T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VİDEO SİNYALLERİNDE HAREKET TESPİT UYGULAMASI

Kaan GENÇAY

YÜKSEK LİSANS TEZİ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Programı

Danışman

Doç. Dr. Umut Engin AYTEN

Temmuz, 2019

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VİDEO SİNYALLERİNDE HAREKET TESPİT UYGULAMASI

Kaan GENÇAY tarafından hazırlanan tez çalışması çalışması 04.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Elektronik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Umut Engin AYTEN

Yıldız Teknik Üniversitesi

Danışman

Jüri Üyeleri

Doç. Dr. Umut Engin AYTEN, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Mürvet KIRCI, Üye
İstanbul Teknik Üniversitesi _____

Dr. Öğr. Üyesi Nihan KAHRAMAN, Üye
Yıldız Teknik Üniversitesi _____

Danışmanım Doç. Dr. Umut EnginAYTEN sorumluluğunda tarafımca hazırlanan Kapalı Alan Konum Belirleme başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim

Kaan GENÇAY

İmza

Aileme

ve

arkadaşlarıma

TEŞEKKÜR

Bu tez çalışması süresince destek ve yardımları ile devamlı yanımda olan tez danışmanım Doç. Dr. Umut Engin AYTEN hocama, arkadaşım Çağlar ÇAVDAR'a ve aileme teşekkür ederim.

Kaan GENÇAY

İÇİNDEKİLER

SİMGE LİSTESİ VIII
KISALTMA LİSTESİ IX
ŞEKİL LİSTESİ X
TABLO LİSTESİ XI
ÖZET XII
ABSTRACT XIV
1 Giriş 1
1.1 Literatür Özeti 1
1.2 Tezin Amacı 3
1.3 Hipotez 5
2 Hareket Tespiti 6
2.1 Hareket Tespitine Genel Bakış 6
2.2 Blok Eşleşme Algoritması 7
2.2.1 Tam Arama 9
2.2.2 Üç Adım Arama 11
2.2.3 Yeni Üç Adım Arama 13
2.2.4 Dört Adım Arama 15
2.2.5 Eşkenar Arama 17
2.3 Arama Algoritmalarının Karşılaştırılması 20
3 Önerilen Yöntemin Uygulanması 23
3.1 Tezde Önerilen Yöntem 23
3.2 Görüntü Değişiminin Tespiti 24

3.3 Durağan ve Hareketli Blokların Tespiti 25
3.4 Hareketli Blokların Sınıflandırılması 26
3.5 Görüntü Değişim Matrisi için Adaptif Eşik Değeri Tespiti 26
3.6 Hareketli ve Yarı Hareketli Blokların Belirlenmesi 27
3.7 Hareket Kestiriminin Uygulaması 29
4 Değerlendirme ve Sonuçlar 34
4.1 Önerilen Yöntemin Değerlendirilmesi 34
5 Sonuç ve Öneriler 37
5.1 Sonuç 37
Kaynakça 38
Tezden Üretilmiş Yayınlar 40

SİMGE LİSTESİ

N	Makro blok kenar piksel sayısı
p	Arama paramatresi
A_x, B_x	Vektör yatay bileşeni
A_y, B_y	Vektör dikey bileşeni
Δx	Vektör yatay uzunluğu
Δy	Vektör dikey uzunluğu
n	Makro blok kenar piksel sayısı
i	Piksel yatay indeksi
j	Piksel dikey indeksi
C_{ij}	Mevcut blok piksel indeksi
R_{ij}	Referans blok piksel indeksi
M	Görüntü çerçevesi kenar piksel sayısı
A_{ilk}	Arama Adım Sayısı
x	Varyans değişkeni
Pr	Olasılık fonksiyonu
ss	Adım mesafe fonksiyonu

KISALTMA LİSTESİ

BEA	Blok Eşleşme Algoritması
BMA	Block Matching Algorithm
DAA	Dört Adım Arama
DS	Diamond Search
EA	Eşkenar Arama
FKA	Faz Korelasyon Algoritması
MFT	Mutlak Farkların Toplamı
MPEG-1	Motion Pictures Experts Group - 1
MPEG-4	Motion Pictures Experts Group – 4
NTSS	New Three Step Search
OAA	Optik Akış Algoritması
OKH	Ortalama Karesel Hata
OMF	Ortalama Mutlak Fark
PSNR	Peak-to-peak Signal Noise Ratio
TA	Tam Arama
TAD	Total Absolute Differences
TMF	Toplam Mutlak Fark
ÜAA	Üç Adım Arama
YÜAA	Yeni Üç Adım Arama

ŞEKİL LİSTESİ

Şekil 2.1 Hareket Vektörleri 6
Şekil 2.2 Makro blokların arama alanı içerisinde karşılaştırılması 7
Şekil 2.3 En iyi makro blok eşleşmesinin aranması 8
Şekil 2.4 Referans ve mevcut çerçeve içerisinde makro blok ile arama işlemi 10
Şekil 2.5 ÜAA algoritması adımları 12
Şekil 2.6 YÜAA algoritması adımları 14
Şekil 2.7 DAA algoritması adımları 16
Şekil 2.8 EA algoritmasında aranan tüm noktalar 18
Şekil 2.9 EA algoritması arama paternleri: Büyük ve küçük arama paternleri 18
Şekil 2.10 EA algoritması arama adımları 19
Şekil 2.11 "susie" veri seti için PSNR değerleri 20
Şekil 2.12 "boat" veri seti için PSNR değeleri 21
Şekil 2.13 "foreman" veri seti için PSNR değleri 21
Şekil 3.1 (t) ve (t+1) zamanlarına ait çerçeveler ve fark çerçevesi 25
Şekil 3.2 Fark çerçevesi üzerinden durağan blokların elde edilmesi 25
Şekil 3.3 Ortalama piksel değeri ve Otsu eşik belirleme yönteminin sonuçları 27
Şekil 3.4 Durağan, yarı durağan ve hareketli blokların elde edilme adımları 28
Şekil 3.5 "boat" veri seti üzerinde uygulanan adımlar 30
Şekil 3.6 "foreman" veri seti üzerinde uygulanan adımlar 30
Şekil 3.7 "susie" veri seti üzerinde uygulanan adımlar 30
Şekil 3.8 Önerilen yönteme ait akış diyagramı 32

TABLO LİSTESİ

Tablo 2.1 Arama alanı içerisindeki tüm makro bloklar için OKH değerleri 8

Tablo 2.2 Arama algoritmalarının veri setlerine göre PSNR değerleri 22

Tablo 4.1 70 çerçeve için toplam arama işlemi sayısı 34

Tablo 4.2 70 çerçeve için her makro blok başına düşen ortalama arama işlemi sayısı 34

Tablo 4.3 "foreman" veri setinin tüm çerçeveleri için elde edilen, blok başına düşen arama işlemi sayısı 35

Tablo 4.4 70 çerçeve için ortalama PSNR değerleri 35

ÖZET

Video Sinyallerinde Hareket Tespit Uygulaması

Kaan GENÇAY

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Umut Engin AYTEN

Günümüzde gelişen teknoloji ile birlikte görüntü işleme gibi yüksek işlem maliyeti gerektiren konular yaygınlaşmakta ve uygulanmaktadır. Hareket tespiti, görüntü işleme alanının bir konusudur ve ihtiyaçlardan dolayı popülerliğini korumaktadır. Gelişen teknolojinin sunduğu görüntülü konuşma servisleri, görüntü depolama ihtiyaçları, kablolu ve kablosuz ağlar arasında görüntü aktarma işlemleri yüksek bant genişliği ve depolama alanı gereksinimi doğurmaktadır. Bu gereksinimler göz önünde bulundurulduğunda, video sıkıştırma ve ara değerlendirme yöntemleri gibi veri sıkıştırma algoritmaları önem kazanmaktadır. Video sıkıştırma formatlarında standartlaşan hareket tespit algoritmaları mevcuttur. İyi bir sıkıştırma için yüksek doğruluk oranı gerekirken, bu yoğun işlemin olabilecek en az işlem maliyeti ile elde edilmesi gerekmektedir. Bu tez çalışmasında, bir hareket sınıflandırma yöntemi kullanılarak sınıflandırılan alanlarda farklı arama algoritmalarının kullanılması önerilmiştir. Çalışmada, Blok Eşleşme Algoritması (BEA) ile hareket tespiti gerçekleştirilmiştir. Öncelikle durağan blokların sınıflandırılması için Mutlak Farkların Toplamı (MFT), hareketli ve yarı hareketli alanların sınıflandırılmasında Otsu eşik belirleme yöntemi, arama algoritmaları olarak Yeni Üç Adım Arama (YÜAA) algoritması ve Eşkenar Arama (EA) algoritması kullanılmıştır. Önerilen yöntemin etkinliğini göstermek için farklı

özelliklerdeki “Susie”, “foreman” ve “boat” veri setleri kullanılmıştır. Deneysel sonuçlar, çalışmanın YÜAA ve EA algoritmalarına çok yakın doğruluk oranı sunduğunu ve YÜAA ve EA algoritmalarından daha az işlem maliyeti getirdiğini söylemektedir.

Anahtar Kelimeler: Hareket tespiti, blok eşleşme algoritması, arama algoritması, yeni üç adım arama, eşkenar arama, makro blok.

ABSTRACT

Application of Motion Detection on Video Signals

Kaan GENÇAY

Department of Electronics and Communication Engineering

Master of Science Thesis

Advisor: Assoc. Prof. Umut Engin AYTEN

Nowadays, requiring highly processing costs subjects such as image processing has become widespread with the developing technology. Motion detection is a subject of image processing and is popular due to needs. These services offered by developing technology such as video communication services,image storage needs, image transfer between wired and wireless networks require high bandwidth and storage space. Considering these requirements, data compression algorithms such as video compression and interim evaluation methods are getting important. There are standardized motion detection algorithms in video compression formats. While a high degree of accuracy is required for good compression, this intensive process has to be achieved with the lowest possible processing costs. In this thesis, using different search algorithms is proposed in the fields classified by using a new motion classification method. For motion detection, Block Matching Algorithm (BMA) is used. In order to classify static blocks Total Absolute Differences (TAD) is used, to classify moving and half-moving blocks Otsu threshold determination method is used. New Three Step Search (NTSS) algorithm and Diamond Search (DS) algorithm were used as the search algorithms. “susie”, “foreman” and “boat” data sets were used to demonstrate the effectiveness of the proposed method. Experimental results suggest that the study provides a very close accuracy to the

NTSS and DS algorithms and brings less transaction costs than the NTSS and DS algorithms.

Key words: Motion detection, block matching algorithm, search algorithm, new three step search, diamond search, macro block.

1

Giriş

1.1 Literatür Özeti

Hareket kestirimi için birçok yöntem geliştirilmiştir ve uygulanmaktadır. Geliştirilen yöntemlerden en sık kullanılanı, literatürde Blok Eşleşme Algoritması (BEA) olarak geçen, kıyaslanan iki görseli belli büyüklüklerde bloklara ayırarak bloklar için hareket vektörü arayışı yapılan yöntemdir. Bloklar için hareket vektörü aranırken çeşitli arama algoritmaları kullanılabilir. Literatürde, arama algoritmaları için birçok çalışma bulunmaktadır. Ayrıca blokların sınıflandırılması ile ilgili çalışmalar da yapılarak, arama algoritmalarının veriminin artması üzerine sonuçlar elde edilmiştir. Bu başlık altında, arama algoritmaları ile ilgili yapılan çalışmalar ve hareketli blokların sınıflandırılması ile ilgili çalışmalar kısa özetler halinde anlatılacaktır.

Arama yönteminin en temeli Tam Arama (TA) algoritmasıdır. Karşılaştırılan iki görüntü bloklara ayrıldıktan sonra bir arama alanı belirlenir. Referans alınan görüntüdeki blok, mevcut görüntüde denk geldiği blok ve arama alanında piksel piksel karşılaştırılır. En uygun blok, en az farkın olduğu blok olacak şekilde aramaya devam edilir. Eşleşen blok bulunduktan sonra hareket vektörü elde edilir. Bu, oldukça maliyetli bir yöntemdir çünkü blok ve arama alanındaki tüm pikseller için bir arama işlemi söz konusudur. Fakat elde ettiği doğruluk oranına bakıldığında, oldukça yüksek olduğu görülmektedir ve diğer tüm hızlı arama algoritmaları, TA algoritmasının doğruluk oranını sağlamaya çalışmaktadır.

Hızlı arama algoritmalarına örnek, ilk defa 1981 yılında J. Jain ve A. Jain tarafından Üç Adım Arama (ÜAA) algoritmasının önerilmesi ile olmuştur. Çalışmada, en iyi eşleşen bloğu üç adımda bulabilen bir algoritma önerilmiştir [1]. Buna göre, blok merkez pikselini merkez alan bir kare belirleyerek karenin köşe ve kenar orta noktalarından arama başlanır, en düşük Ortalama Karesel Hata (OKH) değerine

sahip nokta bulunur. Bulunan nokta, yeni oluşturulacak karenin orta noktası olur ve karenin alanı daraltılarak arama işlemi tekrarlanır. Aynı şekilde, bulunan en düşük OKH değerlikli nokta merkez seçilir ve alanı daraltılarak oluşturulan yeni karenin merkezi olur. Son adım olan üçüncü adımda bulunan en düşük OKH değerlikli piksel, hareket vektörünün sonlandığı noktalar olarak elde edilir. Renxiang Li vd. 1994 yılında ÜAA algoritmasının yeni bir çeşidi üzerine çalışma yapmışlardır. YÜAA adıyla anılan bu algoritmanın özellikleri, arama işleminin hareket vektörü dağılımına uyarlanabilir hale getirilmesiyle elde edilen birinci basamakta merkez taraflı bir kontrol noktası deseni kullanması ve hesaplama maliyetini düşürmek için yarı durdurma tekniği kullanmasıdır. ÜAA ile karşılaştırıldığında, daha küçük hareketlerde hata telafisi ürettiği ve uyumlu bir hesaplama karmaşıklığına sahip olduğu görülmektedir [2].

Lai-Man Po ve Wing-Chung Ma tarafından 1996 yılında geliştirilen yöntemde, gerçek dünyadaki görüntü dizisinin merkez taraflı hareket vektörü dağılımının karakteristiğine dayanarak, hızlı blok hareket tahmini için merkez taraflı kontrol noktası desenine sahip yeni bir Dört Adım Arama (DAA) algoritması önerilmiştir. Bu algoritmada, 2'den 4'e kadar olan arama adımlarını içeren bir yarı durdurma tekniği kullanılmış ve toplam kontrol noktası sayısı 17 ila 27 arasında değişecek şekilde tasarlanmıştır [3].

Shan Zu ve Kai-Kuang Ma 2000 yılındı yaptıkları çalışmada Eşkenar Arama (EA) algoritmasını önermişlerdir. Bu algoritmanın, ÜAA algoritmasından daha iyi performans gösterdiği görülmüştür ve ayrıca YÜAA algoritmasıyla karşılaştırıldığında, EA algoritması yakın performansa ulaşmış, ancak ortalama olarak %22'ye kadar daha az hesaplama gerektirdiği görülmüştür. EA algoritması bu özellikleri ile MPEG-4 video kodlaması için hızlı hareket tahmin algoritması olarak kabul edilmiştir. [4].

1993 yılında Bede Liu ve Andre Zaccarin, yeni bir yaklaşım ile bir algoritma önermişlerdir. Yaptıkları çalışma, hareket alanı ve piksel alt örneklemesine dayanmaktadır. Önce blokların bir kısmı için hareket vektörleri tahmin edilerek alt örneklenmiş bir hareket alanı belirlenir. Bu bloklar için hareket vektörleri, aranan herhangi bir konumdaki piksellerin sadece bir kısmı kullanılarak ve piksel alt

örnekleme modellerini aranan konumlarla değiştirerek belirlenir. Daha sonra alt örneklenmiş hareket alanı enterpolasyonlanır, böylece her piksel bloğu için bir hareket vektörü belirlenir. Önerilen algoritma, MPEG-1 standardı ile uyumlu şekilde çalışmaktadır [5].

Prabhudev I. Hosur ve Kai-Kuang Ma 1999 yılında yaptıkları çalışmada, yerel hareket vektörlerinin mekansal dağılımı hakkında bilgi kullanan hızlı bir blok eşleştirme hareket kestirimi algoritması sunmaktadır. Çalışmada, her bir makro blok için hem arama merkezi hem de arama stratejisi, hareket tahmininin doğruluğunu arttırmak ve yüksek hesaplama kazancı elde etmek için yerel hareket vektörlerinin komşu bloğu uzunluklarına dayanarak belirlenmiştir. Önerilen yöntemin sonuçları, tam arama tarafından sağlananlara yakın amaç ve öznel kaliteye ulaştığını ve aynı zamanda oldukça yüksek bir arama hızı elde ettiğini göstermiştir [6].

2012 yılında SHAN LI vd. hızlı BEA ile ilgili sundukları çalışmada, referans çerçeve dokusu bilgisi ve başlangıçta makro blok artık değeri ile çerçeve-seviye hareket karmaşıklığı uyarlamalı olarak tahmin edilmiş ve daha sonra vektör alanının uzaysal-zamansal korelasyonuna göre blok-seviye hareket karmaşıklığı tahmin edilmiştir. Sabit bloğun aranmasını durdurmak için eşik uygulanmaktadır. İlk arama noktası ve farklı EA arama modları, durağan olmayan blok için hareket türüne göre uyarlamalı olarak seçilmiştir. Önerilen çalışmada elde edilen sonuçlar, diğer algoritmalara göre daha iyi performans göstermiştir [7].

1.2 Tezin Amacı

Günümüzde teknolojinin gelişmesi ile birlikte donanım maliyetleri azalmış ve işlem hızı gelişmiş bilgisayarlar yaygınlaşmıştır. Bununla birlikte yoğun işlemlerin yapılabildiği sistemlerin de kullanımında artış olmuştur. Görüntü işleme, yoğun işlem gerektiren bir konudur ve işlem hızı yüksek bilgisayarların yaygınlaşması ile görüntü işlemenin sistemlere dahil edilmesi kolaylaşmıştır. Gelişen teknoloji, daha kaliteli ve hızlı görüntü kayıtlarının ve aktarımlarının önünü açmıştır. Video görüntülerinin sabit belleklerde kaydedilmesi, kablolu ve kablosuz ağlarda paylaşımı da bu oranda artmıştır. Bu artış, video görüntülerinin boyutlarında

sıkıştırma yapılması ihtiyacı doğurmuş ve bu nedenle çeşitli video sıkıştırma algoritmaları geliştirilmiştir.

Video görüntülerindeki sıkıştırma algoritmalarında, video görüntüsündeki çerçeve dizilerinin birbirleri ile kıyaslanması ve görüntülerdeki hareketlerin tespiti kullanılmaktadır. Bu sayede, iki görüntü çerçevesi arasındaki hareketli alanlar tespit edilerek, sadece hareketli alanların kodlanması ile sıkıştırma yapılabilmektedir. Bu yöntem, her çerçevenin ayrı ayrı kodlanmasından çok daha etkin bir yöntem olarak tercih edilmektedir.

Video görüntülerinin bant genişliği sorunu veya bilinçli olarak çerçeve hızının düşürülmesi ile kaydedilmesi veya aktarılması sonucunda, görüntü dizilerinde bozukluklar meydana gelir. Düşük çerçeve hızlı bir görüntüdeki hareketli nesneler, süreklilik niteliğini kaybeder ve nesnelerde sıçramalar, ani yer değiştirmeleri gibi süreksizlikler algılanmaya başlanır. Bu sorunun giderilmesi için de görüntü işleme yöntemleri mevcuttur. Art arda iki görüntü çerçevesinde hareket kestirimi yapılarak, hareketli alanlar belirlenir ve video ara değerlemesi olarak adlandırılan, iki görüntü çerçevesi arasında yeni bir çerçeve ilave edilir. Bu şekilde video çerçeve hızı arttırılır ve süreksizlik sorunu iyileştirilebilir.

Bu tez çalışmasında, video görüntülerinin sıkıştırılması ve ara değerlemesi için kullanılan hareket tespit algoritmasında iyileştirme amaçlanmıştır. Hareket tespiti için BEA algoritması kullanılmış, algoritmada kullanılan blok arama yöntemleri için ise TA, ÜAA, YÜAA, DAA, EA arama algoritmaları denenmiştir. Video görüntülerinden alınan çerçeveler birbirleri ile kıyaslanarak hareket vektörleri elde edilmiştir. Kıyaslama işlemi yapılırken, oluşturulan fark çerçevesi ve görüntü değişim ikili matrisi üzerinde işlemler yapılarak, görüntü üzerinde bloklar oluşturulmuş ve bu bloklar durağan, yarı hareketli ve hareketli bloklar olarak sınıflandırılmıştır. Hareket tespiti yapılırken sınıflandırılan bloklar ayrı ayrı ele alınarak değerlendirilmiş ve tüm çerçeve üzerinde farklı arama algoritmaları aynı anda kullanılmıştır. Deneysel sonuçlar bize, önerilen çalışmanın tek bir arama algoritması kullanan sistemlerle kıyaslandığında, çok yakın doğruluk oranları ile işlem maliyeti açısından yüksek başarım elde ettiğini söylemektedir.

1.3 Hipotez

Video sıkıştırma ve ara değerlemesi gibi konularda kullanılan hareket tespit algoritmaları, sıklıkla ihtiyaç duyulan bir konudur. Literatürde şu ana kadar yapılan hareket tespit çalışmalarında, az hareketli video görüntülerinde ve sabit arka plan içeren görüntülerde başarılı sonuçlar elde edilmektedir. Bununla birlikte görüntülerdeki hareketin artmasıyla, arka plan ve ön planın aynı anda hareket ettiği görüntülerde, ışık koşullarının uygun olarak kabul edilen seviyeleri ihlal etmesiyle, elde edilen sonuçlar bozulma eğiliminde olmaktadır.

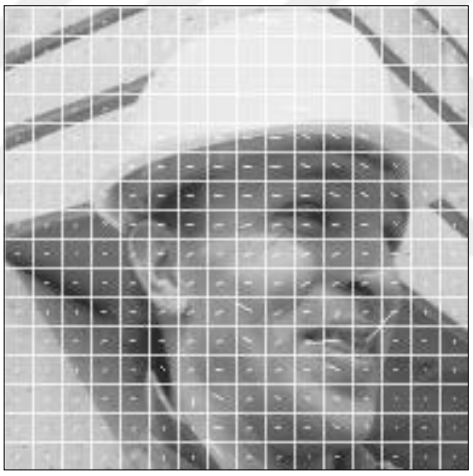
Literatürde video sıkıştırma veya ara değerlemesi için hareket tespiti ile ilgli çalışmalar mevcuttur. Bu çalışmalar başarılı sonuçlar vermektedir fakat işlem maliyetleri oldukça yüksektir. Bu tez çalışmasında geliştirilen yöntemle hareket tespit işleminin maliyeti ve süresi azaltılmaya ve yüksek doğruluk oranı elde edilmeye çalışılmıştır. Görüntüler ön işlemden geçirilerek hareketin hangi alanlarda yoğunlaştığı tespit edilmiş, buna göre hareket tespit algoritması uygulanmıştır. Böylece hareket tespit algoritması için gereken işlem maliyetinin azaltılması ve elde edilen başarım oranının yüksek tutulması hedeflenmiştir. Geliştirilen yöntemlerden bu tez kapsamında ayrıntılı şekilde bahsedilecektir.

2

Hareket Tespiti

2.1 Hareket Tespitine Genel Bakış

Görüntü dizilerinde, ardışık çerçeveler aynı nesneyi içerebilir ve bu nesne bir önceki çerçevedeki konumundan farklı bir konuma hareket etmiş olabilir. Aynı şekilde nesne sabit kalmış ve arka plan hareket etmiş veya hem nesne hem de arka plan aynı anda hareket etmiş olabilir. Bu gibi durumlarda, bu görüntü dizilerinde hareket var denilir [8].



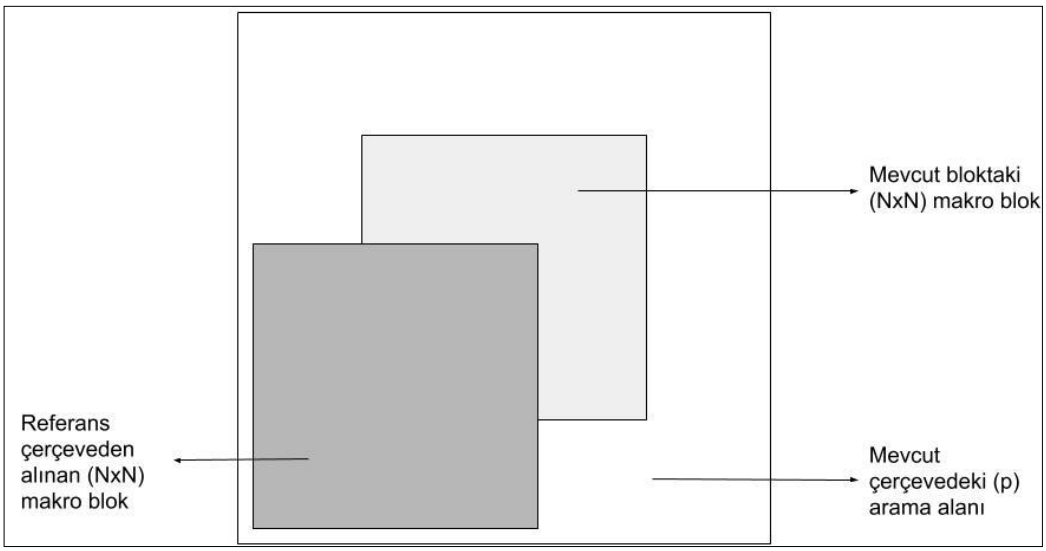
Şekil 2.1 Hareket Vektörleri [9]

Görüntü dizisi, sistemlere aktarılırken, kaydedilirken veya sistemlerde gösterilirken, hareketli dizilerinin tespit edilmesi gerekebilir. Bu ihtiyaç doğrultusunda, görüntü dizilerindeki ardışık çerçeveler, birbirleri ile karşılaştırılarak hareketli kısımları bulunur ve hareket kestirimi yapılmış olur. Görüntü dizilerindeki hareketli kısımların bulunması için, ardışık çerçevelerden bir alan seçilir ve bu alandaki görüntü, bir sonraki çerçevede aranır. Bu yöntemle, Şekil 2.1'deki gibi hareket vektörleri elde edilerek görüntü dizisindeki hareketli alanlar tespit edilmiş olunur. Hareket tespiti için birçok yöntem önerilmiştir. Bunlardan bazıları BEA, Faz Korelasyon Algoritması (FKA) ve Optik Akış

Algoritmasıdır (OAA). Önerilen yöntemde BEA ve çeşitli arama algoritmaları kullanılmıştır.

2.2 Blok Eşleşme Algoritması

BEA, hareket kestiriminde kullanılan en temel yöntemlerden biridir. Yöntemde, mevcut görüntü çerçevesi makro bloklara ayrılarak işlem yapılır. BEA'da, mevcut görüntü çerçevesindeki her makro blok, referans görüntü çerçevesindeki arama alanında karşılık gelen makro bloklar ile kıyaslanmaktadır. Kıyas sonucunda en iyi eşleşme sağlayan makro blok aranarak bir hareket vektörü elde edilmesi amaçlanmaktadır.

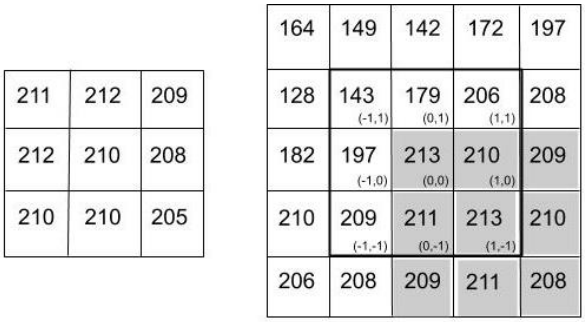


Şekil 2.2 Makro blokların arama alanı içerisinde karşılaştırılması

En iyi eşleşmenin sağlanabilmesi için, seçilen makro bloğun, mevcut görüntü çerçevesindeki tüm alanlarda aranması ve bunun tüm makro bloklar için sırasıyla yapılması öngörülmektedir. Fakat bu işlem, arama ihtimalleri göz önüne alındığında çok büyük maliyeti beraberinde getirmektedir. Bu nedenle seçilen makro bloklar, arama alanı dahilinde kıyaslanarak eşleşme yakalanmaya çalışılmaktadır. Art arda alınan görüntü çerçevelerindeki korelasyon da, eşleşme aranan bloğun, merkez blok ile komşuluğu ihtimalini güçlendirmektedir.

İyi bir makro blok eşleşmesi için, referans görüntü çerçevesindeki makro bloğun, mevcut görüntü çerçevesinde karşılık gelen alanda, bloğun dört tarafında p pikselle sınırlandırılmış bir arama alanı oluşturularak Şekil 2.2'de gösterildiği gibi

kıyaslanma işlemi yapılmaktadır. Burada p olarak kullanılan piksel sayısı, arama parametresi olarak adlandırılmaktadır. Arama işleminin daha iyi sonuç vermesi ve daha geniş alanda aranması istenirse, p parametresinin daha büyük almak gerekmektedir. Fakat bu işlem arama sonucunu iyileştirdiği gibi, işlem maliyetini de oldukça arttırmaktadır.



Şekil 2.3 En iyi makro blok eşleşmesinin aranması

Blok eşleşme algoritmasının bir örneği Şekil 2.3'te gösterilmiştir. Makro blok 3x3 piksellik bir kare olarak seçilmiş ve arama parametresi p=1 olarak seçilmiştir. Referans görüntü çerçevesinden alınan makro blok, mevcut görüntü çerçevesinde karşılık geldiği makro blok ve arama parametresi p=1 dahilinde 1 piksellik komşuluklarda kıyaslanmaktadır. Sol üst (-1,1) komşuluğundan başlayarak ve en son sağ alt (1,-1) komşuluğu olacak şekilde sırayla makro blokların OKH hesabı gibi yapılır ve en küçük değer, eşleşmenin olduğu konum olarak değerlendirilir.

Tablo 2.1 Arama alanı içerisindeki tüm makro bloklar için OKH değerleri

Makro Blok koordinatı	(-1,1)	(0,1)	(1,1)	(-1,0)	(0,0)	(1,0)	(-1,-1)	(0,-1)	(1,-1)
OKH	2674,4	1519,9	125,63	1782,41	669,5	30,42	848,31	121,98	3,65

Her komşuluktan hesaplanan OKH değeri Tablo 2.1'de gösterilmiştir. Değerler arasında en düşük olanı, referans makro blok ile en çok benzerlik olan makro blok konumunu göstermektedir. 9 konum için hesaplanan değerlerde, (1,-1) konumu en küçük OKH değerini verdiği için, hareket (0,0) konumunda (1,-1) konumuna doğru

olmuştur denir ve hareket vektörü denklem (2.1)'de belirtilen formül ile hesaplanabilir:

$$\left(A_x - B_x, A_y - B_y\right) = (\Delta x, \Delta y) \tag{2.1}$$

Bir makro bloğun diğeri ile eşleştirilmesi, yukarıdaki gibi bir OKH sonucuna göre yapılabileceği gibi, işlem daha sadeleştirilerek Ortalama Mutlak Fark (OMF) sonucuna göre de yapılabilir. Bu iki işlem için genelleştirilmiş denklemler, denklem (2.2)'de OMF, denklem (2.3)'te OKH olmak üzere verilmiştir:

$$\mathrm{OMF} = \frac{1}{\mathrm{N}^2} \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \sum_{\mathrm{j}=0}^{\mathrm{n}-1} |\mathrm{C_{ij}} - \mathrm{R_{ij}}| \tag{2.2}$$

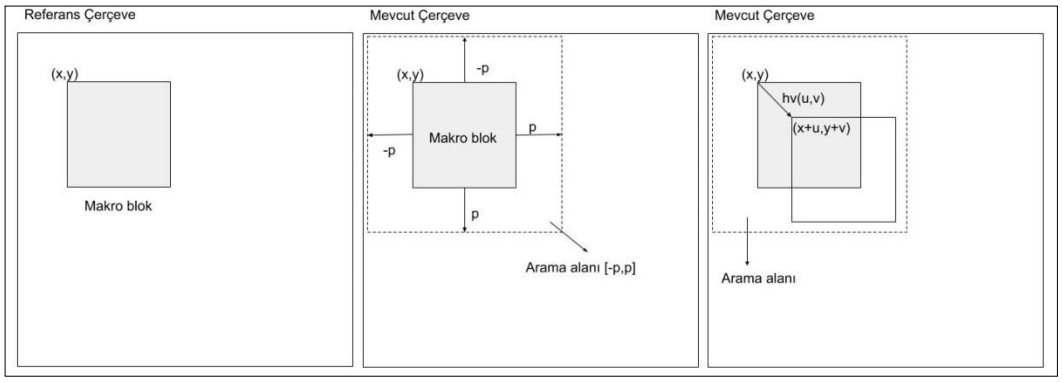
$$\mathrm{OKH} = \frac{1}{N^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \left(C_{ij} - R_{ij}\right)^2 \tag{2.3}$$

Buna göre; n x n piksellik makro blok için, *Cij* ve *Rij* sırasıyla mevcut blok ve referans bloktaki pikselleri göstermektedir. Bu şekilde bulunan sonuç karşılıklı piksellerin farklılıkları ile ilgili bilgi verse de, iki görüntü karşılaştırılması Peak-signal-to-noise-ratio (PSNR) değeri ile karakterize edilir. PSNR, bir sinyalin mümkün olan en fazla gücü ile gösterimin kalitesini etkileyen gürültü arasındaki orandır. Sinyaller geniş bir dinamik aralığa sahip olduğundan PSNR genellikle desibel olarak denklem (2.4)'teki formül ile hesaplanır:

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{(\text{orijinal verinin en büyük değeri})^2}{\mathrm{OKH}}\right] \tag{2.4}$$

2.2.1 Tam Arama

Etraflı arama olarak da bilinen TA algoritması, referans bloktan alınan makro bloktaki değerleri, mevcut blok içerisinde oluşturulan arama alanındaki tüm pikseller için olası eşleşme ihtimallerini değerlendirir ve her bir piksel için OKH hesaplayarak en uygun eşleşmeyi arar. Arama mantığı olarak düz fakat oldukça maliyetli bir algoritmadır. Arama alanı, referans görselden alınan makro bloğun mevcut görselde karşılık geldiği alanda, dört tarafından p arama parametresi kadar piksellerin eklenmesi ile oluşan alandır.



Şekil 2.4 Referans ve mevcut çerçeve içerisinde makro blok ile arama işlemi

Şekil 2.4'te referans ve mevcut bloklar için makro blok ve p arama parametresi ile genişletilen arama alanı gösterilmiştir. Mevcut görseldeki p arama parametresi ile belirlenen arama alanı, belli parametreler dahilinde uygun büyüklükte seçilmelidir. Bu parametrelere görüntünün ne kadar hareketli olduğu, görüntünün çözünürlüğü, arama işleminin yapıldığı işlemcinin gücü örnek olarak gösterilebilir. Görüntü çözünürlüğü yükseldikçe arama alanının da büyümesi, görüntüdeki hareket arttıkça arama alanının büyümesi gerekmektedir. Arama alanı büyüdükçe hesap yapılacak piksel sayısı da arttığı için hesap yapılacak işlemciye yüklenecek iş de artmaktadır.

Mevcut görsel çerçevesinde, ki referans görsel çerçevesinde de aynı uzunlukta olacaktır, makro blok karesinin bir kenarı N uzunlukta ve mevcut görsel çerçevesinde kullanılan arama parametresi p kabul edilirse $(2p + N)^2$ pikselden oluşan bir arama alanı söz konusu olacaktır. Tam arama için bakıldığında makro blok, mevcut bloktaki arama alanında $(2p + 1)^2$ adet eşleşme ihtimali olan olası blokla piksel piksel karşılaştırılacaktır. Karşılaştırma sonucunda en uygun blok seçilerek hareket vektörü elde edilecektir.

Karşılaştırma işleminde referans görüntüdeki ve mevcut görüntüdeki blokların her pikselleri için OKH hesabı yapılacak ve bu hesap sayısı N^2 olacaktır. $(2p + 1)^2$ adet eşleşme ihtimali için, $(2p + 1)^2\ x\ N^2$ adet işlem sayısı söz konusu olacaktır. (M x M) boyutunda bir görüntü çerçevesi düşünüldüğünde, $(M/N)^2$ adet makro blok olduğu (makro blok boyutunun, görüntü çerçevesinin tam katı olduğu varsayılırsa) görülebilir ve sonuçta $(M/N)^2\ x\ (2p + 1)^2\ x\ N^2$, ve sadeleştiğinde $M^2\ x\ (2p + 1)^2$ elde edilir. Bu sonuç, bir görüntü çerçevesi için tam arama ile yapılması gereken

işlem sayısıdır. Bir video görüntüsünün çerçeve-saniye oranı bilindiğinde, saniyede yapılacak olan işlem de hesaplanabilir.

TA algoritmasının sunduğu global optimum çözüme karşın hesaplamadaki maliyeti sebebiyle, sonuç doğruluğundan bir miktar ödün verecek ve fakat maliyetten oldukça kazanç sağlayacak hızlı arama algoritmaları geliştirilmiştir. Bundan sonraki arama algoritmaları başlıklarında hızlı arama algoritmalarından bahsedilecektir.

2.2.2 Üç Adım Arama

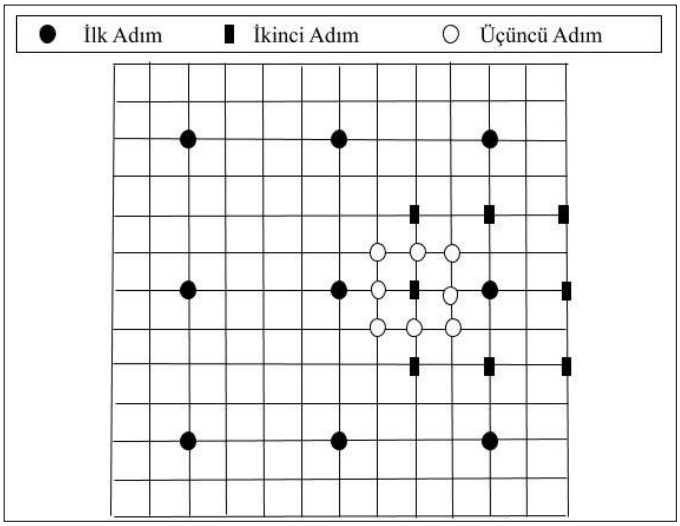
ÜAA algoritması, 1981 yılında önerilmiş güçlü bir arama algoritmasıdır [1]. Temelde, bir arama alanındaki eşleşme aranacak olası blok sayısını sınırlandırarak en doğru hareket vektörünü elde etmeyi amaçlar. ÜAA algoritması başlarda arama alanının nispeten küçük olduğu düşük çözünürlüklü video görüntüleri için önerilmiş olmasına ve tipik olarak p = 7 ile sınırlandırılmış olmasına rağmen, algoritmanın p > 7 olan video görüntüleri için uzatılabileceği görülmüştür [10]. Tabi bu durumda adım sayısı 3'ü geçecektir. Çünkü üç adım arama algoritmasında arama sayısı denklem (2.5)'teki formül ile elde edilebilir:

$$A_{ilk} = 2\ x\ \left(\frac{\log(p+1)}{\log 2}\right) - 1 \qquad (2.5)$$

Ayrıca, n. adımda pikseller arasındaki mesafe, yani adım mesafesi denklem (2.6)'daki formül ile bulunabilir:

$$ss(n) = \ 2^{\mathrm{A_{ilk}-n}} \qquad (2.6)$$

ÜAA algoritmasının her adım için eşit olarak dağıtılmış bir arama modeli kullandığı ve dolayısıyla sadelik ve düzenlilik gösterdiği söylenebilir. Özellikle kontrol noktalarının sayısı, ilk adım için dokuz, sonraki adımlar için sekiz (önceki adımda zaten kontrol edilen konum hariç tutulmaktadır) olarak uygulanmaktadır. Arama parametresi p = 7 için, üç adım arama algoritmasının kontrol noktası, yani eşleşme aramak için karşılaştırma yapılan olası arama noktaları 9+8+8 sonucu olarak 25 şeklinde bulunur.



Şekil 2.5 ÜAA algoritması adımları

Üç adım arama algoritması, Şekil 2.5 baz alınırsa, sırasıyla şu şekilde çalışır:

İlk olarak arama parametresi p = 6 seçilir. Bu parametre, ÜAA algoritmasının başlayacağı merkez pikselin dört tarafından kenarlara olan uzaklığın piksel sayısıdır.

Denklem (2.6)'daki formül kullanılarak adım mesafesi hesaplanır. Örnek için adım mesafesi 4 bulunmuştur. Merkez pikselin 4 tarafından olacak şekilde adım mesafesi uzaklığında bir kare çizilir. İlk adım için en düşük OKH değerinin hesaplanacağı 9 nokta oluşturulmuştur.

En düşük OKH, 9 nokta arasından tespit edildikten sonra, tespit edilen nokta ikinci adımın merkez pikseli olarak seçilir. Adım mesafesi 2'ye bölünerek daha küçük bir alanda arama yapılacaktır.

Merkez pikselin OKH değeri bir önceki adımda hesaplandığı için, ikinci adımda sadece 8 noktanın OKH değerinin hesaplanması yeterlidir. 9 noktadan en düşük OKH değerine sahip olan nokta tespit edilir.

Tespit edilen nokta, üçüncü adımın merkez piksel noktasıdır. Adım mesafesi yarıya düşürülerek arama devam eder. Arama algoritması, arama parametresi 1'den küçük olana kadar devam eder. Bu örnek için adım mesafesi 1 olduğundan, arama algoritması bu adımda son bulur. Üçüncü adımda tespit edilen en düşük OKH

değerine sahip piksel, arama alanı dahilinde en uygun eşleşmenin olduğu piksel olarak seçilir.

ÜAA algoritması ile ortaya çıkan sorunlardan biri, ilk adımda küçük hareket kestirimi için verimsiz hale gelen, düzenli olarak tahsis edilmiş bir kontrol noktası deseni kullanmasıdır [11].

Ayrıca ÜAA algoritması, arama alanı içerisinde seyrek karşılaştırma noktaları seçerek ilerlediği için, hatalı bir karşılaştırma seçimi, arama algoritmasını istenmeyen yöne doğru kaydırarak doğru sonuçtan uzaklaşmaya sebep olabilecektir.

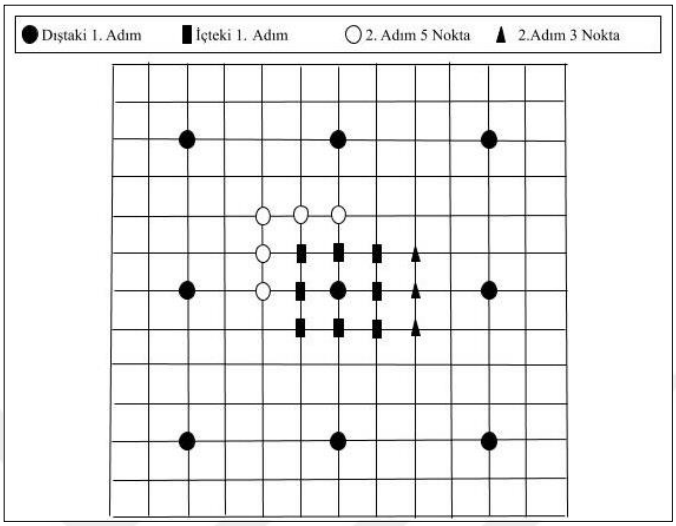
2.2.3 Yeni Üç Adım Arama

ÜAA algoritması, ilk adımda küçük hareketlerin kestirimi için yetersiz kalan, tek biçimli olarak tahsis edilmiş bir kontrol noktası modeli kullanmaktadır. YÜAA algoritması ise, araştırmanın hareket vektörü dağılımına uyarlanabilir hale getirilmesiyle elde edilen birinci basamakta merkez taraflı bir kontrol noktası deseni ve hesaplama maliyetini düşürmek için yarı durdurma tekniği kullanmaktadır [2]. Bununla birlikte ÜAA algoritması, ilk adımında durağan veya yarı-durağan bloklarda görünen küçük hareketleri yakalamak için çok verimli olmayan tek biçimli bir şekilde ayrılmış bir arama modeli kullanır. Bu sorunu çözmek amacıyla, aramayı hareket ölçeğine ve belirsizliğe daha uyumlu hale getirmek için YÜAA algoritması önerilmiştir. YÜAA algoritmasında arama penceresi büyüklüğü hareket ölçeğinin belirsizliğine göre ayarlanır. Belirsizlik, kontrol edilen noktalar arasındaki blok bozulma önleminin farkı ile tahmin edilir. Daha küçük bir fark, daha büyük bir belirsizlik olduğunu gösterir ve bu nedenle bir sonraki adımda arama kapsamı arttırılır. Aksi takdirde, normal üç adım arama algoritması yalnızca 8 komşu araması gerçekleştirilir [2].

Gerçek dünyadaki görüntü dizilerinin blok hareketi genellikle düzgün ve hafif geçişli geçişler barındırmaktadır. Bunun sonucu olarak da global minimum noktaları merkezde toplanmak üzerine eğilim gösterirler.

YÜAA algoritması, üç adım arama algoritması ile kıyaslandığında, gerçek dünyadaki görüntü dizilerinin global minimum noktalarının merkezde

toplanmasına bağlı olarak merkez taraflı bir karşılaştırma yöntemine sahiptir. Durağan ve yarı-durağan blokları tespit ettiği zaman algoritma sonlandırılır ve böylece gereksiz adımları ilerletmeden işlem yükünü azaltma yönünde geliştirilmiştir.



Şekil 2.6 YÜAA algoritması adımları

Yeni üç adım arama algoritması, sırasıyla şu şekilde çalışır:

Üç adım arama algoritmasında olduğu gibi arama parametresi seçilir ve arama adımı hesaplanır. Bu doğrultuda merkez nokta ve arama adımı kadar çevresinde 8 nokta belirlenir. Bunlara ek olarak, merkez nokta etrafında 8 nokta daha Şekil 2.6'daki gibi belirlenir. Bu noktalar, durağan ve yarı-durağan blok tespiti için kullanılacaktır.

Noktalar hesaplandığında, ilk adımda 17 adet karşılaştırma noktası olduğu görülebilir. İlk adım için 17 nokta, OKH hesabı yapılarak karşılaştırılır. Bu adımda en küçük OKH değeri merkez pikselde tespit edilirse arama algoritması sonlandırılır.

İlk adımdaki aramada en küçük OKH değeri, merkez pikselin etrafındaki komşu 8 pikselden birinde bulunursa, arama devam eder. Bulunan piksel, komşu 8 pikselin köşesindeki piksel ise, Şekil 2.6'daki gibi 5 piksel daha, kenarındaki piksel ise 3 piksel daha eklenerek arama devam eder. Bu adımdaki arama toplam arama sayısına, köşe veya kenar piksel olma durumuna göre sırasıyla, 5 veya 3 arama eklenerek yapılır. Çünkü bir önceki aramada, bu adıma dahil olan karedeki bazı

piksellerin OKH değerinin zaten yapıldığı görülmektedir. Bulunan ne küçük OKH değeri ile arama sonlandırılır.

İlk adımdaki aramada en küçük OKH değeri, arama adımı çevresindeki 8 pikselden birinde bulunursa, üç adım arama algoritmasından devam edilerek uygun piksel bulunur.

YÜAA algoritmasında, arama işlemi en az 17 pikselin karşılaştırılmasıyla bulunabilir. En kötü durumda ise 33 karşılaştırma yapılmış olacaktır.

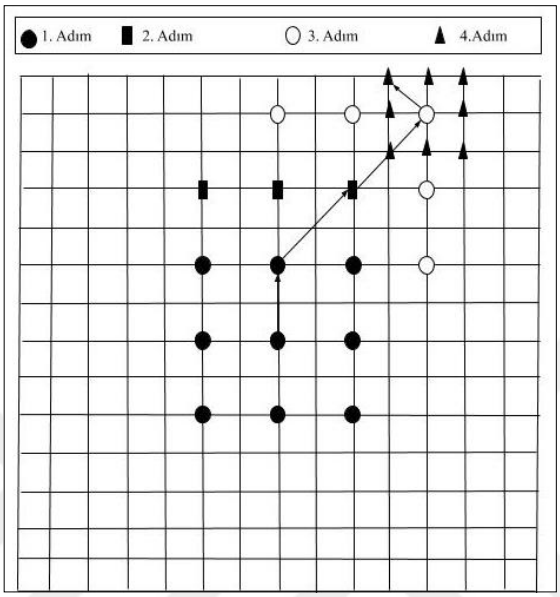
Bu yönüyle genel olarak görüntü dizilerinin merkez taraflı olduğu düşünülürse YÜAA algoritması, ÜAA algoritmasına kıyasla toplamda daha az arama işlemi ile görüntü çerçevesini tamamlayabilir. Ayrıca, yarı durağan görüntü dizilerinde ÜAA algoritmasının ilk adımdaki geniş arama alanını geliştirerek bu görüntüler için de doğruluk oranını arttırmaktadır.

2.2.4 Dört Adım Arama

DAA algoritması, 1996 yılında L.M Po ve W.C Ma tarafından önerilmiştir [3]. Algoritma, ÜAA algoritmasına kıyasla daha küçük bir başlangıç adım boyutu kullanarak, gerçek dünyadaki görüntü dizilerinin merkeze dayalı özelliklerini kullanır. DAA algoritması, daha küçük başlangıç adımı boyutundan dolayı bir arama penceresinin sınırına ulaşmak için dört arama adımına ihtiyaç duyar. YÜAA algoritmasındaki küçük hareket durumuyla aynı olarak, DAA algoritması da ikinci ve üçüncü adım aramasında bir yarım durdurma tekniği kullanır.

DAA algoritması, maksimum 7x7 piksellik yer değiştirmeler için önerilmektedir. ÜAA algoritmasındaki 9x9 piksellik ilk adım arama penceresi yerine, ilk adımda 5x5 piksellik pencerede 9 adet aranacak nokta belirler. Merkez taraflı bu arama bir sonraki adımda, 9 nokta içinden bulunan en küçük OKH değerli pikseli merkez alacak şekilde kaydırılarak devam eder. İlk adımdan sonraki adımlarda arama penceresinin boyu, en küçük OKH değerli pikselin konumuna bağlı olarak değişir. En küçük OKH değerli piksel arama penceresinin ortasındaki piksel ise, arama 3x3 piksellik bir arama pencere boyu ile devam eder ve son adıma giderek algoritma sonlandırılır. Aksi durumda arama penceresinin boyu, ikinci adım için 5x5 olacaktır. En küçük OKH değerli pikselin merkezde bulunma koşulu, ikinci adım

için de geçerlidir. İkinci adımda da merkezde bulunursa, üçüncü adım 3x3 piksellik bir arama penceresiyle devam ederek sonlandırılacak, aksi durumda 5x5 piksellik bir arama penceresiyle devam edecektir. Son adımda, arama penceresi 3x3 piksele düşürülür ve arama bu adımda en küçük OKH değerli pikseli bulur ve arama sonlandırılır.



Şekil 2.7 DAA algoritması adımları

DAA algoritması, Şekil 2.7'de görüldüğü gibi aşağıdaki gibi çalışır [12]:

İlk adımda, 15x15 piksellik arama alanının merkezinde bulunan 5x5 piksel boyutundaki arama penceresinde 9 kontrol noktası aranarak, Şekil 2.7'deki gibi en küçük OKH değerli piksel bulunur. Eğer en küçük OKH değerli piksel, arama alnının merkezindeyse, algoritma dördüncü adımdan devam eder, aksi durumda sonraki adımdan arama işlemi devam eder.

İkinci adımda, arama penceresinin boyu 5x5 piksel boyutundan devam eder. Birinci adımda bulunan en küçük OKH değerli pikselin konumu bu adımda önemlidir:

- İlk adımda bulunan en küçük OKH değerli piksel arama alnının köşesinde ise, Şekil 2.7'de olduğu gibi 5 ek kontrol noktası eklenir.

- İlk adımda bulunan en küçük OKH değerli piksel arama alanının yatay veya dikey ekseninin ortasında bulunursa, Şekil 2.7'de olduğu gibi 3 ek kontrol noktası eklenerek arama devam eder.

Bulunan en küçük OKH değerli piksel, arama alanının merkezinde ise, algoritma dördüncü adımdan devam eder, aksi durumda bir sonraki adıma geçer.

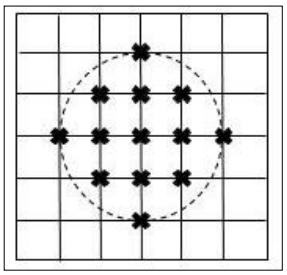
Üçüncü adımda, ikinci adımda yapılanların aynısı tekrar edilir. Fakat bu adımdaki sonuca bağlı olmaksızın algoritma dördüncü adımdan devam eder.

Dördüncü adımda, arama penceresi Şekil 2.7'deki gibi 3x3 piksellik bir alana düşürülür. Bu arama alanında bulunan en küçük OKH değerli piksel, hareket vektörünün yönünü belirler.

DAA algoritması, en iyi durumda 17, en kötü durumda ise 27 arama işlem sayısı ile sonuca ulaşmaktadır. En kötü durum, hareket karmaşıklığının en yoğun olduğu zamanlarda ortaya çıkacaktır. Gerçek dünyadaki görüntü dizilerinin merkeze dayalı küresel minimum hareket vektörü dağılım karakteristiğine dayanarak, bu algoritmanın hızlı sonuç vereceği görülmektedir. Bu şekilde bakıldığında, DAA algoritmasının ÜAA algoritmasından daha iyi performans gösterdiği ve daha küçük hesaplama gereksinimi olan OKH ölçüsü açısından YÜAA algoritmasına benzer performanssa sahip olduğu söylenmektedir. Ayrıca, ÜAA ve YÜAA algoritmaları ile karşılaştırıldığında, kamera yakınlaştırma ve hızlı hareket gibi karmaşık hareket içeren görüntü dizilerinde dört adım arama algoritmasının performansı daha iyi olduğu görülmektedir [3].

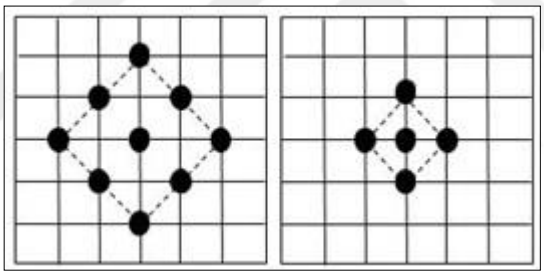
2.2.5 Eşkenar Arama

EA algoritması, elmas arama algoritması olarak da bilinir ve 2000 yılında Shan Zhu ve Kai-Kuang Ma tarafındadn sunulmuştur. Algoritma, ÜAA algoritmasından daha iyi sonuç elde etmektedir. YÜAA algoritmasına yakın sonuçlar elde etse de, %22 daha az hesaplamaya ihtiyaç duyarak maliyeti azaltma yönünde başarısı söz konusudur [4].



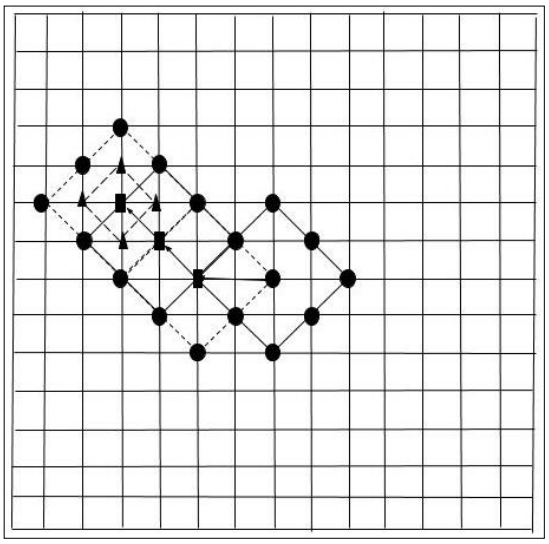
Şekil 2.8 EA algoritmasında aranan tüm noktalar

Yapılan çalışmalarda, hareket vektörlerinin yaklaşık %52,76 ile %98,70'inin dairesel bir destek içine 2 kabuklu bir radyumla sarılı olduğu ve sıfır hareket pozisyonunda merkezlendiği görülmüştür. Ayrıca, gerçek dünyadaki video dizilerinin blok yer değiştirmesinin herhangi bir yönde olabildiği, fakat esas olarak yatay ve dikey yönlerde olduğu (örneğin, kamera kaydırma) görülmüştür [13]. Bu iki nedene bakıldığında, Şekil 2.8'deki gibi daire içinde seçilen arama noktaları, hareket vektörünün bulunması için arama alanı olarak en uygun noktalar olarak görülmektedir.



Şekil 2.9 EA algoritması arama paternleri: Büyük ve küçük arama paternleri

EA algoritmasında kullanılan iki ayrı patern, Şekil 2.9'da gösterilmiştir. İlk patern, büyük eşkenar arama paterni olarak adlandırılmaktadır ve 9 kontrol noktasına sahiptir. ikinci patern ise küçük eşkenar patern olarak adlandırılmaktadır ve 5 kontrol noktasına sahiptir.



Şekil 2.10 EA algoritması arama adımları

EA algoritması, Şekil 2.10'da görüldüğü gibi şu şekilde çalışır:

İlk adımda, büyük eşkenar arama paterni, arama penceresinin merkezinde konumlandırılır. 9 kontrol noktası hesaplanarak en küçük OKH değerli piksel bulunur. Bulunan piksel, arama paterninin orta noktasıysa, üçüncü adımdan devam edilir, aksi durumda bir sonraki adıma geçilir.

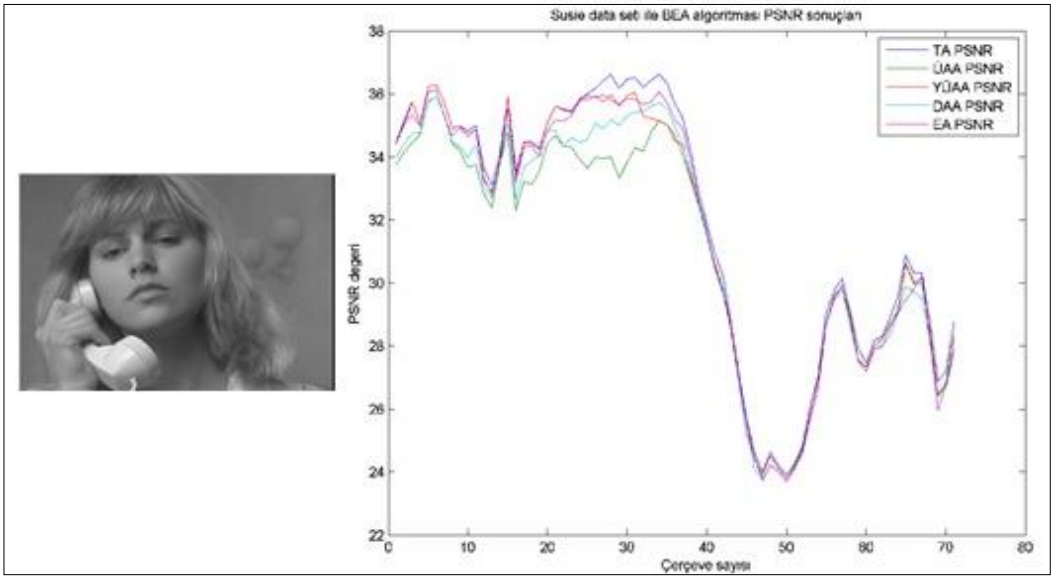
İkinci adım olarak, birinci adımda bulunan en küçük OKH değerli piksel, yeni oluşturulan büyük eşkenar arama paterninin merkez noktası olarak seçilir. Birinci adımda bulunan nokta, eşkenarın köşe noktası ise 5 kontrol noktası, kenar noktası ise 3 kontrol noktası eklenerek devam arama yapılır. Arama sonucu bulunan en küçük OKH değerli piksel merkez konumda ise, üçüncü adıma geçilir, akis durumda merkez konumda bulunana kadar arama bu adımda devam eder.

Üçüncü adımda, arama paterni büyük eşkenar yerine küçük eşkenar arama paterni olarak ayarlanır. Bu adımda 5 kontrol noktası arasında bulunan en küçük OKH değerli piksel, hareket vektörünü oluşturur.

DAA algoritması, ÜAA algoritmasına kıyasla daha iyidir ve YÜAA algoritmasına kıyasla yakın bir OKH performansı yakalamıştır, %22 daha az işlem yaparak daha az maliyet sonucu ortaya koymaktadır. Ayrıca, DAA algoritması ile karşılaştırıldığında, OKH performansı, yeniden oluşturulan görüntü kalitesi ve ortalama arama noktası sayısı açısından daha iyi çalışmaktadır [4].

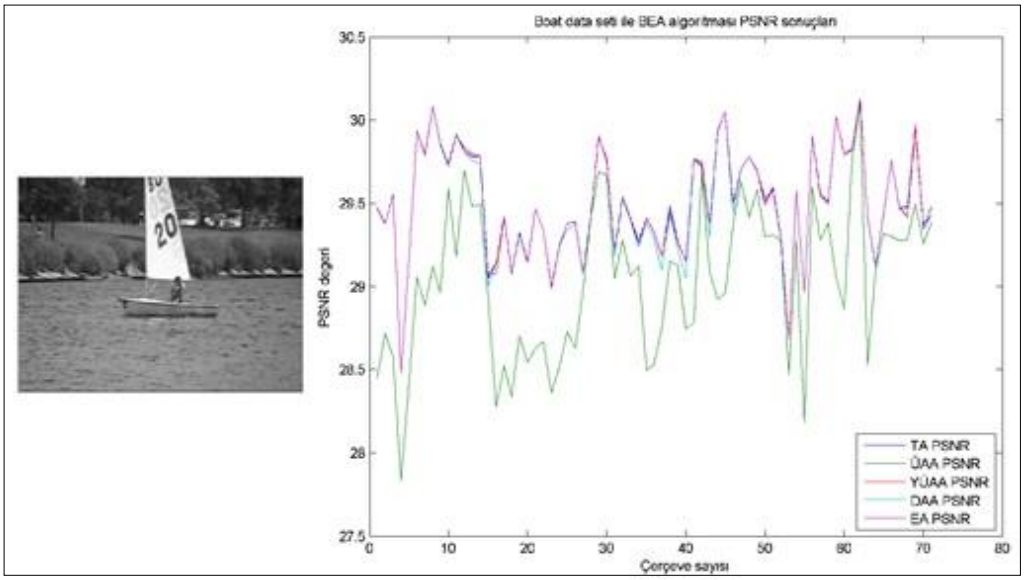
2.3 Arama Algoritmalarının Karşılaştırılması

Hareket kestirimin yöntemi olarak sıkça kullanılan BEA yönteminde, algoritmanın hızlanması adına, sonucun TA algoritmasına yakın fakat çok daha az maliyetli olması için çeşitli arama algoritmaları önerilmiştir. Önerilen ÜAA, YÜAA, DAA ve EA arama algoritmaları, belli veri setleri ile denenmiş ve algoritmanın uygulandığı çerçeveler için PSNR değerleri grafiklerle gösterilmiştir.



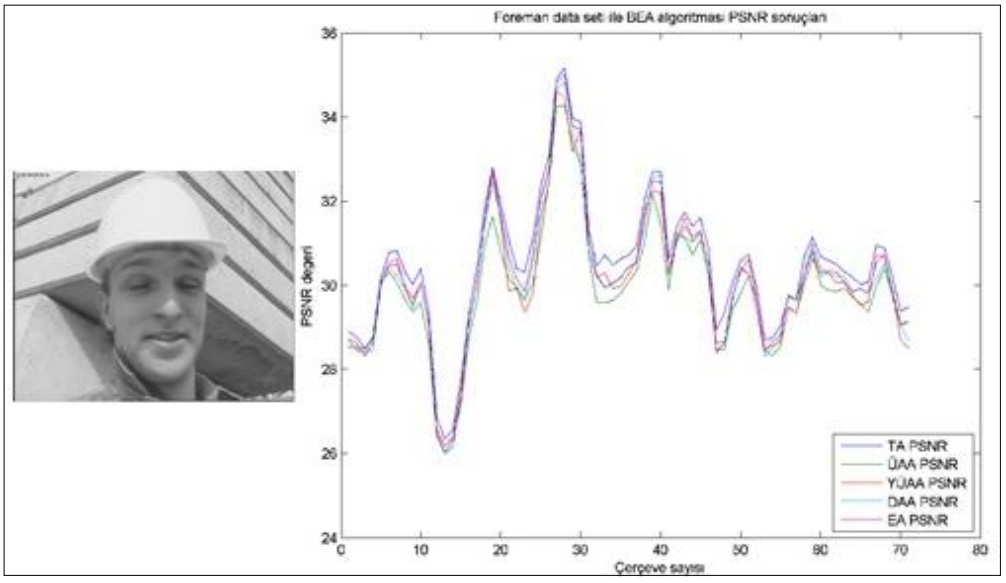
Şekil 2.11 "susie" veri seti için PSNR değerleri

Şekil 2.11'de "susie" data setine ait, 70 çerçeve için elde edilmiş PSNR değerleri görülmektedir. Grafiğe bakıldığında TA üstünlük sağlamıştır, başarım oranı olarak TA algoritmasını YÜAA ve EA sırasıyla takip etmektedir.



Şekil 2.12 "boat" veri seti için PSNR değeleri

Şekil 2.12'de "boat" data setine ait, 70 çerçeve için elde edilmiş PSNR değerleri görülmektedir. Grafiğe bakıldığında, EA ve YÜAA algoritmalarının TA algoritmasına yakın sonuçlar elde etiiği görülmektedir ve başarım oranı olarak TA algoritmasını YÜAA ve EA sırasıyla takip etmektedir.



Şekil 2.13 "foreman" veri seti için PSNR değleri

Şekil 2.13'te "foreman" data setine ait, 70 çerçeve için elde edilmiş PSNR değerleri görülmektedir. Grafiğe bakıldığında TA algoritmasının daha başarılı olduğu

görülmektedir, başarım oranı olarak TA algoritmasını YÜAA ve EA sırasıyla takip etmektedir.

Kullanılan veri setleri, hareket kestirim algoritmalarında temel alınan ve sonuçları karşılaştırılabilen veri setleridir. “susie” veri seti, arka plan ve ön plan hareketlerinin az olduğu bir veri setidir. “foreman” veri setinde kullanılan ilk 70 çerçeve, arka planın nispeten durağan olduğu, ön planın ise daha hareketli olduğu bir veri setidir. “boat” veri setinde kullanılan çerçeveler ise, arka plan ve ön planın hareketli olduğu bir veri setidir.

Tablo 2.2 Arama algoritmalarının veri setlerine göre PSNR değerleri

	TA	**ÜAA**	**YÜAA**	**DAA**	**EA**
Foreman	33,5504	32,8789	33,3228	33,0641	33,2855
Susie	34,7849	34,1323	34,6618	34,2599	34,4873
Boat	32,4006	32,2318	32,3999	32,2343	32,4000

Tablo 2.2’de, her veri setinden alınan 70’er çerçeveden elde edilen PSNR değerlerinin arama algoritması başına hesaplanan ortalama PSNR değerlerini göstermektedir. Sonuçlara göre, TAA’nın en iyi sonucu verdiği, YÜAA ve EA’nın birbirine çok yakın sonuçlar verdiği ve ÜAA ve DAA’nın ise geride kaldığı görülmektedir.

3

Önerilen Yöntemin Uygulanması

3.1 Tezde Önerilen Yöntem

Hareket kestiriminde, temelde görüntü dizisindeki art arda iki görüntü çerçevesi alınarak pikselleri karşılaştırılır. Karşılaştırma, çerçeveler belirli büyüklükteki bloklara bölünerek blok bazlı olarak yapılır. Bloklardaki piksel farklılıkları sonucunda referans bloğun, aranan en düşük OKH değerlikli alana doğru hareket ettiği varsayımı yapılır. Görüntü dizilerinde sahne değişimi olmadığı taktirde ise bu varsayım büyük oranda doğru olur.

Karşılaştırılan bloklardaki piksellerde, en başta az bir işlemle farklılık olup olmadığı anlaşılabilir. Farklılık yoksa o blokta arama işleminin keskin ve maliyetli bir arama yöntemi ile yapılmaması sağlanarak işlem maliyeti ve zamandan kazanım sağlanabilir. Sadece farklılık olan bloklar anlaşıldığında ise o bloklarda daha keskin arama yöntemleri kullanılarak doğru hareket vektörünün çıkarımı sağlanabilir. Bu sayede tek bir arama algoritması kullanımı ile karşılaştırıldığında, maliyetten veya PSNR başarımından bir kazanımın sağlanabilir.

Başarım oranını arttırmak, işlem maliyetini azaltmak amaçlarıyla, ön bir işlemle hareketli, yarı hareketli ve durağan blokların saptanması ve hareket durumuna göre uygulanacak olan arama algoritmasının seçilmesi önerilmektedir.

Önerilen yöntemde, görüntü dizisinden alınan art arda iki çerçeve, birbiri ile piksel bazlı olarak karşılaştırılarak bir fark çerçevesi elde edilir. Elde edilen fark çerçevesinde, belirli boyuttaki bloklar gezdirilerek her blok için OKH hesabı yapılır. OKH değeri belli eşikten küçük olan bloklar durağan blok olarak sınıflandırılır. Fark çerçevesi, adaptif bir eşik değeri ile filtrelenerek hareket değişim matrisi elde edilir. Hareket değişim matrisi üzerinde, belirlenen büyüklükteki bloklar gezdirilerek, blok içindeki toplam değer bulunur, belirlenen

eşik değeri ile kıyaslanarak bloğun hareketli mi ya da yarı hareketli mi olduğuna karar verilir.

Hareketli olarak sınıflandırılan blok, başarım oranı yüksek, maliyet olarak nispeten yüksek olan bir arama algoritması kullanılarak aranır ve en yüksek başarım oranı ile hareket vektörünün elde edilmesi sağlanır. Yarı hareketli olarak sınıflandırılan blok, maliyet olarak en düşük ve başarım oranı nispeten daha düşük olan bir arama algoritması ile aranır. Durağan olarak sınıflandırılan bloklarda ise arama yapılmaz. Böylece çerçevenin toplamına bakıldığında, tek bir arama algoritmasının kullanılmasına kıyasla işlem yükü ve arama maliyeti azalmış, başarım oranı yükselmiş olacaktır.

3.2 Görüntü Değişiminin Tespiti

Görüntü dizisinden art arda seçilen iki görüntü çerçevesi arasındaki görüntü değişiminin belirlenmesi, hareketli blokların sınıflandırılması ve uygun arama algoritması kullanılarak hareket kestirimi yapılması için gereken ilk adımdır. Görüntü değişimi belirlenerek çerçevedeki hareketli olabilecek alanların anlaşılması sağlanabilir.

Elde edilen iki görüntü çerçevesinden, zaman odaklı ele alındığında (t-1) zamana ait olan çerçeve referans çerçeve olarak adlandırılmakta, (t) zamana ait olan ise mevcut çerçeve olarak ele alınmaktadır.

Referans çerçeve ve mutlak çerçeve karşılaştırıldığında, hareketsiz kısımlarındaki piksel değerlerinin aynı veya çok yakın olduğu, hareketli alanlarındaki piksel değerlerinin ise birbirlerinden farklı oldukları görülebilir. Referans çerçeve ve mevcut çerçevenin karşılıklı piksel değerlerinin birbirlerinden farkının mutlak değeri, fark çerçevesini verir. Görüntü çerçevelerinde, arka plan, ön plan veya her ikisinin de değişimini, gürültü bozulmalarından ayırabilmek için (i,j) konumundaki karşılıklı piksellerin kıyaslanması yerine, (i, j) konumunda merkezlenmiş belirli bir boyuttaki karşılıklı bloğun OMF değeri, (i,j) pikselinin fark değeri olarak belirlenebilir [17].



Şekil 3.1 (t) ve (t+1) zamanlarına ait çerçeveler ve fark çerçevesi

Referans çerçeve ve mevcut çerçevenin farkı Şekil 3.1'deki gibi alındığında fark çerçevesi elde edilir. Çerçevede, iki çerçeve arasındaki değişen alanlar görülebilir ve bu alanların hareket eden noktalardan oldukları söylenebilir.

3.3 Durağan ve Hareketli Blokların Tespiti

Hareketli noktalardan oluşan alanların belirlendiği fark çerçevesi üzerinde işlem yapılarak durağan ve hareketli blokların tespiti yapılabilir. İstatistiklere göre referans ve mevcut çerçeve arasında, 16x16 boyundaki bloklar ile karşılaştırma yapıldığında, durağan bloklardaki MFT %98 oranında 512'den küçük çıkmaktadır [7].



Şekil 3.2 Fark çerçevesi üzerinden durağan blokların elde edilmesi

Piksel değerlerinin mutlak farkı alınarak oluşturulan fark çerçevesi üzerinde, 16x16 piksel boyundaki bloklar ile pikseller üzerindeki değerler toplanır. Toplam değer 512'den küçük çıkarsa blok, durağan blok olarak sınıflandırılır. 512'den büyük olduğu durumda ise hareketli blok olarak sınıflandırılır. Şekil 3.2'deki gibi fark çerçevesi üzerinde işlem yapıldığında, siyah renkli durağan bloklar ve beyaz renkli hareketli bloklar görülebilir.

3.4 Hareketli Blokların Sınıflandırılması

Fark çerçevesinde yapılan işlemle durağan bloklar ayrıldıktan sonra kalan blokların, hareketli veya yarı hareketli olarak sınıflandırılması gerekmektedir. Sınıflandırma işlemi için ilk olarak fark çerçevesine bir eşik değeri uygulanır ve görüntü değişim ikili matrisi elde edilir. Matris üzerinde bloklar gezdirilerek 1 değerleri toplanır ve her blok için toplanan 1 değerleri, bir eşik değeri ile karşılaştırılarak bloğun hareketli mi yarı hareketli mi olduğuna karar verilir.

3.5 Görüntü Değişim Matrisi için Adaptif Eşik Değeri Tespiti

Fark çerçevesinin oluşturulduğu Şekil 3.2'ye bakıldığında, görüntüdeki siyaha yakın gri tonlarının hareketsiz alanları ve arka planı gösterdiği, beyaza yakın gri alanların ise hareketli alanı, cismi gösterdiği anlaşılmaktadır. Bu iki alanın birbirinden ayrılması için bir işleme ihtiyaç vardır. Fark çerçevesi ile piksel değişiminin hesaplanmasının ardından belirli bir eşik değerinden geçirilerek ikili matrisin elde edilmesi gerekmektedir. Eşik değeri, sabit bir değer olarak da verilebilir. Fakat sabit değeri bütün çerçevelerde kullanarak yapılan işlemde, yanıltıcı sonuçlar meydana gelebilir. Görüntü çerçevelerinin ışık oranı, hareket oranı, dolayısıyla histogram oranı farklılık göstereceğinden, sabit verilen eşik değeri, her görüntü çerçevesi için istenen geçerli eşik değerini yansıtmayacaktır.

Eşik değerinin her görüntü çerçevesi için adaptif olarak değişebilir olması, en doğru sonucun elde edilmesi için önemlidir. Bu nedenle her çerçeve için ayrı ayrı eşik değeri hesaplanmasına ihtiyaç vardır. Adaptif şekilde eşik değerinin hesaplanabilmesi için Otsu eşik belirleme yöntemi tercih edilmiştir.

Hareketli alanları hareketsiz alanlardan ayırt edebilmek için görüntünün arka plan ve ön plan olarak iki alandan oluştuğu varsayılır. Arka plan ve ön plan renk değerleri temel alındığında, iki sınıfın da kendi içinde türdeş, yani her noktasında benzer değerde olması ve diğer sınıftan da farklı olması beklenir. Bu noktada ölçü olarak varyans değeri kullanılır.

Varyans, bir dizi sayının, bu sayıların aritmetik ortalaması civarında nasıl dağıldığının yorumlanmasını sağlayan bir ölçüdür. Denklem (3.1)'de verilen formülde, varyans değerinin nasıl hesaplanacağı gösterilmiştir.

$$\sigma^2 = \sum_{i=0}^{N}(x_i - x'_i)^2 \Pr(x_i) \quad (3.1)$$

Bu ifadeye sadık kalarak, varyans değeri ne kadar büyükse dağınıklığın o kadar büyük olduğu, varyans değeri ne kadar küçükse de sayıların bir değer etrafında yoğunlaştığı çıkarımı yapılabilir. Sınıf içi varyans değerinin en küçük olması veya sınıflar arası varyansın en büyük olmasını sağlayan değer, eşik değeri olarak seçilir [14].



Şekil 3.3 Ortalama piksel değeri ve Otsu eşik belirleme yönteminin sonuçları

Şekil 3.3'te, ilk olarak görüntünün piksel değerlerinin ortalaması eşik değeri olarak kullanılmıştır. Ardından, Otsu eşik belirleme algoritması kullanılarak bulunan eşik değerine göre filtreleme işlemi yapılmıştır.

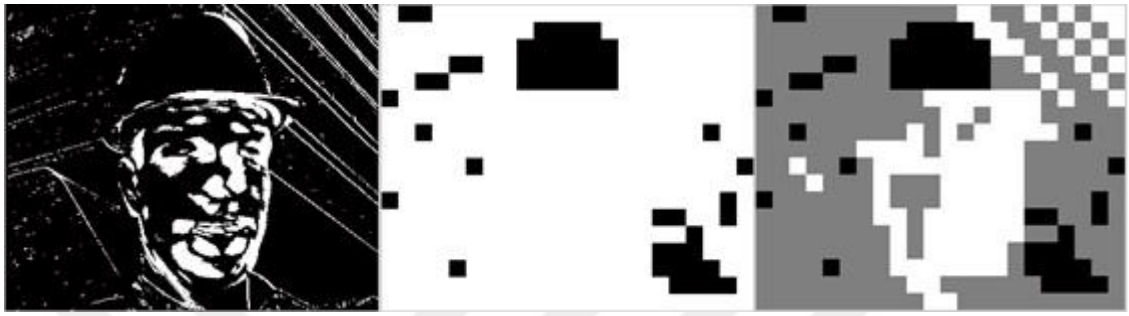
3.6 Hareketli ve Yarı Hareketli Blokların Belirlenmesi

Referans çerçeve ve mevcut çerçeve kıyaslanarak elde edilen fark çerçevesi, ikili matrise dönüştürüldükten sonra, çerçevenin belirlenen boyuttaki bloklara ayrıştırılarak blokların hareketli ve yarı hareketli olarak sınıflandırılması işlemi yapılır.

Bloklar, BEA algoritmasında kullanılan 16x16 makro blok boyutlarında oluşturulurlar. Durağan olarak sınıflandırılan bloklar bu adımda es geçilir. Diğer bloklar için, blokta bulunan 1 değerleri toplanır. Toplam değer, belli bir eşik değerinden büyük ise blok, hareketli blok olarak sınıflandırılır. Toplam değer eşik değerinden küçük ise blok, yarı hareketli blok olarak sınıflandırılır.

Kullanılan eşik değeri, her blok için sabit olarak ele alınabilir. Çünkü fark çerçevesi ikili matrise dönüştürüldükten sonra elde edilen 1 ve 0 değerlerinden oluşan

matriste, sadece referans ve mevcut çerçevenin birbirinden farklı olan pikselleri 1 değeri, geri kalan pikseller 0 değerini almıştır. Artık görüntünün ışık oranı, hareket oranı gibi parametreler geride bırakılmış ve parametrelerden bağımsız bir ikili matriste işlem yapılmaya başlanmıştır. Bu nedenle, tüm çerçeveler için uygun görülen sabit bir eşik değeri, hesaplanan tüm bloklar için hareketli veya yarı hareketli olarak sınıflandırılabilecek şekilde kullanılabilir.



Şekil 3.4 Durağan, yarı durağan ve hareketli blokların elde edilme adımları

Blok boyutu 16x16 olarak ele alındığında bloktaki 256 adet pikselin 1 değerleri toplanır. Toplam değer, eşik değeri ile kıyaslanır. Eşik değeri %50 olarak seçildiğinde, bloğun hareketli olarak değerlendirilmesi için 128 adet pikselin 1 değerine sahip olması gerekmektedir. Aynı şekilde %25 eşik değeri seçildiğinde ise 64 adet pikselin 1 değerine sahip olması beklenmektedir. Bloktaki değerlerin toplamı eşik değerinin altında kalırsa blok, yarı hareketli blok olarak sınıflandırılır ve gri renkle renklendirilir. Şekil 3.4'te ilk olarak, ikili matris olarak filtrelenmiş görselde, eşik değeri 50 olarak kullanılmıştır. Ardıdnan durağan olarak sınıflandırılan bloklar siyahla gösterilmiş ve son olarak sadece beyaz bloklar üzerinde yapılan işlem sonucu yarı hareketli gri bloklar ve hareketli beyaz bloklar gösterilmiştir.

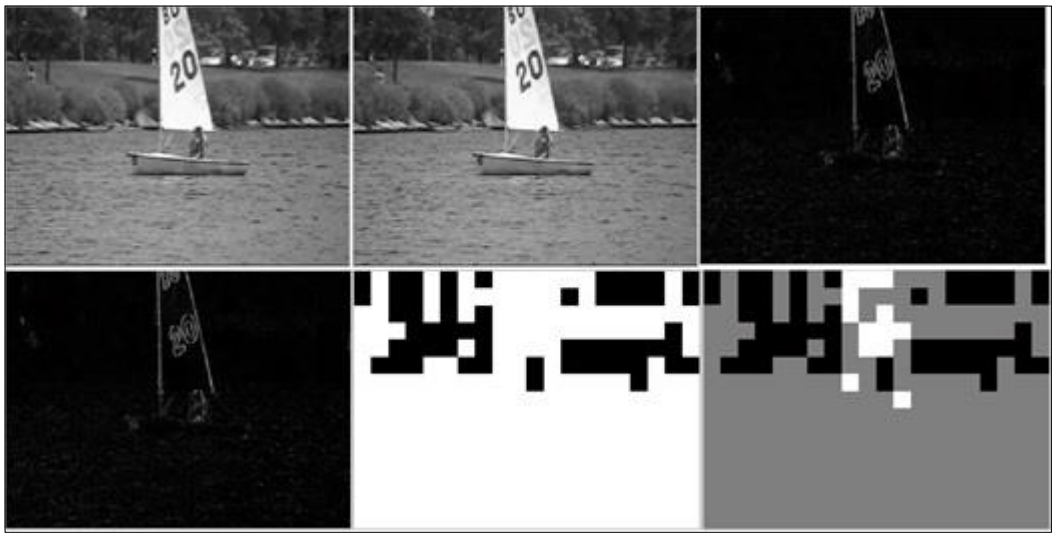
Eşik değerinin büyümesi ile bazı hareketli blokların hareketsiz olarak sınıflandırılması ve sonuçta hatalı değerlendirilmesi ile PSNR değerinin düşmesi beklenen sonuçtur. Aynı zamanda eşik değerinin küçülmesi ile hareketli olarak sınıflandırılan blokların artması, bunun sonucu olarak da arama işlemi maliyetinin ve ortalama işlem süresi artması beklenir. Foreman gibi, hareket yoğunluğunun fazla olduğu veri setlerinde, küçük eşik değeri ile hareketli olarak sınıflandırılan blokların sayısı fazla olur ve doğru sonuca yaklaşılır. Bu nedenle önerilen

yöntemde eşik değeri, en iyi PSNR sonucu üreten ve işlem süresi olarak da tolere edilebilir bir artış gösteren değer olan %20 olarak seçilmiştir.

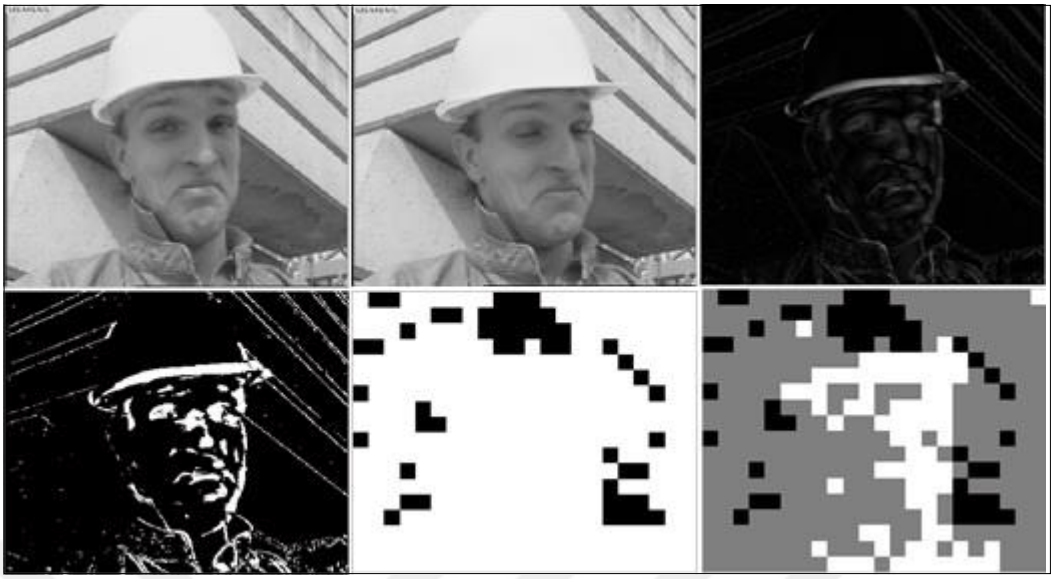
3.7 Hareket Kestiriminin Uygulaması

Hareket kestirim algoritması, hareketli ve yarı hareketli olarak sınıflandırılan bloklara uygulanarak hareket vektörleri elde edilmesi amaçlanmaktadır. Hareketli ve yarı hareketli bloklara tek bir arama algoritması yerine, farklı algoritmalar uygulanarak çerçeve genelinde ortalama PSNR oranı ve arama işlem sayısı yönünde geliştirme amaçlanmaktadır.

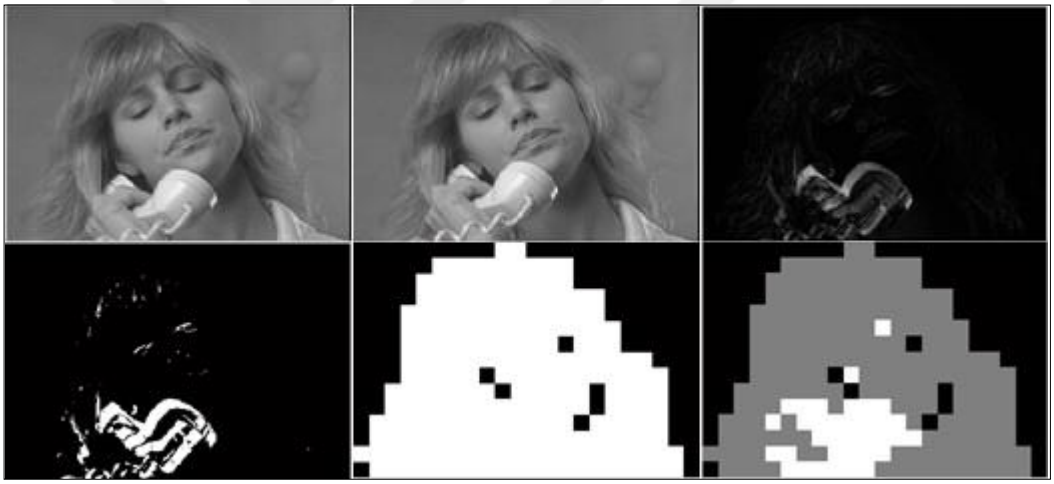
Durağan olarak sınıflandırılan bloklarda arama işlemi yapılmayacaktır. Durağan bloklarda hareket ihtimalinin yok sayılması nedeniyle, işlem maliyeti de göz önünde bulundurulduğunda bu bloklar es geçilecektir. Hareketli ve yarı hareketli olarak sınıflandırılan bloklardan yarı hareketli olarak sınıflandırılan bloklar, hareket ihtimalinin göz önünde bulundurulduğu bloklardır. Bu bloklarda eğer hareket varsa yüksek doğruluk oranı ile bulabilecek, aynı zamanda hareket yoksa da en az adımda bunu fark edebilecek bir arama algoritması kullanılmalıdır. Hareketli olarak sınıflandırılan bloklarda ise, doğruluk oranı en yüksek olan algoritma seçilerek, en doğru hareket vektörü elde edilmeye çalışılacaktır. Bu şekilde hem PSNR oranı yükseltilmeye çalışılacak, hem de arama işleminin maliyeti azaltılmaya çalışılacaktır.



Şekil 3.5 “boat” veri seti üzerinde uygulanan adımlar



Şekil 3.6 “foreman” veri seti üzerinde uygulanan adımlar



Şekil 3.7 “susie” veri seti üzerinde uygulanan adımlar

Veri seti olarak, hareket kestirim uygulamalarında sıkça kullanılan “boat”, “foreman” ve “susie” veri setleri tercih edilmiştir. “boat” veri setinde, arka plan deniz görselinden oluşmakta ve dalgalar arka planın sürekli hareket halinde olması sonucunu doğurmaktadır. Ön planda ise yavaşça hareket eden bir tekne görseli yer almaktadır. “foreman” veri setinde, hızlı şekilde hareket eden bir işçi ve hareket eden kamera sayesinde hareketli bir arka plan görseli yer almaktadır. “susie” veri setinde ise, sabit bir arka plan ve yavaş hareketler içeren bir insan görseli bulunmaktadır.

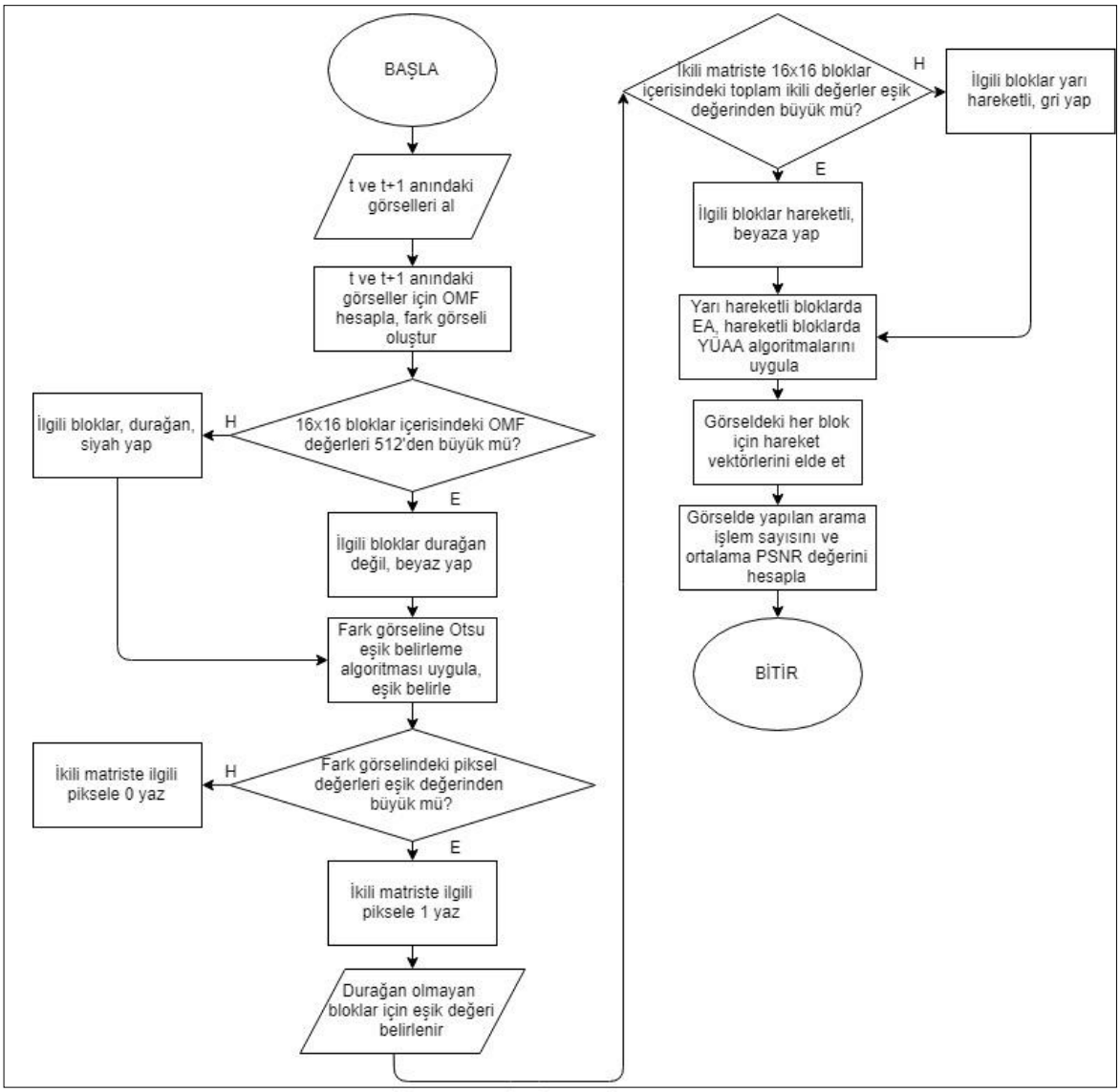
Üç veri seti olan “boat”, “foreman” ve “susie” için de önerilen algoritma sırasıyla Şekil 3.5, Şekil 3.6 ve Şekil 3.7’deki gibi uygulanmıştır. Şekillerde, uygulanan algoritmanın tüm adımları gösterilmeye çalışılmıştır. Buna göre, Şekil 3.5’teki ilk iki görsel, t ve t+1 zamanındaki iki görseli içermektedir. Hareket kestirimi bu iki görsel arasındaki farklılıklar aranarak uygulanacaktır.

Şekil 3.5’te üçüncü görsel olarak, t ve t+1 zamanında alınan görsellerin mutlak farkı alınarak oluşturulan fark görseli yer almaktadır. Fark görseli, referans ve mevcut görsellerin karşılıklı piksellerinin mutlak farkı alınarak oluşturulur ve iki görselin birbirinden ne kadar farklı ve hangi konumlarının farklı olduğu konusunda bilgi verebilir. Elde edilen fark görseli üzerinde, Otsu eşik belirleme yöntemi kullanılarak bir eşik elde edilir. Belirlenen eşik, fark görselindeki pikselleri değerlendirmekte kullanılır. Fark görselindeki piksel değerleri, Otsu eşik değeri ile kıyaslanarak, eşik değerinden büyük olan piksel yerlerine 1, küçük olan piksel yerlerine 0 değeri yazılarak Şekil 3.5’teki dördüncü görseldeki gibi bir ikili matris görseli elde edilir. Bu şekilde, beyaz piksellerin hareket eden alanlar, yani değişen pikseller olduğu, siyah piksellerin ise değişmeyen, sabit pikseller olduğu görülebilmektedir.

Bir sonraki adımda, bloklar ilk olarak durağan ve hareketli olarak sınıflandırılmıştır. Fark görselindeki, t ve t+1 zamanına ait görsellerin mutlak fark değerleri ele alındığında, 16x16 piksel boyutundaki bir blok içerisindeki mutlak fark değerleri 512 değerinden küçükse, o blok durağan olarak sınıflandırılmıştır. 512 değerinden büyük bloklar ise, bir sonraki adımda yarı durağan veya durağan olarak sınıflandırılmak üzere ayrılmıştır. Şekil 3.5’teki beşinci görselde, durağan bloklar siyah, diğer bloklar ise beyaz olarak gösterilmiştir.

Elde edilen ikili matris görseli, yarı hareketli ve hareketli blokların tespitinde kullanılmıştır. Bunun için, ikili matris görseli üzerinde 16x16 piksel boyutunda bloklar belirlenerek blok içerisindeki ikili sayılar toplanmıştır. Bu işlemde, durağan olarak belirlenen bloklar es geçilmiştir ve sadece Şekil 3.5’teki beşinci görselde, beyaz olarak işaretlenen bloklarda işlem yapılmıştır. Her blok içerisindeki toplam değer, belirlenen ortalama bir eşik değeri ile kıyaslanmış ve eşik değerinden büyük bloklar hareketli, küçük bloklar ise yarı hareketli olarak sınıflandırılmıştır. Şekil

3.5'teki altıncı görselde eşik değeri 50 olarak seçilmiş ve yarı hareketli bloklar gri renk ile, hareketli bloklar ise beyaz renk ile gösterilmiştir.



Şekil 3.8 Önerilen yönteme ait akış diyagramı

Elde edilen gri ve beyaz bloklar, sırasıyla yarı hareketli ve hareketli bloklar olarak sınıflandırılmıştır. Farklı arama algoritmaları gri ve beyaz bloklara uygulanmış ve siyah bloklarda arama işlemi yapılmamıştır.

Yarı hareketli ve hareketli bloklar için iki ayrı arama algoritması kullanılmıştır. Yarı hareketli bloklarda, hareket olma ihtimalinin öngörülmesi fakat hareket vektörünün merkez noktada olma olasılığının da yüksek olması nedeniyle, merkez yanlı bir arama algoritması kullanılması önerilmiştir. Ayrıca, hareket vektörünün

merkez noktada olması durumunda bunu en az adımda bulabilecek bir arama algoritması olması da, arama işlem maliyetini azaltmak adına önem taşımaktadır. Bu nedenlerle, gri olarak gösterilen yarı hareketli bloklarda EA algoritması tercih edilmiştir. Diğer algoritmalarla kıyaslandığında, merkez yanlı algoritmalar arasında, en iyi durumu en az adım sayısında bulan ve buna rağmen yüksek PSNR oranı ile yüksek doğruluk sağlayan bir algoritma olması tercih edilme sebepleridir. Hareketli bloklar için ise, en yüksek doğrulukta hareket vektörünün elde edilmesi önemlidir. Bu nedenle, hareket kestiriminde arama algoritmalarını kıyaslanması sonucu, en yüksek PSNR değerini elde eden YÜAA arama algoritması, hareketli bloklardaki hareket vektörünün aranması için tercih edilmiştir.

Şekil 3.8'de, önerilen yöntemin akış diyagramı gösterilmiştir. Buna göre, önerilen yöntemdeki hareket sınıflandırma yöntemi ve arama algoritmaları ile hareket vektörlerinin bulunması detaylı şekilde akış diyagramında gösterilmiştir.

4

Değerlendirme ve Sonuçlar

4.1 Önerilen Yöntemin Değerlendirilmesi

Önerilen algoritmada, YÜAA ve EA arama algoritmalarının bir arada kullanılmasıyla, ilgili algoritmaların ayrı ayrı kullanılmasına kıyasla PSNR oranı ve arama işlem sayısı kapsamında iyileştirmelerin sağlandığı görülmüştür.

Tablo 4.1 70 çerçeve için toplam arama işlemi sayısı

	YÜAA	**EA**	**Gri Bloklar: EA** **Beyaz Bloklar: YÜAA**	**Gri Bloklar: YÜAA** **Beyaz Bloklar: EA**
Foreman	514885	402726	350197	411122
Susie	460441	359738	266170	292999
Boat	344510	261661	214419	274820

Tablo 4.2 70 çerçeve için her makro blok başına düşen ortalama arama işlemi sayısı

	YÜAA	**EA**	**Gri Bloklar: EA** **Beyaz Bloklar: YÜAA**	**Gri Bloklar: YÜAA** **Beyaz Bloklar: EA**
Foreman	20,1442	15,7561	13,7010	16,0846
Susie	19,6518	15,3537	10,3667	12,5053
Boat	16,1742	12,2846	9,3500	12,9023

Tablo 4.3 "foreman" veri setinin tüm çerçeveleri için elde edilen, blok başına düşen arama işlemi sayısı

	YÜAA	**EA**	**Gri Bloklar: EA** **Beyaz Bloklar: YÜAA**	**Gri Bloklar: YÜAA** **Beyaz Bloklar: EA**
Foreman	22,1 [4]	13,4 [7]	12,6540	13,1297

Tablo 4.4 70 çerçeve için ortalama PSNR değerleri

	YÜAA	**EA**	**Gri Bloklar: EA** **Beyaz Bloklar: YÜAA**	**Gri Bloklar: YÜAA** **Beyaz Bloklar: EA**
Foreman	33,3228	33,2855	33,2303	33,3263
Susie	34,6618	34,4873	34,6124	34,4964
Boat	32,3999	32,4000	32,3987	32,3987

Veri setlerinden alınan 70'er görsel üzerinde sırasıyla ayrı ayrı YÜAA, EA, önerilen algoritma ve kıyaslama için önerilen yöntemdeki arama algoritmaları yer değiştirilerek uygulanmış ve çeşitli sonuçlar elde edilmiştir. Tablo 4.1'de, işlenen tüm çerçeveler içinde yapılan arama işlem sayısı, her bir veri seti için algoritma başına kıyaslanmıştır. Buna göre, gri bloklarda EA, beyaz bloklarda YÜAA kullanılan önerilen algoritmada en düşük arama işlem sayısı değeri elde edildiği görülmüştür. Arama işlem sayısı, makro blok başına hesaplanarak da bir sonuç elde edilebilir. Hesaplanan bu değer, arama algoritmalarının kıyaslanmasında bir parametre olarak kullanılabilir. Tablo 4.2'de, her bir veri seti için makro blok başına düşen arama işlem sayısı, algoritma bazlı olarak gösterilmiştir. Burada da beklendiği üzere en düşün değerler, önerilen algoritma olan gri bloklarda EA, beyaz bloklarda YÜAA algoritması kullanıldığında elde edilmiştir.

Tablo 4.3'teki değerlere bakıldığında, YÜAA algoritması için elde edilen değer, "foreman" veri setinin tamamı uygulandığında 22,1 olarak elde edilmekte [2], EA algoritması uygulandığında ise 13,4 olarak elde edilmektedir [4]. "foreman" veri

setinin tamamı ile işlem yapılıp kıyaslandığında, önerilen algoritmanın blok başına ortalama 12,654 arama işlem sayısı oranı ile sonucu iyileştirdiği söylenebilir.

Arama algoritmalarının kıyaslanmasında ve başarısının ölçülmesinde en önemli parametre olan PSNR değerleri, Tablo 4.4'te kıyaslanmıştır. İşlenen görüntünün hareket yoğunluğuna, arka planın ve ön planın hareketlerine, kameranın sallanmasına, ışık oranlarına bağlı olarak arama algoritmalarının başarım oranları değişmektedir. "foreman" ve "susie"veri setlerinde YÜAA daha yüksek PSNR oranı sağlarken, "boat" veri setinde EA daha yüksek PSNR oranı sağlamaktadır. Gri blokların EA, beyaz blokların YÜAA olduğu önerilen algoritmayla kıyaslandığında, tüm veri setleri için, YÜAA ve EA algoritmalarına çok yakın değerler verdiği görülmektedir. "susie" veri seti için EA algoritmasından daha iyi sonuçlar elde edilmiştir. "boat" veri seti için ise YÜAA ve EA'dan daha iyi sonuçlar elde edildiği görülmektedir. "foreman" veri seti için, gir bloklarda YÜAA, beyaz bloklarda EA kullanılan algoritmanın, hem önerilen algoritmadan hem de YÜAA ve EA algoritmalarından daha iyi sonuç verdiği görülmektedir. Bunun nedeni, bu veri setinde veri setinin teknik özellikleri nedeniyle YÜAA algoritmasının daha başarılı olması ve YÜAA algoritmasının gri bloklarda kullanılarak doğruluk oranını bu şekilde etkilemesinden dolayıdır. Önerilen algoritma, işlenen görüntünün özelliklerine göre bu şekilde de kullanılabilir.

YÜAA ve EA algoritmalarına çok yakın PSNR değerleri elde edilirken, arama işlemi sırasında kontrol noktalarında harcanan işlem maliyeti yönünden oldukça yüksek bir kazanım sonucu elde edilmiştir. Tablo 4.2'deki arama işlem sayıları hesaplandığında, YÜAA algoritmasına kıyasla arama işlem sayısı ortalama %37, EA algoritmasına kıyasla %19 azalmıştır.

5

Sonuç ve Öneriler

5.1 Sonuç

Bu çalışmada, görüntü dizilerinde hareket tespiti için bir algoritma geliştirilmiştir. Geliştirilen algoritma, görüntüyü 16x16 boklara ayırarak işlem yapmaktadır. İlk olarak hareketli ve durağan blokların tespiti, görüntülerin MFT değerleri üzerinden işlem yapılarak elde edilir. Hareketli bloklar, hareketli ve yarı hareketli olarak sınıflandırılmak üzere, Otsu eşik belirleme yöntemi ile, ve ardından elde edilen ikili görüntü değişim matrisi üzerinden işlem yapılarak sınıflandırılır.

Tablo 4.2 göz önüne alındığında, önerilen yöntemin arama işlem sayısını azaltması ve Tablo 4.4 göz önüne alındığında PSNR başarım oranının EA ve YÜAA algoritmalarına yakın sonuçlar vermesi yönünde iyileştirdiği görülmektedir. Elde edilen sonuçların, görüntü dizilerinin hareket karmaşıklığı, kameranın sabit olmaması, arka plan ve ön plan hareketlerinin aynı anda olması, görüntünün ışık miktarı gibi parametrelere de bağlı olduğu görülmüş olup; bu parametrelerden bağımsız olarak arama işlem sayısının YÜEE ve EA algoritmalarına göre sırasıyla %37 ve %19 azaldığı, PSNR oranının ise parametrelere bağlı olarak YÜAA ve EA algoritmalarına çok yakın değerler verdiği görülmektedir. Bu sonuçlara göre, önerilen algoritmanın hareket tespiti uygulamalarında bir alternatif olarak kullanılabileceği söylenebilir.

Gelecekte yapılan çalışmalarda, çalışmada ortaya konan hareket sınıflandırma yöntemi geliştirilerek daha fazla sınıflara ayrılması amaçlanmaktadır. Ayrıca, sınıflandırılan bloklarda kullanılan arama algoritmaları çeşitlendirilerek her sınıflandırılan blok için en etkin arama algoritması kullanılması hedeflenmektedir.

Kaynakça

[1] J. Jain ve A. Jain, "Displacement Measurement and Its Application in Interframe Image Coding", IEEE Transactions on Communications, vol. 29, no. 12, pp. 1799-1808, Dec 1981.

[2] R. Li, B. Zeng, ve M. L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 428-442, Aug. 1994.

[3] L. Po ve W. Ma. "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313-317, Jun. 1996.

[4] S. Zhu ve K. Ma "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, vol. 9, no. 2, pp. 287-290, Feb. 2000.

[5] B. Liu ve A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors", IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, no. 2, pp. 148-157, Apr. 2003.

[6] P. I. Hosur ve K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation", 1999

[7] S. Li, Q. Yi ve M. Shi "Fast Block-match Motion Estimation Based on Multilevel Adaptive Diamond Search", Physics Procedia, pp. 1783-1790, Dec. 2012.

[8] B. Pesquet-Popescu, M. Cagnazzo ve F. Dufaux, "Motion Estimation Techniques", TELECOM ParisTech, pp. 33-34, 2016.

[9] MPEG-4 Video Group, "MPEG-4 Video Verification Model Version 10.0", ISO/IEC JTC1/SC19/WG11 Document MPEG98/N1992, San Jose, Feb. 1998

[10] T. Ramaprabha ve M. M. Sathik, "Three step Vs Four step Block matching search algorithm in Stereo image Compression", nternational Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 6, pp. 167-170, Jun. 2012.

[11] D. D. Bhavsar ve R. N. Gonawala, "Three Step Search Method for Block Matching Algorithm", IRF International Conference, pp. 101-104, 2014.

[12] M. Gaikwad, "Implementation of four step search algorithm of motion estimation using FPGA", International Journal of Advanced Research in Computer Science and Electronics Engineering, vol. 1, no. 3, May. 2012.

[13] D. Zhang ve G. Lu, "Segmentation of Moving Objects in Image Sequence: A Survey", Circuits, Systems, and Signal Processing, vol. 20, no. 2, pp. 143-183, Mar. 2001.

[14] D. Liu ve J. Yu, “Otsu method and K-means”, 2009 Ninth International Conference on Hybrid Intelligent Systems, pp. 344-349, Aug. 2009.

Tezden Üretilmiş Yayınlar

İletişim Bilgisi: kaan1gencay@gmail.com

Konferans Bildirileri

1. K. Gençay, "Raspberry Pi Kullanarak Düşük Maliyetli Gerçek Zamanlı Hareket Tespiti ", 21. Akademik Bilişim Konferansı, 11-15 Şub 2019, Ordu.