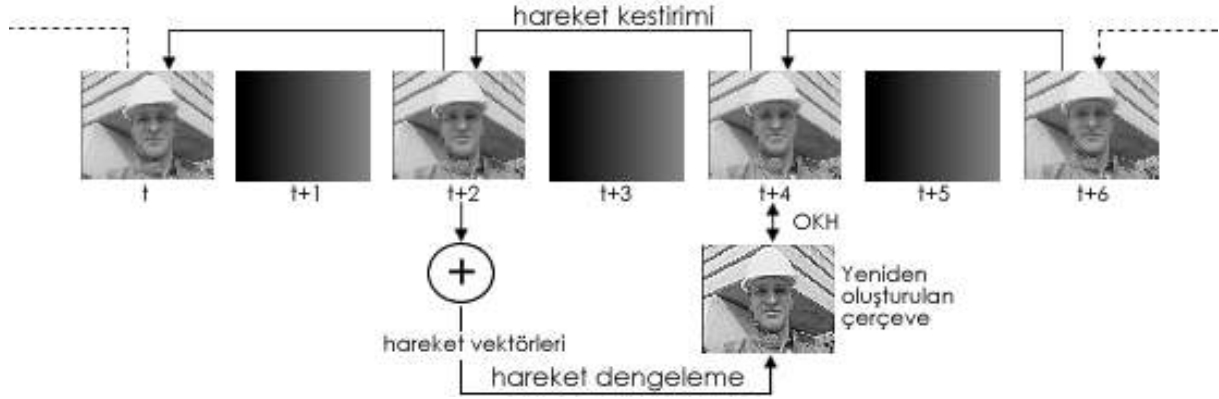
Gerçekleştirilen hareket kestirimi yöntemlerinin karşılaştırılması için “Foreman”, “News” ve “Salesman” adlı videolar seçilmiştir (Şekil 3.23). Videoların tümü QCIF formatında, YCbCr uzayında 4:2:0 örnekleme ile kodlanmış tam renkli video görüntüleridir.

Herhangi bir hareket kestirimi algoritmasında bloklar karşılaştırılırken piksellerin yalnızca parlaklık bilgisi kullanılabileceği gibi, RGB bilgisi de kullanılabilir. C_{ijk} çalışılan bloktaki bir pikselin, R_{ijk} ise arama alanındaki bir pikselin kırmızı, yeşil ya da mavi renk bileşenini göstermek üzere tam renkli görüntülerde, örneğin, OMF (3.8)’deki gibi hesaplanabilir.

$$OMF = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^2 |C_{ijk} - R_{ijk}| \quad (3.8)$$

Çalışılan blok ile referans blok karşılaştırılırken yalnızca parlaklık bilgisinin kullanılması,

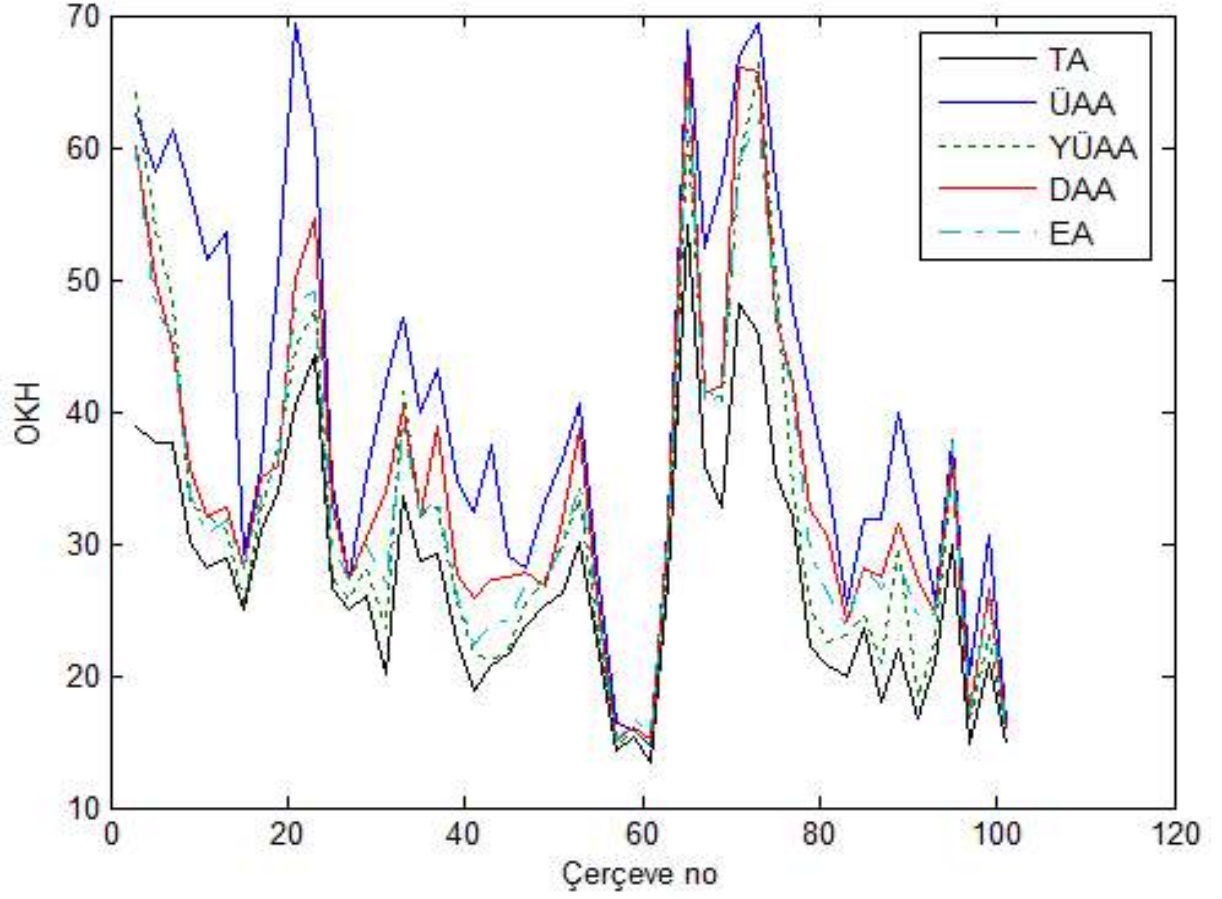
RGB bileşenlerinin kullanılmasına göre daha az işlemle blok eşleştirme ölçütünün hesaplanmasını sağlamaktadır. Parlaklık bilgisinin kullanılması ile hareket kestirimi başarılı bir biçimde yapılabilir. Eldeki video görüntüleri YCbCr uzayında kodlandıklarından, Y parlaklık bileşeni hareket kestirimi sırasında doğrudan kullanılmıştır.



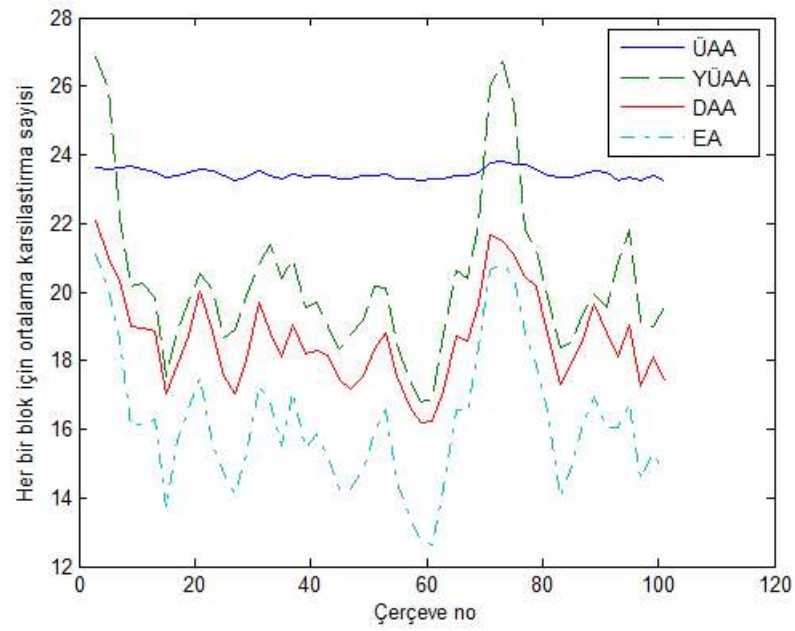
Şekil 3.24 Hareket kestirimi yöntemlerinin karşılaştırılmasında kullanılan yöntem. Video görüntüsünden birer çerçeve atlanarak 50 çerçeve alınmıştır. Çalışılan çerçeve ile referans çerçeve arasında hareket kestirimi yapılmıştır. Çalışılan çerçeve bir sonraki çalışılan çerçevenin hareket kestiriminde referans çerçeve olarak kullanılmaktadır.

Hareket kestirimi yöntemlerinin karşılaştırılması için “Foreman”, “News” ve “Salesman” videolarından, zamansal sıralamadaki üçüncü çerçeveden başlayıp, birer çerçeve atlanarak toplam 50 çerçeve alınmıştır. Her bir çalışılan çerçeve için iki önceki çerçeve referans olarak belirlenmiş ve hareket kestirimi algoritması uygulanmıştır. Elde edilen hareket vektörleri ve referans çerçeve kullanılarak çalışılan çerçeve yeniden oluşturulmuş ve özgün çerçeve ile yeniden oluşturulan çerçeve arasındaki OKH incelenmiştir. Uygulanan test yordamı Şekil 3.24’de gösterilmiştir.

Her üç videodan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinde blok boyutu 8 ve arama parametresi 7 olarak alınmıştır. “Foreman” videosundan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinden elde edilen OKH başarımı Şekil 3.25’de, bir çalışılan çerçevedeki her bir blok için yapılan ortalama karşılaştırma sayıları Şekil 3.26’da verilmiştir.



Şekil 3.25 Foreman videosundan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinden elde edilen OKH başarımları



Şekil 3.26 Foreman videosundan alınan çerçevelere uygulanan hareket kestirimi yöntemlerinde, bir çerçevedeki ortalama blok karşılaştırma sayıları

Çizelge 3.2 Hareket kestirimi yöntemlerinin çerçeve başına OKH başarımları

	TA	ÜAA	YÜAA	DAA	EA
Foreman	27.51	40.51	31.94	34.53	32.93
News	22.91	25.59	24.13	25.46	24.87
Salesman	8.99	9.87	9.44	9.58	9.51

Foreman, News ve Salesman videolarından alınan çerçevelere uygulanan hareket kestirimi yöntemlerinin çerçeve başına OKH başarımları Çizelge 3.2’de saptanmıştır.

3.8.1 Hareket Kestirimi Yöntemlerinden Elde Edilen Ortalama PSNR Başarımı

Video işleme yöntemlerinden elde edilen sonuçların görsel olarak değerlendirilmesi öznel ya da nesnel olarak yapılabilir. Örneğin, TA ile referans çerçeveden yeniden oluşturulan bir çalışılan çerçevenin görsel niteliği kişiden kişiye değişebilecek öznel bir değerlendirme ile ele alınabilir. Öznel yöntemler ile her zaman güvenilir sonuçlar elde edilemeyeceği için video işlemede kullanılan nesnel yöntemler geliştirilmiştir.

Video işlemede sıklıkla kullanılan nesnel değerlendirme ölçütlerinden birisi PSNR (Peak Signal to Noise Ratio) başarımının incelenmesidir. PSNR logaritmik bir ölçekte ölçülür. PSNR özgün çerçeve ile yeniden oluşturulan çerçeve arasındaki OKH’ya dayanmaktadır. maksP çerçevede kullanılan maksimum parlaklığı göstermek üzere PSNR (3.9)’daki gibi hesaplanabilir. Bir görüntü ya da görüntü dizisi için yüksek PSNR genellikle yüksek niteliği, düşük PSNR ise düşük niteliği göstermektedir.

$$PSNR = 10 \times \log_{10} \left(\frac{maksP^2}{OKH} \right) \quad (3.9)$$

Genellikle tek renkli görüntüler için, 40 dB üzerindeki PSNR özgün görüntüye çok yakın mükemmel bir görüntüyü, 30 ile 40 dB arasındaki PSNR bozulunun görünür, ancak kabul edilebilir olduğu iyi bir görüntüyü, 20 ile 30 dB arasındaki PSNR oldukça düşük nitelikli bir görüntüyü belirtir ve 20 dB’in altındaki PSNR kabul edilmez. Tüm çerçeveler için ortalama bir PSNR değeri bulunurken, çerçeve başına PSNR değerlerinin hesaplanarak ortalamalarının alınması yerine her bir çerçeve için hesaplanan OKH değerlerinin ortalamasından bir PSNR değeri hesaplanmalıdır. OKH ya da PSNR başarımının iki görüntü arasındaki görsel

bozulumla her zaman ilişkili olmadığı bilinmektedir (Wang vd., 2002).

Çizelge 3.3 Hareket kestirimi yöntemlerinin ortalama PSNR başarımı (dB)

	TA	ÜAA	YÜAA	DAA	EA
Foreman	33.73	32.05	33.08	32.74	32.95
News	34.52	34.04	34.30	34.07	34.17
Salesman	38.59	38.18	38.38	38.31	38.34

Foreman, News ve Salesman videolarından alınan çerçevelere uygulanan hareket kestirimi yöntemlerinin ortalama PSNR başarımları Çizelge 3.3’de saptanmıştır. Wang vd. (2002) tarafından belirtildiği gibi ortalama PSNR başarımı, yeniden oluşturulan çerçeveler ile özgün çerçeveler arasında hesaplanan OKH değerlerinin ortalamasının PSNR’ye dönüştürülmesi biçiminde hesaplanmıştır.

3.8.2 Karşılaştırma Sonuçlarının Değerlendirilmesi

Hareket yöntemlerinin uygulanmasından alınan sonuçlar kullanılan videoların hareket içeriğine göre değerlendirildiğinde çok hareketli ya da kamera hareketinin belirgin olduğu videolarda ya da çerçevelerde hareket kestirimi başarımının düştüğü görülmektedir. Bir nesnenin değişmez bir arkaplanda hareket ettiği durumda, çalışılan çerçevede nesnenin hareketinden dolayı açığa çıkan arkaplan bir önceki referans çerçevede nesne yüzünden gizlenmiştir. Benzer biçimde çalışılan çerçevedeki arkaplana ilişkin bazı bölgeler nesne hareketi yüzünden örtülecektir. Nesne hareketinden dolayı açığa çıkan ya da gizlenen bölgelerdeki blokların hareket kestirim başarımı nesnenin hızı, hareket kestiriminde kullanılan blok boyutu, arama parametresi gibi etmenlere bağlıdır. Görece az hareketli videolarda referans çerçevede oluşturulan uygun bir arama penceresinde, nesne hareketinden dolayı değişen bölgelerdeki bloklar doğru biçimde eşleştirilebilmektedir. Çok hareketli videolarda ya da çerçevelerin hızlı hareket eden bölgelerinde bazı blokların arama penceresinde yanlış eşleştirilmeleri söz konusudur ve bu durum hareket kestirimi başarımını azaltır. Bu nedenle Foreman videosundan diğer iki videoya göre daha düşük bir hareket kestirimi başarımı elde edilmektedir. Buna karşılık görece en az hareketli Salesman videosunda hareket kestirimi oldukça başarılıdır.

TA'da arama penceresindeki tüm aday bloklara bakıldığından beklenildiği gibi en başarılı hareket kestirimi sonucu elde edilmiştir. Ancak hızlı yöntemlere göre TA'nın hesaplama karmaşıklığı oldukça yüksektir. Yapılan denemelerde arama parametresi 7 olarak alındığı için arama penceresinde 225 aday blok söz konusudur ve her bir çalışılan blok 225 aday blokla karşılaştırılmaktadır. Bu nedenle Şekil 3.26'da, TA için blok başına ortalama karşılaştırma sayısı grafiği belirtilmemiştir.

Hızlı hareket kestirimi yöntemlerinden, merkez-yanlı arama örüntüleri kullanan YÜAA, DAA ve EA yöntemleri TA'ya yakın sonuçlar üretmektedir. ÜAA algoritmasındaki seyrek karşılaştırma noktalarından oluşan örüntü nedeniyle, çalışılan blok ve aday blokların karşılaştırılmasından elde edilen OMF değerlerinin oluşturduğu hata fonksiyonundaki yerel minimum noktalarına yakalanılması olasılığı yüksek olduğundan, ÜAA'nın başarımı diğer hızlı hareket kestirimi yöntemlerine göre düşüktür.

YÜAA merkez-yanlı bir ÜAA algoritması olduğu için hata fonksiyonundaki global maksimum noktanın bulunması yüksek olasılıklıdır. YÜAA, ÜAA'da kullanılan arama örüntüsüne ek olarak yeni arama noktaları kullanmasına karşın, ÜAA'ya göre başarımının daha iyi olması bu eksikliğini karşılamaktadır. Her ne kadar DAA ve EA ile benzer başarıma sahip olsa da, YÜAA'nın başarımı diğer iki yöntemle göre biraz daha iyidir. Ancak DAA ve EA algoritmalarında ortalama karşılaştırma noktalarının sayısı YÜAA'ya göre daha azdır. Buna göre EA yöntemi özellikle öne çıkmaktadır. Zhu ve Ma (2000) tarafından bildirildiği üzere, EA yöntemi YÜAA'ya göre hesaplanan karşılaştırma noktalarından ortalama olarak %20-%25 tasarruf sağlamaktadır. EA, DAA'ya göre daha az karşılaştırma noktasına gereksinim duymaktadır (Şekil 3.26) ve daha iyi OKH başarımına sahiptir (Şekil 3.25).

OKH ya da PSNR başarımı bakımından TA'ya yakınlıklarına göre yöntemler YÜAA, EA, DAA ve ÜAA biçiminde sıralanabilir. Öte yandan zamansal karmaşıklığı gösteren ortalama karşılaştırma noktaları bir ölçüt olarak kullanıldığında EA en iyi başarıma sahiptir. Elde edilen sonuçlardan PSNR başarımı ve zamansal karmaşıklık ölçütlerine göre en uygun hızlı blok eşleştirme algoritmasının EA olduğu söylenebilir.

4. HAREKET KESTİRİMİ YÖNTEMLERİNİN GELİŞTİRİLMESİ

4.1 Hiyerarşik Arama

Hızlı BEA’nda arama penceresinde oluşturulan hata fonksiyonunun yerel minimum noktalarına yakalanılması olasılığı vardır. Hızlı BEA’nda, TA’nın zamansal karmaşıklığı kabul edilebilir bir sınırın altına çekilmesine karşılık eşleştirme başarımından ödün verilir. Hem TA’nın zamansal karmaşıklığının azaltılması hem de arama penceresindeki global minimum noktanın daha yüksek doğrulukla kestirilebilmesi için hiyerarşik BEA geliştirilmiştir.

Hiyerarşik aramada çalışılan ve referans çerçeveler farklı çözünürlüklerde temsil edilir ve en düşük çözünürlük seviyesinden başlayarak, sırayla her bir seviyede hareket kestirimi yapılır. Hareket vektörünün kaba bir kestirimini elde etmek için hareket kestirimi ilk olarak en düşük seviyede yapılır. Hesaplanan hareket vektörü bir başlangıç kestirimi olarak bir sonraki yüksek çözünürlük seviyesine aktarılır. Başlangıç kestirimini düzeltmek için daha yüksek çözünürlük seviyesinde hareket kestirimi yapılır. Bu işlem dizisi en yüksek çözünürlük seviyesine erişilene dek yinelenir. Aramanın karmaşıklığını azaltmak için düşük çözünürlüklerde daha küçük blok boyutları ve yüksek çözünürlüklerde, arama iyi bir başlangıç kestirimi ile başladığından, daha küçük arama parametresi kullanılır. Çalışılan çerçevenin çözünürlüğü azaltılırken, hareket hızı azaldığından hiyerarşik yöntemler ile çok hareketli videolarda düşük karmaşıklıkla hareket kestirimi yapılabilir (Al-Mualla vd., 2003).

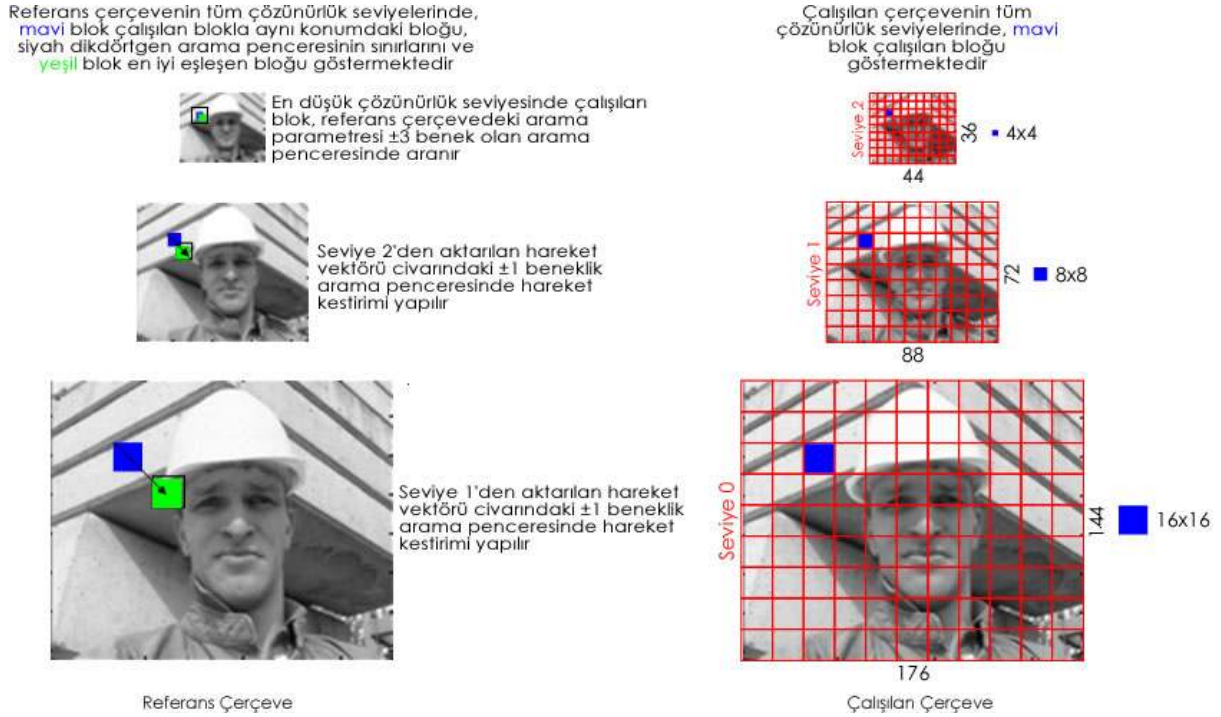
Bazı hiyerarşik BEA’nda tüm hiyerarşi seviyeleri için aynı çerçeve boyutu alınırken, daha düşük seviyelerde daha büyük blok boyutları kullanılabilir. Diğer yöntemlerde ise hiyerarşinin tüm seviyelerinde aynı blok boyutu alınırken, daha düşük seviyelerde alt-örneklenmiş çerçeveler kullanılır. Her iki durumda da, herhangi bir seviyede bir sonraki yüksek seviyeye göre daha az blok elde edilmektedir. Bu nedenle, herhangi bir seviyede hesaplanan bir hareket vektörü daha yüksek seviyede birden fazla bloğa aktarılacaktır (Al-Mualla vd., 2003).

Tipik bir hiyerarşik blok eşleştirme algoritması aşağıdaki gibi gerçekleştirilebilir: (Richardson, 2002)

- 1) Çalışılan ve referans çerçeveler Seviye 0’da tam çözünürlüktedir. Seviye 1’i elde etmek için seviye 0’daki çalışılan çerçeve alçak geçiren filtreden geçirilir ve yatay ve düşey doğrultularda altörneklenir. Aynı işlem referans çerçeveye de uygulanır. Alçak geçiren

filtreleme için basit bir ortalama filtresi kullanılabileceği gibi daha başarılı bir sonuç için Gauss filtresi de kullanılabilir.

- 2) Seviye 1 altörneklenecek Seviye 2 elde edilir ve bu işlem istenilen sayıda hiyerarşi seviyesi elde edilene dek yinelenir.
- 3) En düşük çözünürlük seviyesinde hareket kestirimi yapılarak başlangıçtaki ‘kaba’ hareket vektörü elde edilir.
- 4) Bir sonraki daha yüksek çözünürlük seviyesinde, ‘kaba’ hareket vektörünün konumu civarında hareket kestirimi yapılarak en iyi eşleşen blok bulunur.
- 5) Seviye 0’daki en iyi eşleşen blok bulunana dek adım 4 yinelenir.



Şekil 4.1 Hiyerarşik blok eşleştirme algoritması

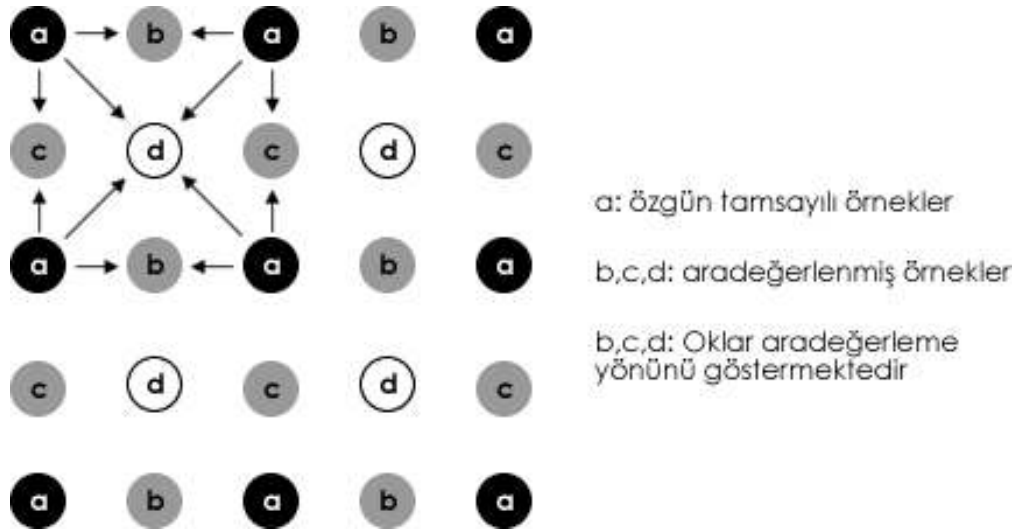
Hiyerarşik hareket kestirimi 3 seviye için Şekil 4.1’de gösterilmiştir. QCIF formatında bir video ile çalışıldığı varsayıldığından özgün çerçeve boyutu 176x144’tür. Altörnekleme yatay ve dikey doğrultularda birer piksel atlanarak yapıldığından seviye 0, seviye 1 ve seviye 2’deki çerçeve boyutları sırasıyla 176x144, 88x72 ve 44x36’dır. En düşük çözünürlük seviyesi olan seviye 2’de hareket kestirimi, blok boyutu 4x4 ve arama parametresi ± 3 piksel alınarak yapılır. Herhangi bir blok için hesaplanan hareket vektörü 2 ile çarpılarak seviye 1’deki

karşılık gelen bloğa aktarılır. Seviye 1'deki hareket kestirimi, blok boyutu 8x8 alınarak daha düşük çözünürlük seviyesinden aktarılan hareket vektörü civarındaki ± 1 piksel parametrelili arama penceresinde yapılır. Böylelikle en düşük çözünürlük seviyesinde hesaplanan 'kaba' hareket vektörü daha yüksek çözünürlük seviyesinde hesaplanacak hareket vektörü için bir başlangıç tahmini olarak kullanılmaktadır. Hesaplanan hareket vektörü 2 ile çarpılarak seviye 0'a aktarılır ve bu seviyede hareket kestirimi blok boyutu 16x16 alınarak daha düşük çözünürlük seviyesinden aktarılan vektör civarındaki ± 1 piksel parametrelili arama penceresinde gerçekleştirilir.

Seviye 2'deki ± 3 piksellik arama parametresi seviye 1'de ± 6 piksellik arama parametresine karşılık gelir. Seviye 1'de daha düşük çözünürlük seviyesinden aktarılan hareket vektörü civarında ± 1 piksellik bir arama penceresi oluşturulduğu için seviye 1'de ± 7 piksellik arama penceresi elde edilir. Seviye 1'deki ± 7 piksellik arama parametresi seviye 0'da ± 14 piksellik arama parametresine eşdeğerdir. Seviye 0'da da ± 1 piksellik bir arama penceresi oluşturulduğu için en yüksek çözünürlük seviyesinde ± 15 piksellik bir arama penceresi elde edilir. Sonuç olarak belirtilen hareket kestirimi parametreleriyle hiyerarşik yöntemin uygulanması blok boyutunun 16x16 ve arama parametresinin ± 15 piksel alındığı TA'ya karşılık gelmektedir.

4.2 Piksel-altı (Sub-pixel) Hareket Kestirimi

BEA'nın pek çoğunda hareketli bir nesnenin ardışıl video çerçeveleri arasında tamsayıli öteleme yaptığı varsayılır. Ancak hareketli nesnenin gerçek öteleme miktarı kesirli bir sayı olabilir. Bu olguya dayanarak hareket dengelemesinin daha iyi yapılabilmesi için tamsayıli konumlardaki piksellerden kesirli konumlardaki piksellerin aradeğerleme ile elde edilmesi gerekmektedir. Şekil 4.2'de yarım-piksel (half-pixel) aradeğerlemesi gösterilmiştir. Tamsayıli konumlardaki a örnekleri siyahla gösterilmiştir. Kesirli konumlardaki (gri) b ve c örnekleri tamsayıli piksel çiftleri arasında doğrusal aradeğerleme ile elde edilir ve (beyaz) d örnekleri, tamsayıli konumdaki dört piksel kullanılarak aradeğerlenir.

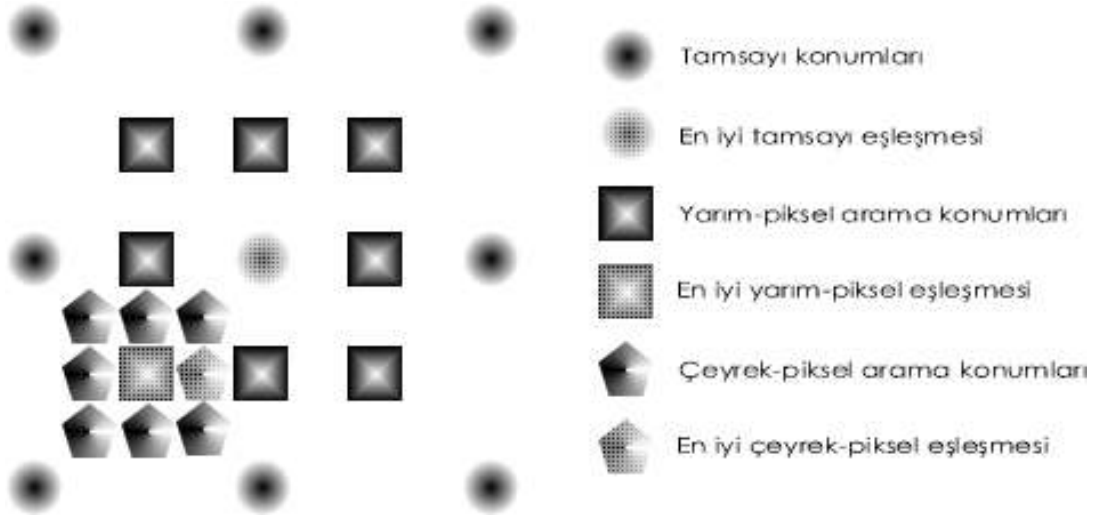


Şekil 4.2 Yarım-piksel aradeğerlemesi (Richardson, 2002)

Piksel-altı BEA'nda arama penceresindeki örnekler aradeğerlenerek daha yüksek çözünürlüklü, aradeğerlenmiş bir arama alanı elde edilir. Aradeğerlenmiş alanda tamsayı ve kesirli piksel konumlarında arama yapılır ve en iyi eşleşen blok bulunur. Video sıkıştırmasında fark bloğunu oluşturmak için eşleşen bloktaki örnekler çalışılan bloktaki örneklerden çıkarılır (Richardson, 2002). Böylelikle kesirli değerlerle ötelenen bloklarda fark enerjisi daha etkin hareket dengelemesi ile en aza indirilir.

Yarım-piksel aradeğerlemesi ile hareket kestirimi yapıldığında karşılaştırma noktalarının sayısı tamsayı konumlarında yapılan hareket kestirimine göre dört katına çıkmaktadır. Toplamdaki karmaşıklık aradeğerleme aşaması ile daha da artar. Algoritmanın karmaşıklığını azaltmak için önce en iyi eşleşen tamsayı konumu bulunur ve bu konumun komşuluğundaki aradeğerlenmiş konumlarda hareket kestirimi yapılır.

Çeyrek-piksel (quarter-pixel) hareket kestiriminde (Şekil 4.3), ilk önce en iyi eşleşen tamsayı konumu bulunur; daha sonra tamsayı konum civarındaki bir komşulukta en iyi eşleşen yarım-piksel konumu elde edilir; en sonunda yarım-piksel konumu civarında çeyrek-piksel konumundaki en iyi eşleşen blok bulunur (Zhang vd., 2004).



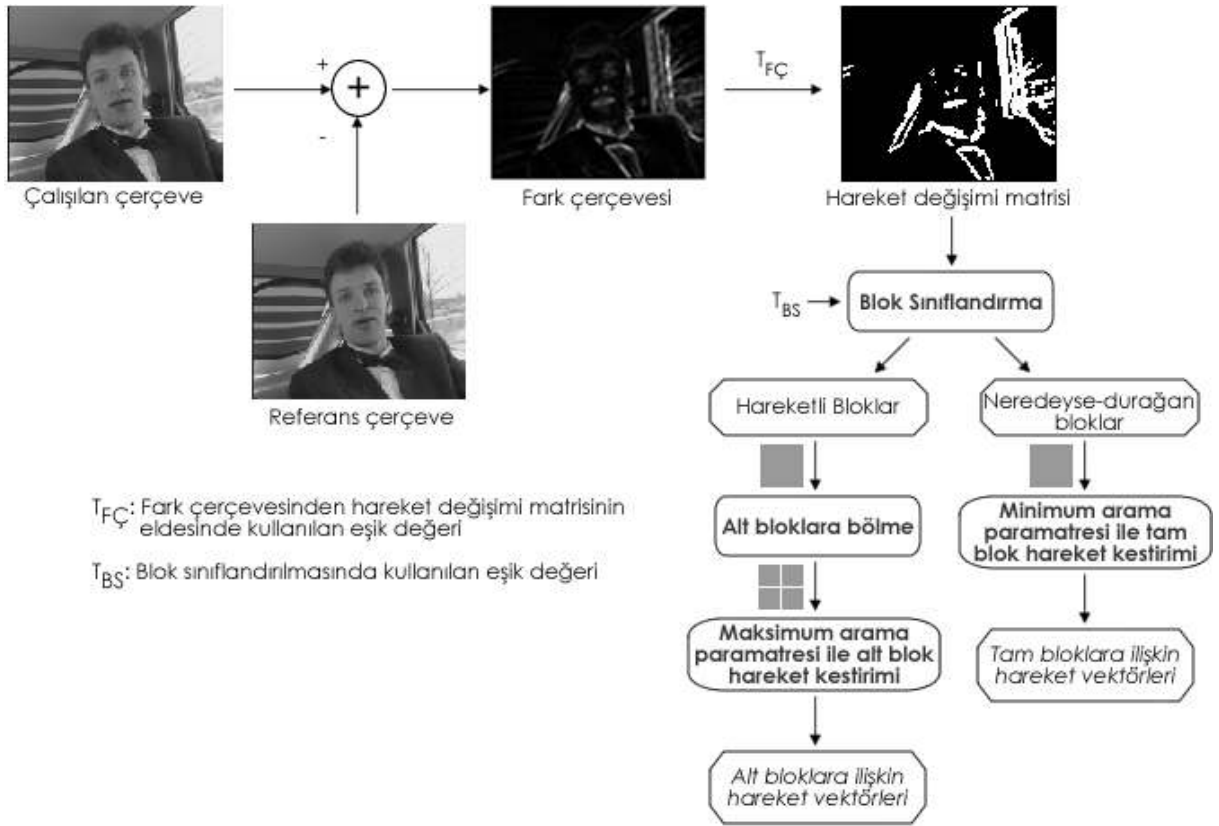
Şekil 4.3 Çeyrek-piksel hareket kestirimi (Zhang vd., 2004)

4.3 Tezde Önerilen Yöntem

BEA’nda çalışılan çerçeve birbiriyle örtüşmeyen, sabit boyutlu bloklara ayrılmaktadır. Her bir blok için hareket vektörünün belirlenebilmesi için referans çerçevede arama parametresiyle belirlenen bir arama penceresi oluşturulmaktadır. Hareket kestiriminin başarımı, kullanılan blok boyutu ve arama parametresiyle ilintilidir. Büyük bir blok kullanıldığında özellikle hareket eden kenarlarda hareketin belirlenme başarısı düşer. Çok hareketli videolarda çalışılan blokla doğru eşleşen referans bloğun bulunabilmesi için uygun bir arama parametresi seçilmelidir. Durağan ya da neredeyse bloklar için arama maliyetini arttıracak biçimde büyük bir arama parametresinin kullanılmasına gerek yoktur.

Hareket kestiriminin başarımını arttırmak için, çalışılan çerçevenin referans çerçeveye göre hareket içeriğini göz önünde bulundurarak uyarlamalı blok boyutu ve arama parametresi kullanan bir yöntem önerilmektedir.

Çalışılan çerçeve ile referans çerçeve arasındaki hareket değişimi belirlenerek, hareket değişiminin fazla olduğu bloklar 4 alt bloğa ayrılmaktadır. Belirlenen maksimum arama parametresi kullanılarak her bir alt blok için ayrı hareket kestirimi yapılmaktadır. Hareketin olmadığı ya da az olduğu düşünülen bloklarda minimum arama parametresi ile merkez civarında hareket kestirimi uygulanmıştır. Yöntemin blok diyagramı Şekil 4.4’de gösterilmiştir.



Şekil 4.4 Tezde önerilen yöntemin blok diyagramı

4.3.1 Hareket Değişiminin Belirlenmesi

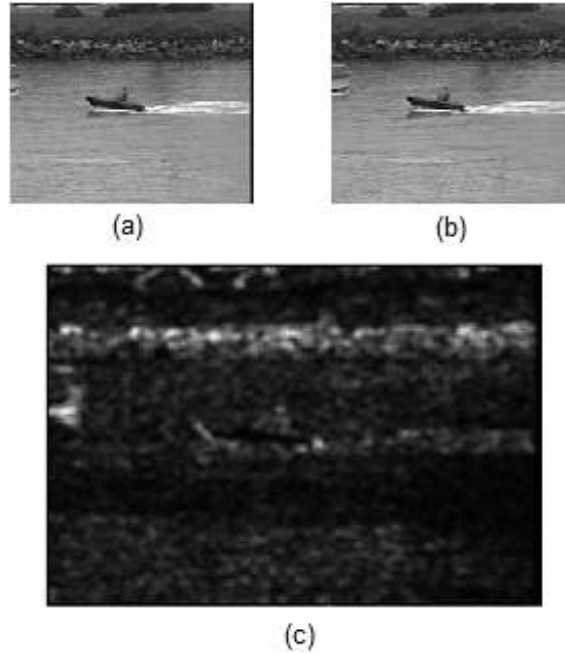
Çalışılan ve referans çerçevelerde hareket eden bölgelerdeki pikseller arasında bir benzerlik yoktur. Hareket eden bölgelerde, çalışılan ve referans çerçevelerdeki karşılıklı piksellerin farklarının mutlak değeri önemli ölçüde artar. Her iki çerçevede hareketin sınırlı olduğu bölgelerdeki piksel farklarının mutlak değeri sıfıra yaklaşır. Öyleyse çalışılan çerçeve ve referans çerçeve arasındaki fark sinyali belirli bir eşik değeri ile karşılaştırılarak hareket değişimi belirlenebilir.

Fark çerçevesi, çalışılan çerçeve ile referans çerçeve farkının mutlak değeri alınarak hesaplanabilir. Nesnelerin hareketlerinden ya da parlaklık değişiminden kaynaklanan değişimleri gürültüden kaynaklanan zamansal değişimlerden ayırt edebilmek için, belirli bir (x, y) konumundaki karşılıklı iki piksel farkının mutlak değerini almak yerine (x, y) konumunda merkezlenmiş bir pencerede karşılıklı piksellerin OMF'ı (x, y) konumundaki fark sinyali olarak belirlenebilir (Zhang ve Lu, 2001). Fark çerçevesi hesaplandıktan sonra uygun bir eşik değeri kullanılarak ikili (binary) hareket değişimi matrisi elde edilir. Hareket değişimi matrisinde 1 olan değerler karşılık gelen pikselin hareket eden bir bölgeye ilişkin olduğunu

göstermektedir.

$FÇ(x, y)$ fark çerçevesinde (x, y) konumundaki fark sinyalinin, $T_{FÇ}$ kullanılan eşik değerini ve $HDM(x, y)$ hareket değişimi matrisinin (x, y) konumundaki değerini göstermek üzere hareket değişimi matrisi (4.1)'deki gibi hesaplanabilir.

$$HDM(x, y) = \begin{cases} 0, & FÇ(x, y) < T_{FÇ} \\ 1, & FÇ(x, y) \geq T_{FÇ} \end{cases} \quad (4.1)$$



Şekil 4.5 Fark sinyalinin ilgili pikselin 3x3 komşuluğundaki piksellerin OMF'ı ile hesaplandığı fark çerçevesi. (a) Referans çerçeve (b) Çalışılan çerçeve (c) Fark çerçevesi

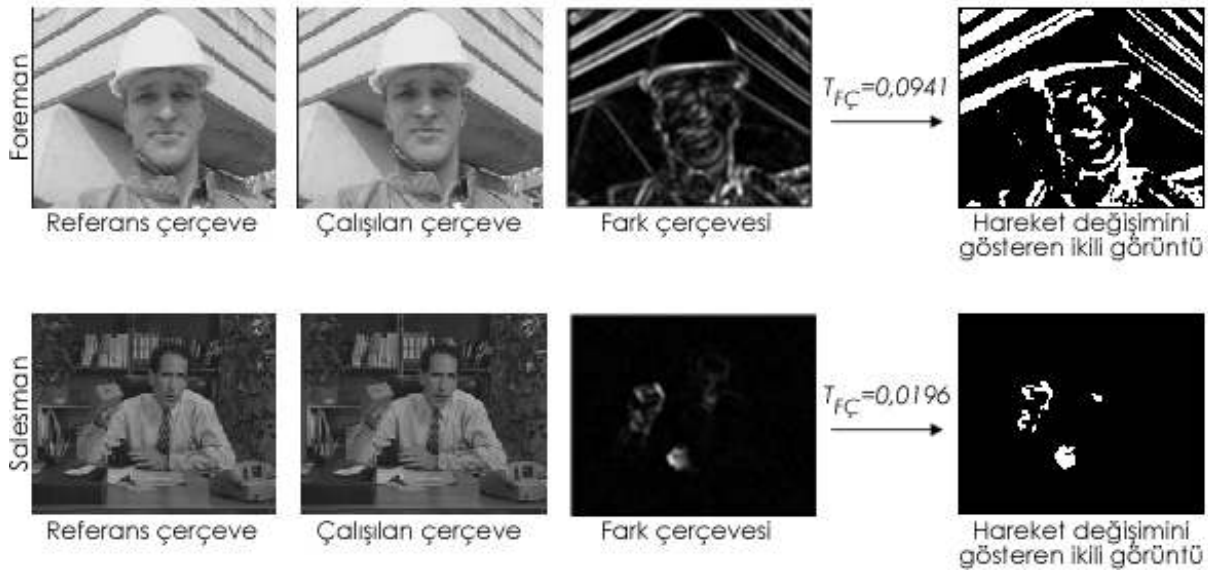
4.3.2 Hareket Değişimi Matrisinin Hesaplanmasında Kullanılan Eşik Değeri

Hareket değişimi matrisinin hesaplanmasında kullanılan eşik değeri, çalışılan ve referans çerçevelerdeki hareket içeriğine bağlıdır. Aynı videonun çok hareketli çerçeveleri için kullanılan eşik değeri ile az hareketli çerçeveleri için kullanılacak eşik değerleri birbirinden farklıdır.

Fark çerçevesindeki her bir fark sinyalinin ilgili pikselin 3x3 komşuluğundaki piksellerin OMF'ı hesaplanarak bulunduğu fark çerçevesi Şekil 4.5'de gösterilmiştir. Her bir çalışılan çerçevede en uygun eşik değerinin bulunması için eşik değerinin otomatik olarak

hesaplanması tercih edilmiştir. Şekil 4.5(c)'deki fark çerçevesi incelendiğinde kabaca gri seviyesinde bir görüntünün elde edildiği görülür. Fark sinyalinin düşük olduğu bölgeler tipik bir gri seviye görüntüsündeki arkaplan ve yüksek olduğu bölgeler nesneler olarak düşünülebilir. Bu durumda problem, piksellerin arkaplana ya da nesnelere ilişkin olduğu iki gruba birbirinden ayıran eşik değerinin bulunmasına dönüşür. Otomatik eşikleme için “Otsu yöntemi” kullanılmıştır.

İki gruba birbirinden ayıran eşik değeri seçilirken bir grubun kendi içerisinde homojen ve diğerinden farklı olması istenir. Bir grubun homojenliğinin ölçüsü olarak varyans kullanılabilir. Yüksek homojenliğe sahip bir grubun varyansı düşüktür. Düşük homojenliğe sahip bir grubun ise varyansı yüksektir. Otsu otomatik eşikleme yönteminde grup içi varyansların ağırlıklı toplamını en aza indiren eşik değeri iki gruba ayırıcı ölçüt olarak seçilir. Ağırlıklar her bir grubun olasılıklarıdır (Shapiro ve Stockman, 2001).



Şekil 4.6 Video çerçevelerindeki hareket içeriğine göre fark çerçevesinin ve eşikleme değerinin değişimi

Çok hareketli bir videonun çerçeveleri için kullanılacak eşik değerleri az hareketli bir videonun çerçevelerinde kullanılacak eşik değerlerinden büyük olacaktır. Çok hareketli “Foreman” ve az hareketli “Salesman” videolarından rasgele alınmış ardışıl video çerçevelerinden elde edilen fark çerçeveleri ve otomatik olarak belirlenmiş T_{FC} eşik değerleri Şekil 4.6’de gösterilmiştir.

4.3.3 Çalışılan Çerçeveadaki Blokların Sınıflandırılması

Hareket değişimi matrisi kullanılarak çalışılan çerçevedeki birbiriyle örtüşmeyen bloklar “hareketli” ve “neredeyse-durağan” bloklar olarak sınıflandırılabilir. Sınıflandırma için hareket değişimi matrisi hareket kestirimi için verilen blok boyutu kullanılarak birbiriyle örtüşmeyen bloklara ayrılır ve her bir bloktaki 1 olan değerlerin toplam sayısı hesaplanır. Hesaplanan toplam değer bir eşik değerine eşit ya da ondan büyük ise ilgili blok “hareketli” blok olarak sınıflandırılır. Bir bloğa hareket değişimi matrisinde karşılık gelen 1’lerin sayısı eşik değerinden küçükse o bloğun “neredeyse-durağan” olduğuna karar verilir.

Blokların sınıflandırılmasında kullanılan T_{BS} eşik değeri, fark çerçevesinin eşiklemede kullanılan $T_{FÇ}$ eşik değeri kadar kritik değildir. Bir bloğun hareketli olup olmadığına pratik olarak tüm videolar için kullanılabilecek tek bir eşik değeriyle karar verilebilir. T_{BS} eşik değeri deneysel olarak belirlenmiştir. Blok boyutu 8x8 alındığında, hareket değişimi matrisinde her bir bloktaki 64 tane ikili değerden 1 olanları sayılmaktadır. T_{BS} yüzdesel değer olarak belirtilmektedir. Örneğin T_{BS} %25 alınırsa bir bloğun “hareketli” blok olarak sınıflandırılması için o bloktaki 1 olan ikili değer sayısının en az 16 olması gerekmektedir. T_{BS} ’nin deneysel olarak belirlenmesi için Foreman, News ve Salesman videolarından Bölüm 3.8’deki gibi elde edilen çerçevelere, blok boyutu 8x8, arama parametresi 7 ve T_{BS} sırasıyla %20, %40 ve %60 alınarak önerilen yöntem uygulanmıştır. Üç farklı T_{BS} eşik değeri için elde edilen ortalama PSNR başarımları ve ortalama hareket kestirimi süreleri sırasıyla Çizelge 4.1 ve Çizelge 4.2’de gösterilmiştir. Hareket kestirimi süreleri incelenirken sonuçlar kendi aralarında görece karşılaştırıldığından süre ölçümünün yapıldığı test platformu belirtilmemiştir.

Çizelge 4.1 Farklı T_{BS} eşik değerlerine göre PSNR başarımları (dB)

	$T_{BS} = \%20$	$T_{BS} = \%40$	$T_{BS} = \%60$
Foreman	33.83	33.62	33.33
News	35.24	34.96	34.66
Salesman	39.78	39.29	38.84

Çizelge 4.2 Farklı T_{BS} eşik değerlerine göre ortalama hareket kestirimi süresi (sn)

	$T_{BS} = \%20$	$T_{BS} = \%40$	$T_{BS} = \%60$
Foreman	0.5740	0.5008	0.4490
News	0.4343	0.4187	0.4195
Salesman	0.4184	0.4074	0.3999

T_{BS} küçük bir değer alındığında alt bloklara bölünen blok sayısı ve hareket kestirimi süresi artmaktadır. Foreman gibi çok hareketli bir videoda küçük bir eşik değeri seçildiğinde kestirim süresi artarken en iyi PSNR başarımı elde edilmektedir. News ve Salesman gibi görece az hareketli videolarda da küçük bir eşik değeri ile en iyi PSNR başarımı elde edilirken, alt bloklara ayrılan blok sayısı az olduğundan, kestirim süresindeki artış çok azdır. Buna göre önerilen yöntemde blokların sınıflandırılmasında kullanılan eşik değeri tüm videolar için %20 olarak seçilmiştir.

4.3.4 Sınıflandırılan Bloklara Hareket Kestiriminin Uygulanması

Hareketli blokların her biri 4 tane alt bloğa ayrılarak maksimum arama parametresi ile hareket kestirimi yapılır. Örneğin tam blok boyutu 8x8 alınırsa hareketli bir blok 4 tane 4x4 boyutlu alt bloğa ayrılmaktadır. Neredeyse-durağan bloklar için tam blok boyutu ve minimum arama parametresi kullanılarak hareket kestirimi yapılır.

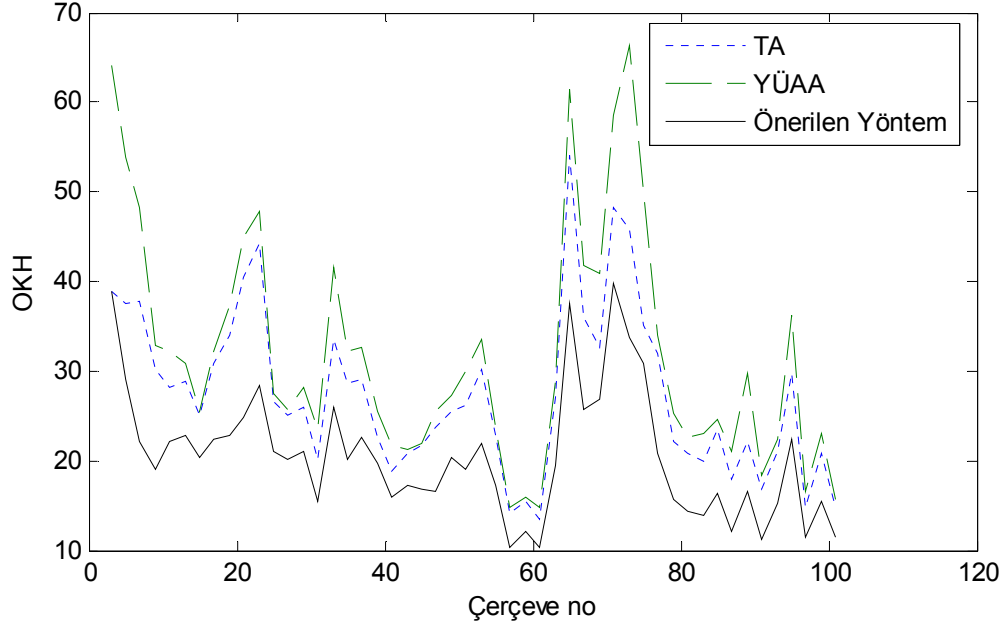
Hareketli bloklar için tam blok boyutuna göre küçük bir blok boyutuyla geniş bir alanda hareket kestirimi yapılmaktadır. TA kullanıldığında en iyi başarımlar elde edilmektedir. Ancak hareketli bir blok, alt bloklara bölündüğü için hareket kestirimi 4 kez yinelenacaktır.

Yöntemin zamansal karmaşıklığını iyileştirmek için alt blokların hareket kestirimi EA algoritması ile yapılmaktadır. EA, gerek zamansal karmaşıklığının düşük olması, gerekse de alt bloklar için görece büyük arama bir alanında global minimumun diğer hızlı arama yöntemlerine göre başarıyla bulunması nedeniyle tercih edilmiştir.

4.3.5 Önerilen Yöntemden Elde Edilen Sonuçlar

Önerilen yöntemin değerlendirilmesi için Bölüm 3.8’de hareket kestirimi algoritmalarının karşılaştırılmasında kullanılan test yordamı uygulanmıştır. Yöntem Foreman, News ve

Salesman videoları için TA ve YÜAA ile karşılaştırılmıştır. Tüm algoritmalarda blok boyutu 8x8 alınmıştır. TA ve YÜAA için arama parametresi 7, önerilen yöntemde maksimum arama parametresi 7 ve minimum arama parametresi 2 olarak seçilmiştir.

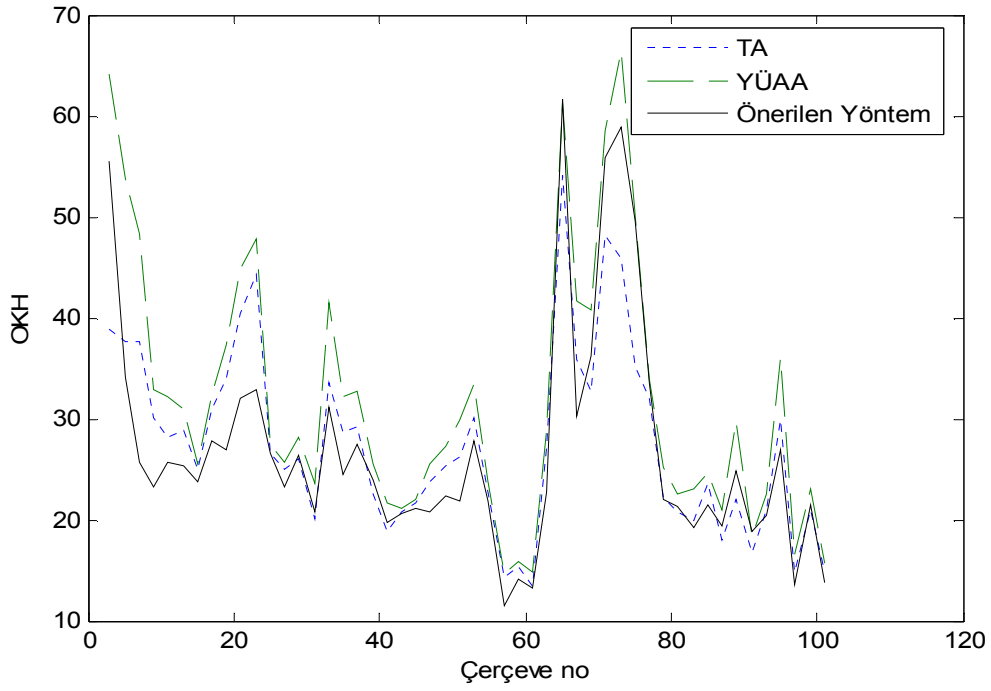


Şekil 4.7 Alt blokların hareket kestiriminin TA ile yapıldığı önerilen yöntemin Foreman videosu için TA ve YÜAA ile OKH karşılaştırması

Çizelge 4.3 Alt blokların hareket kestiriminin TA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının PSNR başarımları (dB)

	TA	YÜAA	Önerilen Yöntem
Foreman	33.73	33.08	35.00
News	34.52	34.30	36.20
Salesman	38.59	38.38	40.53

Önerilen yöntemde alt blokların hareket kestiriminin TA ile yapıldığı durumda elde edilen OKH başarımları Foreman videosu için Şekil 4.7’de, tüm videolar için yöntemlerin ortalama PSNR başarımları Çizelge 4.3’de gösterilmiştir.



Şekil 4.8 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin Foreman videosu için TA ve YÜAA ile OKH karşılaştırması

Çizelge 4.4 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının PSNR başarımları (dB)

	TA	YÜAA	Önerilen Yöntem
Foreman	33.73	33.08	33.83
News	34.52	34.30	35.24
Salesman	38.59	38.38	39.78

Çizelge 4.5 Alt blokların hareket kestiriminin EA ile yapıldığı önerilen yöntemin, TA ve YÜAA algoritmalarının ortalama hareket kestirimi süreleri (sn)

	TA	YÜAA	Önerilen Yöntem
Foreman	0.9629	0.1068	0.5701
News	0.9346	0.0825	0.4298
Salesman	0.9399	0.0817	0.4381

Önerilen yöntemde alt blokların hareket kestiriminin EA ile yapıldığında elde edilen OKH başarımları Foreman videosu için Şekil 4.8’de verilmiştir. Karşılaştırılan yöntemler için elde edilen ortalama PSNR başarımları ve ortalama hareket kestirimi süreleri sırasıyla Çizelge 4.4 ve Çizelge 4.5’de saptanmıştır.

Alt blok hareket kestiriminde TA kullanıldığında önerilen yöntemin OKH başarımları TA dahil diğer yöntemlere göre belirgin olarak iyileştirilmektedir. OKH başarımlarının tek ölçüt olarak alındığı durumda yöntem TA’nın yerine rahatlıkla kullanılabilir.

Alt blok hareket kestiriminde EA kullanıldığında ise önerilen yöntem çok hareketli videolarda TA’dan biraz daha iyi bir başarımlar gösterirken, az hareketli videolarda TA’ya göre belirgin olarak daha iyidir. Yöntemin TA’ya göre hemen hemen eş başarımlar göstermesinin nedeni, bazı çerçevelerdeki alt blokların EA ile hareket kestiriminde global minimum noktadan uzaklaşılmasıdır. Alt blok hareket kestirimi hızlı bir blok eşleştirme algoritması ile gerçekleştirildiğinden yöntemin bu biçimiyle bilinen hızlı algoritmalarla karşılaştırılması uygun olacaktır. Önerilen yöntem hızlı algoritmalarla YÜAA’ya göre videonun hareket içeriğinden bağımsız olarak daha iyi bir başarımlar göstermektedir.

5. VIDEO ÇERÇEVE ARADEĞERLEMESİ

Video-konferans, görüntülü-telefon ve video oyunları gibi pek çok çoğulortam uygulamalarında düşük bit hızıyla video sıkıştırması önemli bir rol oynamaktadır. Bu çoğulortam uygulamalarının kullanıldıkları ortamlarda çoğu zaman mevcut iletişim bant-genişliği sınırlıdır.

Sınırlı bantgenişliği kısıtını karşılamak üzere konumsal ve zamansal niteliğin azaltıldığı iki yaklaşım ele alınmaktadır. Konumsal niteliğin azaltılması için nicemleme aşaması daha kabaca gerçekleştirilebilir ancak bloksal bozukluklar ortaya çıkar. Bu konumsal bozukluklar sonradan filtreleme ve örtüşen bloklarla hareket dengelemesi gibi yöntemler ile düzeltilebilir. Zamansal nitelik ise çerçeve hızı azaltılarak düşürülebilir. Zamansal niteliğin azaltılması ile çok daha fazla bittin tasarruf edilebileceğinden bu yaklaşım konumsal niteliğin azaltılmasına göre daha etkilidir.

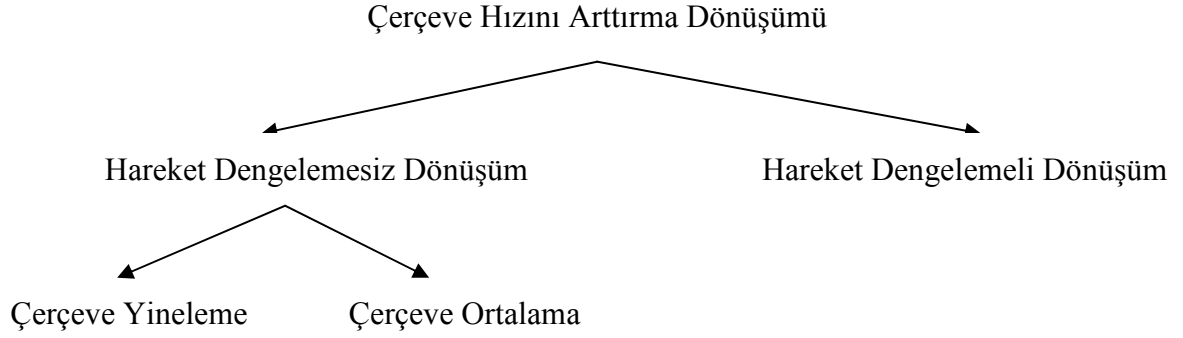
Çerçeve hızı azaltılırken, örneğin, 30 çerçeve/sn'lik tam çerçeve hızı için, çerçeve hızı her 3 çerçeveden 2'sinin atılmasıyla, 10 çerçeve/sn'ye düşürülebilir. Videonun çerçeve hızının azaltılarak aktarılması ve gösterilmesi sonucunda görüntüdeki hareketler kesintili gözükür. Bu nedenle videonun görüntüleme sırasında özgün çerçeve hızında ya da daha yüksek bir çerçeve hızında görüntülenmesi gerekmektedir. Bunun için çerçeve hızını artırma dönüşümü olarak da bilinen zamansal çerçeve aradeğerlemesi yöntemlerinden biri uygulanarak videonun niteliği iyileştirilebilir.



Şekil 5.1 Çerçeve aradeğerlemesi

Çerçeve aradeğerlemesinde istenilen çerçeve hızını elde edebilmek için özgün çerçevelerden

yenileri üretilir. Aradeğerleme yapıldığında belirgin bir hesaplama yükü ortaya çıkarmadan, atlanılan çerçevelerin ne kadar doğrulukla yeniden oluşturuldukları temel sorundur.

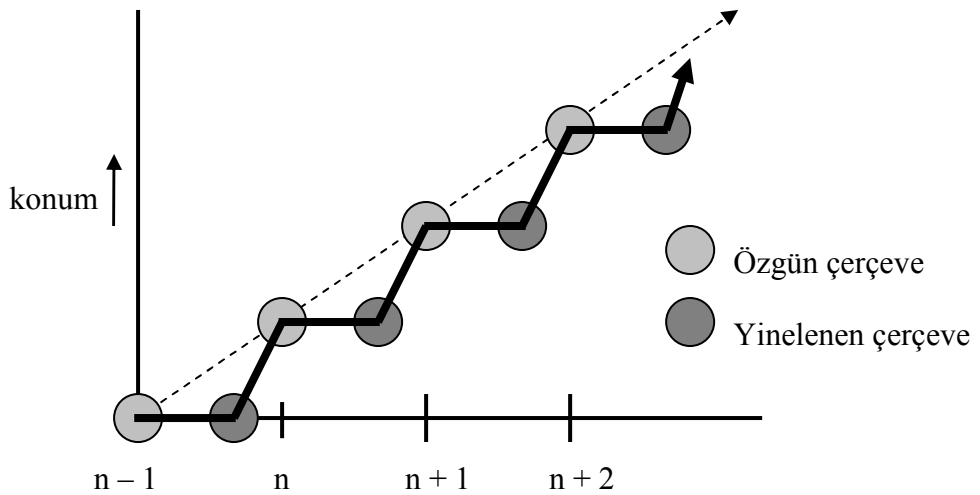


Şekil 5.2 Çerçeve aradeğerleme yöntemleri

Çerçeve aradeğerleme yöntemleri genel olarak iki gruba ayrılabilir (Şekil 5.2). İlk gruptaki yöntemlerde hareket bilgisi dikkate alınmadan video çerçeveleri yeniden oluşturulur. İkinci grup yöntemlerde ise hareket dengelemeli dönüştürme yöntemleri kullanılmaktadır.

5.1 Hareket Dengelemesiz Çerçeve Aradeğerlemesi

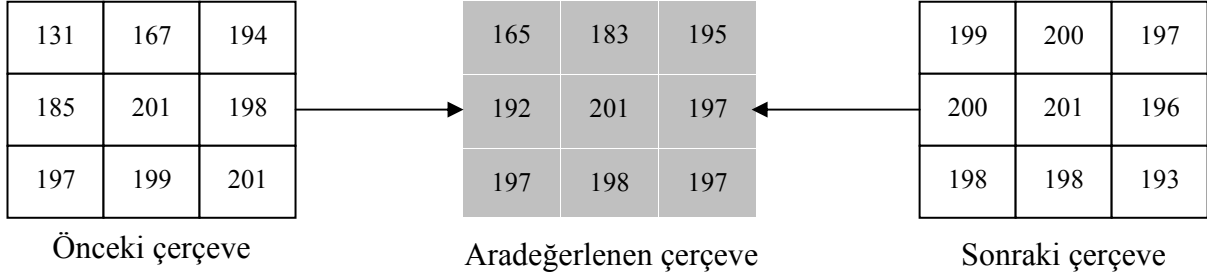
Hareket dengelemesinin yapılmadığı yöntemlerde nesnelerin hareketleri dikkate alınmaksızın yeni çerçeveler oluşturularak istenen çerçeve hızı elde edilir. Yöntemlerde hareket dikkate alınmadığından hareketin hiç olmadığı ya da az olduğu videolarda başarılı bir çerçeve aradeğerlemesi yapılabilir. Bu yalın yöntemler insan gözünün oldukça hassas olduğu, istenmeyen kesintili ya da bulanık görüntülere neden olmaktadır.



Şekil 5.3 Çerçeve yineleme

5.1.1 Çerçeve Yineleme

En yalın çerçeve aradeğerleme yöntemi çalışılan özgün çerçevenin bir sonrakine kadar istenen çerçeve hızına göre yinelenmesidir (Şekil 5.3). Çerçeve yineleme sonucunda özellikle büyük ya da karmaşık hareketlerin olduğu alanlarda görüntü dizisi kesikli olacaktır.



Şekil 5.4 Çerçeve ortalaması ile aradeğerlenen çerçevenin piksellerinin belirlenmesi

5.1.2 Çerçeve Ortalama

Çerçeve ortalamasında ara çerçevelere ilişkin pikseller, bir önceki ve bir sonraki özgün çerçevelerdeki aynı konumda bulunan karşılıklı piksellerin aritmetik ortalaması alınarak belirlenir (Şekil 5.4).



Şekil 5.5 Çerçeve ortalaması ile aradeğerlemede, çerçevelerdeki hareket durumuna göre niteliğin değişmesi (a) Az hareketli video çerçeveleri (b) Çok hareketli video çerçeveleri

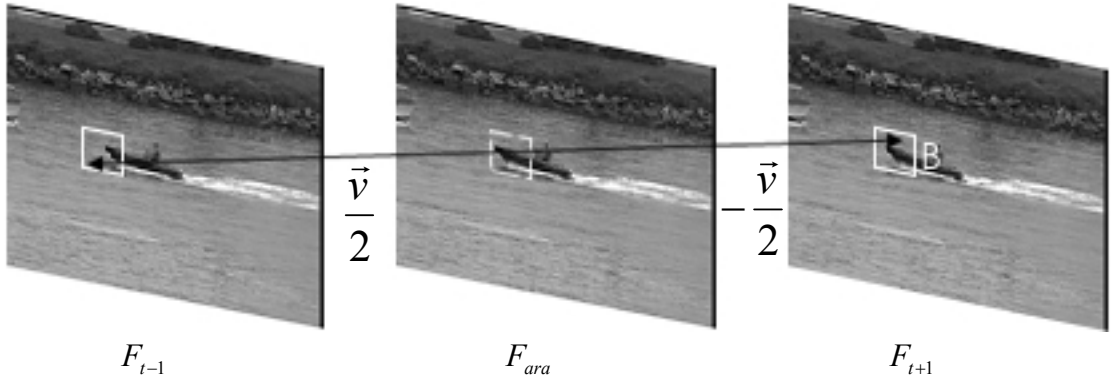
Önceki ve sonraki çerçeve arasında hareketin hiç olmadığı ya da az olduğu videolarda çerçeve ortalaması iyi bir sonuç vermektedir. Ancak Şekil 5.5(b)'de gösterildiği gibi büyük ya da

karmaşık hareketlerin olduğu videolarda aradeğerlenen çerçevelerde elde edilen görüntü bulanıktır.

5.2 Hareket Dengelemeli Çerçeve Aradeğerleme (HDÇA)

Hareket dengelemesiz yöntemlerde, video çerçevelerinde kesintili ya da bulanık alanlar gözlemlendiğinden aradeğerleme başarımını arttırmak için hareketi de dikkate alan hareket dengelemeli çerçeve aradeğerleme yöntemleri geliştirilmiştir. HDÇA yöntemleri video görüntüsündeki hareketin düzgün, sürekli ve ötelemeli olduğu varsayımına dayanmaktadır.

HDÇA yöntemlerinde aradeğerlenen çerçeve birbirleriyle örtüşmeyen bloklara ayrılır ve her bir blok için önceki ve sonraki çerçevelerde karşılık gelen bloklar arasında hareket kestirimi yapılır. Aradeğerlenen çerçevedeki çalışılan bloğa ilişkin pikseller, bloğa atanan hareket vektörünün yarısının belirttiği önceki ve sonraki bloklardaki piksellerin aritmetik ortalaması alınarak belirlenir.



Şekil 5.6 Hareket dengelemeli çerçeve aradeğerleme

F_{t-1} ve F_{t+1} çerçevelerindeki bilgi kullanılarak F_{ara} çerçevesi şöyle elde edilebilir: Şekil 5.6'da gösterildiği gibi, B bloğu F_{t-1} çerçevesinden F_{t+1} çerçevesine $\vec{v}$ vektörü kadar hareket ederse, B bloğunun doğrusal aradeğerleme ile F_{t-1} çerçevesinden F_{ara} çerçevesine $\vec{v}_i/2$ vektörü kadar hareket ettiği varsayılır. $\vec{p} = [x, y]^T$ herhangi bir pikselin konumunu, $I(\vec{p}, t)$ t zamanında $[x, y]$ pikselinin yeğinliğini ve $\vec{v}$ B bloğunun hareket vektörünü belirtmek üzere (5.1)'deki eşitlikler yazılabilir.

$$I(\vec{p} - \vec{v}, t - 1) = I\left(\vec{p} - \frac{\vec{v}}{2}, t\right) = I(\vec{p}, t + 1) \quad \forall \vec{p} \in B \quad (5.1)$$

Bu durumda, F_{ara} çerçevesindeki bir bloğun F_{t-1} ve F_{t+1} çerçevelerinden aradeğerlenmesi (5.2)'deki gibi yapılabilir.

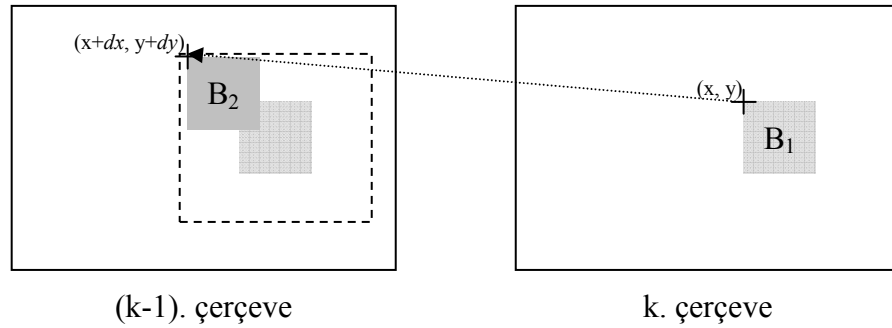
$$\tilde{I}\left(\bar{p} - \frac{\bar{v}}{2}, t\right) = \frac{1}{2} \{I(\bar{p} - \bar{v}, t-1) + I(\bar{p}, t+1)\} \quad \forall \bar{p} \in B \quad (5.2)$$

(5.2)'de $\tilde{I}(\bar{p}, t)$ t zamanında $[x, y]$ pikselinin yeniden oluşturulan yeğinliğini (ya da şiddetini) belirtmektedir (Chen vd., 1998).

5.2.1 HDÇA'nın Uygulanması

Çerçeve numarası k ile gösterildiğinde $(k-1)$.ve k . çerçeveler arasına aradeğerlenmiş çerçeve şöyle yerleştirilebilir: (Wong ve Au, 1995)

- $(k-1)$. çerçeve, k . çerçeve ve aradeğerlenen çerçeve birbiriyle örtüşmeyen, $N \times N$ 'lik bloklara ayrılır.
- İlk olarak ileri hareket kestirimi yapılır. k . çerçevede (x, y) konumunda bulunan B_1 bloğu ile, $(k-1)$. çerçevede tanımlanan arama penceresinde en iyi eşleşen blok aranır (Şekil 5.7)



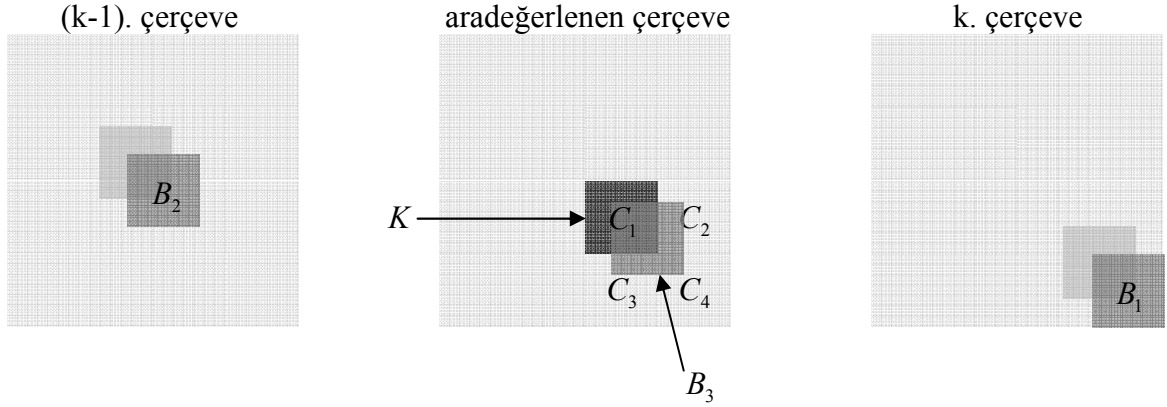
Şekil 5.7 Çerçeve aradeğerlemede ileri hareket kestirimi

$(k-1)$. çerçevede en iyi eşleşen B_2 bloğunun konumu $(x + dx, y + dy)$ ise, blokların doğrusal öteleme hareketi yaptıkları varsayımına dayanarak, B_2 bloğu aradeğerlenen

çerçevede $\left(x + \frac{dx}{2}, y + \frac{dy}{2}\right)$ konumunda bulunmalıdır. Ne var ki, aradeğerlenen çerçevede

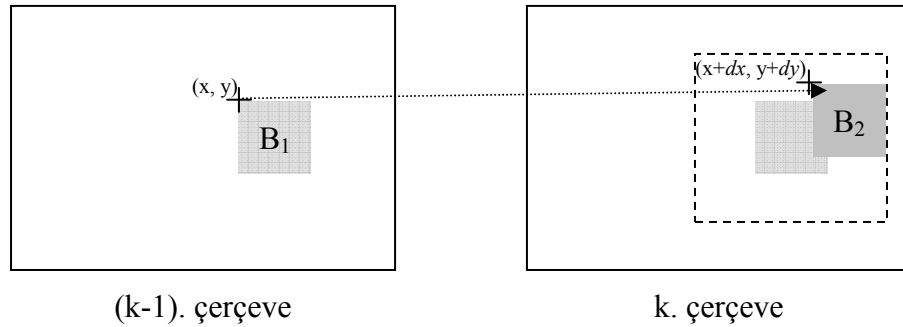
$\left(x + \frac{dx}{2}, y + \frac{dy}{2}\right)$ konumundaki B_3 bloğu blok kafesine tam olarak oturmaz. B_3

bloğu genellikle blok kafesindeki C_1, C_2, C_3 ve C_4 bloklarını kaplayacaktır (Şekil 5.8).



Şekil 5.8 Aradeğerlenen çerçevede hareket vektörünün yarısı kadar ötelenen bloğun blok kafesinde kapladığı bloklar

Bu sorunu çözmek için aradeğerlenen çerçevedeki her bir blok için aday hareket vektörleri listesi tutulur. C_1, C_2, C_3 ve C_4 bloklarının aday hareket vektörleri listelerine $\left(\frac{dx}{2}, \frac{dy}{2}\right)$ ileri hareket vektörü ve B_3 bloğunun C_1, C_2, C_3 ve C_4 bloklarıyla kesişim alanı eklenir. Aradeğerlenen çerçevedeki herhangi bir blok için, daha büyük kesişim alanına sahip hareket vektörü adayları daha küçük kesişim alanına sahip olanlardan daha güvenilirdir.



Şekil 5.9 Çerçeve aradeğerlemesinde geri hareket kestirimi

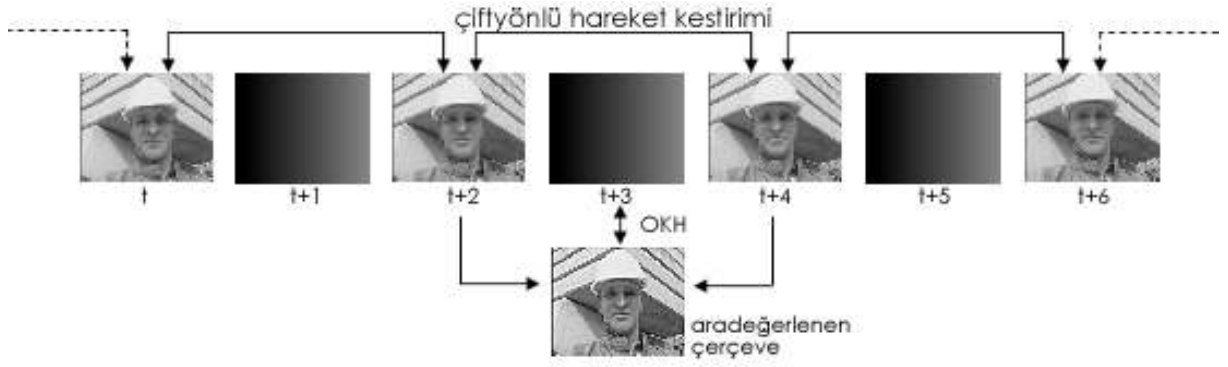
- Geri hareket kestirimi yapılır. $(k-1)$. çerçevede (x, y) konumundaki B_1 bloğu için arama alanı bu kez k . çerçevede oluşturulur (Şekil 5.9). En iyi eşleşen B_2 bloğunun konumu k . çerçevede $(x + dx, y + dy)$ ise, $\left(\frac{dx}{2}, \frac{dy}{2}\right)$ geri hareket vektörü, aradeğerlenen çerçevede

$\left(x + \frac{dx}{2}, y + \frac{dy}{2}\right)$ konumundaki B_3 bloğunun kapladığı blokların hareket vektörü aday listelerine eklenir.

- İleri ve geri hareket kestirimi yapıldıktan sonra aradeğerlenen çerçevedeki her bir C bloğu aday hareket vektörleri listesine sahiptir. Aradeğerlenen çerçevede (i, j) konumundaki C bloğu için en uygun hareket vektörü en büyük kesişim alanını veren hareket vektörüdür. Seçilen hareket vektörünün ilişkin olduğu B_3 bloğu C bloğuna “en yakın” olan bloktur. En uygun aday (v_x, v_y) vektörü ileri hareket vektörü ise, C bloğu $(k-1)$. çerçevede $(i + v_x, j + v_y)$ konumundaki blok ve k . çerçevede $(i - v_x, j - v_y)$ konumundaki bloğun aritmetik ortalaması alınarak hesaplanır. Şekil 5.8’deki C_1 bloğu için en büyük kesişim alanını veren en uygun hareket vektörü ile ilişkilendirilen B_3 bloğu C_1 bloğunun tamamını kapsayamaz; koyu gri ile gösterilen L-şeklindeki K alanı açıkta kalır. K alanının tamamlanması için, K içerisindeki pikseller B_3 bloğu içerisindeki piksellere benzediğinden, C_1 bloğunun B_3 bloğu ile aynı hareket bilgisini paylaştığı varsayılmıştır. $(k-1)$. çerçevede $(i + v_x, j + v_y)$ konumundaki (ya da k . çerçevede $(i - v_x, j - v_y)$ konumundaki) bloğun görüntü sınırlarının dışına çıkması durumunda k . çerçevede $(i - v_x, j - v_y)$ konumundaki (ya da $(k-1)$. çerçevede $(i + v_x, j + v_y)$ konumundaki) blok doğrudan C bloğuna kopyalanır. En uygun aday (v_x, v_y) vektörü geri hareket vektörü ise, k . çerçevede $(i + v_x, j + v_y)$ konumundaki blok ve $(k-1)$. çerçevede $(i - v_x, j - v_y)$ konumundaki bloğun aritmetik ortalaması bulunur. Bloklardan birisinin görüntü sınırlarının dışına çıkması durumunda ileri hareket kestirimine benzer biçimde doğrudan kopyalama yapılır.

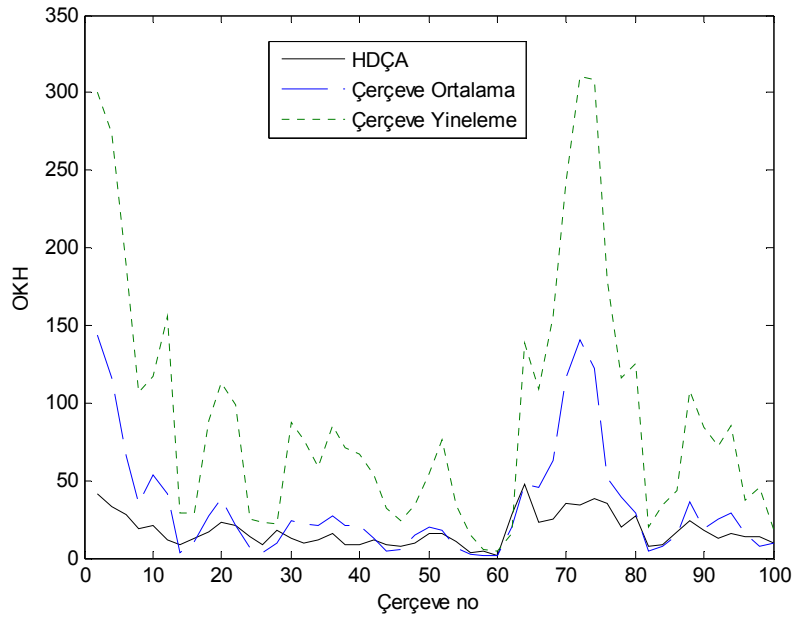
5.2.2 HDÇA’sinden Elde Edilen Sonuçlar

HDÇA algoritmalarının testi için “Foreman”, “News” ve “Salesman” videolarından, zamansal sıralamadaki ilk çerçeveden başlayıp, birer çerçeve atılarak toplam 50 çerçeve alınmıştır. Atılan çerçeveler aradeğerleme ile yeniden oluşturularak özgün çerçeveler ile karşılaştırma yapılmıştır. Uygulanan test yordamı Şekil 5.10’da gösterilmiştir.



Şekil 5.10 HDÇA yöntemlerinin testinde kullanılan yöntem. Görüntü dizisinden atılan çerçeve, aradeğerleme ile yeniden oluşturularak özgün çerçeve ile karşılaştırılıyor

Çift yönlü hareket kestirimi ile HDÇA algoritması hareket dengelemesiz yöntemlerden çerçeve yineleme ve çerçeve ortalama ile karşılaştırılmıştır. HDÇA için hareket kestiriminde blok boyutu 8×8 ve arama parametresi 7 alınmıştır. İleri ve geri hareket kestirimlerinde EA kullanılmıştır.



Şekil 5.11 Foreman videosu kullanılarak yapılan aradeğerlemede yöntemlerin OKH başarımları

Foreman videosunda aradeğerleme yöntemlerinin OKH başarımları Şekil 5.11'de gösterilmiştir. Yöntemlerin Foreman, News ve Salesman videoları için aradeğerlenen çerçeve başına PSNR başarımları Çizelge 5.1'de saptanmıştır.

Çizelge 5.1 Çerçeve aradeğerleme yöntemlerinin PSNR başarımı (dB)

	Çerçeve Yineleme	Çerçeve Ortalama	HDÇA
Foreman	28.49	32.96	35.64
News	34.98	37.47	37.17
Salesman	38.55	42.18	43.51

Hareket kestirimi yöntemlerinde olduğu gibi video çerçeve aradeğerleme yöntemleri de videonun hareket içeriğine duyarlıdır. Foreman gibi çok hareketli bir videoda hareket dengelemeli yöntem diğer iki yönteme göre daha iyi bir başarımlı göstermektedir. News ve Salesman videoları görece az hareketli videolardır. Buna karşın HDÇA'nın News videosunda başarımlı çerçeve ortalamasına göre biraz düşmektedir. Bu sonuç News videosunda sınırlı bir alandaki karmaşık hareket içeriğinden kaynaklanmaktadır. Salesman videosunda hareket içeriği daha yalın ama arkaplan karmaşık olduğundan HDÇA çerçeve ortalamaya göre çok daha iyi bir başarımlı göstermektedir.

Çizelge 5.2 HDÇA'de tekyönlü hareket kestirimi sonucu PSNR başarımının düşmesi (dB)

	Tekyönlü Hareket Kestirimi	Çiftyönlü Hareket Kestirimi
Foreman	34.99	35.64
News	36.47	37.17
Salesman	42.90	43.90

Gerçekleştirilen HDÇA algoritmasında hareket vektörlerinin özellikle nesne sınırlarındaki gerçek hareketi gösterebilmesi için çiftyönlü hareket kestirimi yapılmaktadır. Hareket kestirimi tekyönlü yapıldığında (yalnızca ileri hareket kestirimi) PSNR başarımının düştüğü Çizelge 5.2'de saptanmıştır.

6. SONUÇ ve ÖNERİLER

Bu çalışmada hareket kestirimi yöntemleri ve video çerçeve aradeğerlemesi ele alınmıştır. Bilinen hareket kestirimi yöntemlerinden TA, ÜAA, YÜAA, DAA ve EA algoritmaları gerçekleştirilerek başarımları incelenmiştir. Hareket kestirimi yöntemlerinden elde edilen görsel niteliğin iyileştirilmesi için bir hareket kestirimi yöntemi önerilmiştir. Önerilen yöntemde iki video çerçevesi arasındaki çok hareketli ve az hareketli bölgelerde farklı parametrelerle hareket kestirimi uygulanmıştır. Böylelikle hareket vektörlerinin yanlış hesaplanması olasılığının yüksek olduğu hareket bölgelerinde daha hassas bir hareket kestirimi tekniği uygulanmıştır.

Hareket kestiriminin ardışıl iki çerçeve arasında tekyönlü yapıldığı durumda özellikle sahnedeki nesnelerin sınırlarında hareket vektörleri yanlış hesaplanabilmektedir. Nesnelerin gerçek hareketlerini gösteren hareket vektörlerinin elde edilmesi için video çerçeve aradeğerlemesi çiftyönlü hareket kestirimi ile gerçekleştirilmiştir.

Çalışılan bloğun referans çerçevede oluşturulan arama penceresinde arandığı hareket kestirimi yöntemlerinden TA diğer hızlı yöntemlere göre daha yüksek PSNR başarımına sahiptir. Çünkü TA'da arama penceresindeki tüm aday bloklara bakıldığı için hata yüzeyindeki global minimumun bulunması güvence altındadır. Buna karşılık TA'nın zamansal karmaşıklığı oldukça yüksektir. Hızlı hareket kestirimi yöntemlerinde, algoritmada belirlenen arama örüntüsüne göre arama penceresindeki bazı bloklar ile karşılaştırma yapıldığı için global minimumun kaçırılarak yerel minimuma yakalanılması olasılığı söz konusudur. Bu nedenle hızlı hareket kestirimi yöntemlerinde TA'ya göre PSNR başarımından ödün verilirken TA'nın zamansal karmaşıklığı büyük ölçüde iyileştirilmektedir. Arama parametresinin 7 alınarak çok hareketli bir videoda uygulanan EA'da, PSNR başarımı TA'ya göre yaklaşık olarak 0.8 dB düşerken, TA'da 225 olan çerçeve başına ortalama blok karşılaştırma sayısı EA'da 16'ya düşmektedir. EA'da PSNR başarımının düşmesine karşılık düşük zaman maliyetli hareket kestirimi ile kabul edilebilir bir sonuç alınmaktadır.

Gerçek videolardaki blok hareket vektörlerinin merkez-yanlı özelliğiyle uyumlu arama örüntüleri kullanan hızlı hareket yöntemlerinde daha iyi bir başarımlar elde edilmektedir. ÜAA algoritmasında merkez-yanlı bir arama örüntüsü kullanılmadığından durağan ya da neredeyse-durağan bloklardaki küçük hareketlerin belirlenmesi zorlaşmaktadır. ÜAA'da arama yolunun global minimumdan uzaklaşarak kolaylıkla yerel minimuma yakalanılması söz konusudur. Merkez-yanlı arama örüntüleri kullanan YÜAA ve EA görece az hareketli bir videoda PSNR

başarımını ÜAA'ya göre sırasıyla 0.25 ve 0.12 dB iyileştirmektedir.

Hızlı hareket kestirimi yöntemleri ile ilgili şu saptamalar yapılabilir: Hareket kestirimi yöntemlerinin başarısı videonun hareket içeriğiyle doğrudan ilişkilidir. Genellikle çok hareketin olduğu bir videoda hareket kestirimi başarısı düşerken az hareketli videolarda hareket kestirimi başarısı yüksek olmaktadır. Hızlı BEA'nda farklı şekil ya da boyutlardaki arama örüntülerinin ve hareket vektörü dağılımının merkez-yanlı özelliğinin arama hızı ve eşleştirme başarısı üzerinde büyük bir etkisi vardır.

Hareket kestiriminin başarısını arttırmak için önerilen yöntemde blok boyutu ve arama parametresi değişkenleri çerçevelerdeki hareket içeriğine göre uyarlamalı hale getirilmiştir. Çalışılan çerçevedeki bloklar “hareketli” ve “neredeyse-durağan” bloklar olmak üzere iki sınıfa ayrılmıştır. “Hareketli” bloklar alt bloklara ayrılarak daha büyük bir arama penceresinde, her bir alt blok için hareket vektörleri hesaplanmıştır. “Neredeyse-durağan” bloklar için tam blok boyutu ile daha küçük bir arama penceresinde hareket kestirimi yapılmıştır. Referans çerçeveden çalışılan çerçeveye belirgin olarak hareket eden bölgelerde daha küçük blok boyutu kullanılarak hareket vektörleri hesaplandığından hareket kestirimi başarısı artmaktadır. Hareketin görece az olduğunun belirlendiği bölgelerde ise hareket vektörünün yüksek olasılıkla merkez civarında bir bloğu göstereceği varsayılmıştır. Buna bağlı olarak arama penceresi yeteri kadar küçültülerek “neredeyse-durağan” bloklar için arama maliyetinin azaltılması amaçlanmıştır.

Önerilen yöntemde alt blokların hareket kestiriminde TA ya da EA kullanılmıştır. TA kullanıldığında, çok hareketli bir video için PSNR başarımı TA'ya göre 1.26 dB, YÜAA'ya göre 1.91 dB, az hareketli bir video için TA'ya göre 1.68 dB, YÜAA'ya göre 1.90 dB geliştirilmiştir. Önerilen yöntemin PSNR başarımı TA dahil diğer yöntemlere göre belirgin olarak iyileştirilmektedir. Alt blok hareket kestiriminde EA kullanıldığında, çok hareketli bir video için PSNR başarımı TA'ya göre 0.09, YÜAA'ya göre 0.74 dB, az hareketli bir video için TA'ya göre 0.71 dB, YÜAA'ya göre 0.94 dB geliştirilmiştir. Alt blok hareket kestiriminde EA kullanıldığında yöntemin başarısının TA'ya yaklaşmasının nedeni, EA'da bazı alt bloklar için global minimum noktadan uzaklaşılmasıdır. Alt blok hareket kestirimi EA gibi hızlı bir blok eşleştirme algoritması ile gerçekleştirildiğinde yöntemin bu biçimiyle hızlı BEA ile karşılaştırılması uygun olacaktır. Önerilen yöntem videonun hareket içeriğinden bağımsız olarak hızlı BEA'ndan daha iyi bir başarımla göstermektedir.

Önerilen yöntemde fark çerçevesinden yararlanılarak blokların sınıflandırılmasında kullanılan

eşik değeri otomatik olarak belirlenmektedir. Fark çerçevesine uygulanan eşik değeri tüm çerçeve için sabittir. Fark çerçevesindeki farklı bölgelerde hareket içeriği değişebileceğinden yerel bir eşikleme yöntemiyle blokların sınıflandırılma başarısı artırılabilir.

Video çerçeve aradeğerlemesinde hesaplanan hareket vektörlerinin nesnelerin gerçek hareketlerini göstermeleri gerekmektedir. Bu nedenle gerçekleştirilen HDÇA yönteminde çift yönlü hareket kestirimi ile bir blok için gerçek hareket vektörünün hesaplanması amaçlanmaktadır. Video çerçeve aradeğerleme algoritmalarının başarımı videoların hareket içerikleri ile ilintilidir. Çok hareketli videolarda HDÇA yöntemleri hareket dengelemesiz yöntemlere üstünlük sağlamaktadır. Gerçekleştirilen HDÇA yönteminin PSNR başarımı çok hareketli bir videoda çerçeve yinelemeye göre 7.15 dB, çerçeve ortalama göre 2.67 dB daha geliştirilmiştir. Aradaki başarımların farkı az hareketli videolarda azalmaktadır. Görece az hareketli bir videonun karmaşık bir hareket içeriği varsa çerçeve ortalama HDÇA'ya göre daha başarılı olabilmektedir. Böyle bir videoda çerçeve ortalamanın PSNR başarımı HDÇA'ya göre 0.3 dB daha iyidir.

Video çerçeve aradeğerlemesinin zamansal karmaşıklığını azaltmak için pratik olarak belirli bir yöntemle önceki ve sonraki çerçevedeki hareket değişimi belirlenerek, kullanılacak çerçeve aradeğerleme algoritmasına karar verilebilir. Hareket değişiminin az olduğu çerçevelerde doğrudan çerçeve ortalama, çok olduğu çerçevelerde HDÇA algoritması uygulanabilir.

KAYNAKLAR

- Al-Mualla, M. E., Canagarajah, C. N. ve Bull, D.R., (2003), "Reduced Complexity Motion Estimation Techniques: Review And Comparative Study", Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, Volume 2, Issue 14-17, Page(s): 607-610.
- Bigun, J., (2006), Vision with Direction, Springer-Verlag Berlin Heidelberg.
- Chen, Y. -K. ve Chen, T. -C., (2004), "A Fast And Effective Method For Motion Estimation", IEEE International Conference on Multimedia and Expo, Volume 1, Issue 27-30, Page(s): 547-550.
- Chen, Y. -K., Vetro, A., Sun, H. ve Kung, S. Y., (1998), "Frame-Rate Up-Conversion Using Transmitted True Motion Vector", IEEE Second Workshop on Multimedia Signal Processing.
- Cheung, C. -H. ve Po, L. -M., (2002), "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", ", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 12, Page(s):1168-1177.
- Choi, B. -T., Lee S. -H. ve Ko, S. -J., (2000), "New Frame Rate Up-Conversion Using Bi-Directional Motion Estimation", IEEE Transactions on Consumer Electronics, Volume 46, Issue 3, Page(s): 603-609.
- Dane, G. ve Nguyen, T. Q., (2004), "Motion Vector Processing For Frame Rate Up Conversion", IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 3, Page(s): iii- 309-12.
- Gonzales, R. C. ve Woods, R. E., (2002), Digital Image Processing, Prentice-Hall, Inc.
- Jack, K., (2005), Video Demystified, Elsevier Inc.
- Kalra, S. ve Chong, M. -N., (1998), "Bidirectional Motion Estimation Via Vector Propagation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, Issue 8, Page(s): 976-987.
- Koga, T., vd, (1981), "Motion Compensated Interframe Coding for Video Conferencing", Proc. Nat. Telecommunications Conf., pp. G5.3.1-G5.3.5.
- Po, L. ve Ma, W., (1996), "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 3, Page(s): 313-317.
- Reoxiang, L., Bing, Z. ve Liou, M. L., (1994), "A New Three-step Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Volume 4, Issue 4, Page(s): 438-442.
- Richardson, Iain E. G., (2003), H.264 and MPEG-4 Video Compression, John Wiley and Sons Ltd.
- Richardson, Iain E. G., (2002), Video Codec Design, John Wiley and Sons Ltd.
- Salomon, D., (2002), Data Compression Methods, Springer-Verlag New York, Inc.
- Shapiro, L. G. ve Stockman, G. C., (2001), Computer Vision, Prentice-Hall, Inc.
- Wang, J., Patel, N. ve Grosky, W., (2004), "A Fast Block-Based Motion Compensation

Video Frame Interpolation”, Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, Volume 2, Issue 7-10, Page(s): 1740-1743.

Wang, Y., Ostermann, J. ve Zhang, Y., (2002), Video Processing and Communications, Prentice-Hall, Inc, Signal Processing Series.

Wong, C. –K. ve Au, O. C., (1995), “Fast Motion Compensated Temporal Interpolation for Video”, IEEE Transactions on Computers, Vol.44, No.3, Page(s): 453-457.

Zhai, J., Yu, K., Li, J. ve Li, S., (2005), “A Low Complexity Motion Compensated Frame Interpolation Method”, IEEE International Symposium on Circuits and Systems, Kobe, Japan.

Zhang, D. ve Lu, G., (2001), “Segmentation of Moving Objects in Image Sequence: A Survey”, in Circuit, Systems, and Signal Processing, Page(s): 143-183.

Zhang, X., Al, H., Hu, R. ve Li, D., (2004), “A Novel Algorithm For Sub-Pixel Block Motion Estimation”, Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, Page(s): 587-590, Hong Kong.

Zhu, S., ve Ma, K. –K., (2000), “A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation”, IEEE Trans. Image Processing, vol. 9, Page(s): 287-290.

ÖZGEÇMİŞ

Doğum tarihi 07.03.1979

Doğum yeri İstanbul

Lise 1994-1997 Fenerbahçe Lisesi

Lisans 1997-2003 Yıldız Teknik Üniversitesi Kimya-Metalurji Fak.
Metalurji Mühendisliği Bölümü

Yüksek Lisans 2004-2007 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Müh. Anabilim Dalı,
Bilgisayar Müh. Programı