

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SAYISAL VİDEO İNDEKSLEME İÇİN BÖLÜMLEME
ALGORİTMALARININ KARŞILAŞTIRILMASI VE
YENİ BİR ALGORİTMA GERÇEKLENMESİ**

YÜKSEK LİSANS TEZİ

Barış BORU

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞT.

Tez Danışmanı : Yrd. Doç. Dr. Halil İbrahim ESKİKURT

Mayıs 2007

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SAYISAL VİDEO İNDEKSLEME İÇİN BÖLÜMLEME
ALGORİTMALARININ KARŞILAŞTIRILMASI VE
YENİ BİR ALGORİTMA GERÇEKLENMESİ**

YÜKSEK LİSANS TEZİ

Barış BORU

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞT.

Bu tez 25 / 05 /2007 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Yrd.Doç.Dr. Halil İbrahim ESKİKURT
Jüri Başkanı

Prof. Dr. Hüseyin EKİZ
Üye

Yrd. Doç.Dr. İbrahim ÖZÇELİK
Üye

TEŐEKKÜR

Bu tezde, video indeksleme iin kullanılan mevcut video blmlendirme algoritmaları incelenerek kullanımların ve performans karŐılaŐtırmalarının yapılabileceėi arayzler tasarlanmış, mevcut algoritmaların dezavantajlarını giderecek yeni bir blmlendirme algoritması tasarımı yapılmıŐtır.

Bu alıŐmamın her safhasında bana yardımcı olan, gayretini esirgemeyen danıŐmanım Yrd. Do. Dr. Halil İbrahim EŐKİKURT'a, grŐleri ile bana yol gsteren Yrd. Do. Dr. Cabir VURAL'a, bu noktaya gelmemi saėlayan ve desteklerini hibir zaman eksik etmeyen aileme, ayrıca mesai arkadaŐım zdemir ETİN'e teŐekkr bir bor bilirim.

Mayıs 2007

BarıŐ BORU

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
ŞEKİLLER LİSTESİ	v
TABLolar LİSTESİ.....	vii
ÖZET.....	viii
SUMMARY.....	ix
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
SAYISAL GÖRÜNTÜ VE VİDEO.....	3
2.1. Sayısal Görüntü Temelleri	3
2.2. Sayısal Video.....	5
BÖLÜM 3.	
VİDEO VERİTABANLARI VE ERİŞİM.....	7
3.1 Video Veritabanlarında Arama Yapmanın Farklılıkları	8
3.2. Video İndeksleme.....	9
3.3. Video Bölümleme.....	10
3.4. Video Bölümlendirme Algoritmalarının Karşılaştıkları Zorluklar .	12
3.5. Anahtar Kare ve Özellik Çıkartımı.....	14
BÖLÜM 4.	
VİDEO BÖLÜMLENDİRME ALGORİTMALARI	16
4.1. Piksel Karşılaştırması.....	16

4.2. Blok Tabanlı Piksel Karşılaştırması.....	18
4.3. Histogram Tabanlı Yöntemler.....	19
4.3.1. Genel histogram karşılaştırması.....	21
4.3.2. HSV ve YIQ histogram karşılaştırması.....	24
4.3.3. Blok tabanlı histogram karşılaştırması.....	25
4.4. Kademeli Bölüm Geçişleri İçin İkiz Karşılaştırma Tekniği.....	26
BÖLÜM 5.	
FİLTRELENMİŞ VIDEO HİSTOGRAM KARŞILAŞTIRMASI.....	29
5.1. Giriş.....	29
5.2. Filtrelenmiş Video Histogram Karşılaştırması.....	30
5.3. Değerlendirme ve Karşılaştırmalar.....	36
BÖLÜM 6.	
VIDEO BÖLÜMLENDİRME ALGORİTMALARI UYGULAMA VE OTOMATİK PERFORMANS KARŞILAŞTIRMASI ARAYÜZLERİ.....	40
6.1. Video Bölümlendirme Algoritmaları Uygulama Arayüzü.....	40
6.2. Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzleri.....	44
6.3. Video Bölümlendirme Algoritmaları Performans Karşılaştırmaları.....	49
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER.....	56
KAYNAKLAR.....	58
ÖZGEÇMİŞ.....	60

ŞEKİLLER LİSTESİ

Şekil 2.1.	Örnek Bir Sayısal Resim Ve Piksel Yapısı.....	3
Şekil 2.2.	Örnek bir videoya ait ardışıl 8 kare.....	
Şekil 3.1.	Örnek bir video ve videoya ait bölümler	16
Şekil 3.2.	Bir videonun bölümlendirilerek özet bilgilerin bölümler referans edilerek veritabanına kaydedilmesi.....	11
Şekil 3.3.	a Örnek solma ve erime efektleri 3.4.b erime efekti solma efekti	12
Şekil 3.4.	Ardışıl karelerde oluşan bir kamera hareketi	13
Şekil 3.5.	Ardışıl karelerde oluşan parlaklık değişimi	13
Şekil 3.6.	Bir video bölümüne ait örnek bir anahtar kare.....	14
Şekil 4.1.	100×100 piksel boyutuna sahip örnek bir siyah-beyaz bir resim ve resme ait gri koyuluk değer histogramı.....	21
Şekil4.2.	a. Örnek bir videodan alınmış ardışıl iki kare bu karelere ait Kırmızı, Yeşil, Mavi Renklere ait Histogram grafikleri - b. Karelerin histogram farkları.....	23
Şekil4.3.	İkiz Karşılaştırma tekniği tarafından kademeli geçişlerin algılanması.....	27
Şekil5.1.	a. örnek bir video karesi - b.video karesinin sobel filtresi ile filtrelenmesi -c.video karesinin farklı ağırlıklara sahip sobel filtresi ile filtrelenmesi.....	31
Şekil5.2.	a. Örnek bir video karesi - b. Video karesinin motion blur filtresi ile filtrelenmesi.....	32
Şekil5.3.	Filtrelenmiş video histogram karşılaştırması filtreleme adımları..	33
Şekil5.4.	Filtrelenmiş Video Histogram Karşılaştırması algoritmasına ait akış diyagramı.....	35

Şekil5.5.	Obje ve kamera hareketleri içeren örnek bir video için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri.....	37
Şekil5.6.	Parlaklık değişimi ve kamera hareketlerinin yoğun olduğu örnek bir video için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri.....	38
Şekil 6.1.	Video Bölümlendirme Algoritmaları Arayüzü.....	41
Şekil 6.2.	Video Bölümlendirme Algoritmaları Arayüzü akış diyagramı	43
Şekil 6.3.	Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü.....	44
Şekil 6.4.	Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü akış diyagramı.....	47
Şekil 6.5.	Video Bölümlendirme Algoritmaları Performans Karşılaştırması Arayüzü grafiksel sonuçları.....	48
Şekil 6.6.	Chevrolet.avi videosu için bölümlendirme algoritmaları tarafından elde edilen kare geçiş değerleri	49
Şekil 6.7.	Chevrolet.avi videosu için Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü tarafından elde edilen karşılaştırma grafiği	51
Şekil 6.8.	As.avi videosu için bölümlendirme algoritmaları tarafından elde edilen kare geçiş değerleri	52
Şekil 6.9.	As.avi videosu için Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü tarafından elde edilen karşılaştırma grafiği	53

TABLULAR LİSTESİ

Tablo6.1	Chevrolet.avi videosu için algoritmaların elde ettikleri sayısal değerler Chevrolet.avi videosu için algoritmaların elde ettikleri sayısal değerler	51
Tablo 6.2	As.avi videosu için algoritmaların elde ettikleri sayısal değerler..	54

ÖZET

Anahtar kelimeler: Video İndeksleme, Video Bölümlendirme, Video veritabanlarına hızlı erişim, Anlamsal video erişimi

Sayısal videoların günümüzde analog videoların yerini alması, ucuz ve kolay kaydedilip depolanabilmesi gibi özelliklerinden dolayı binlerce saatlik sayısal video arşivleri ortaya çıkmıştır. Bu video veritabanlarına erişimin hızlı olması için video içinde anlamsal olarak arama yapılabilmesi büyük bir ihtiyaç haline gelmiştir. Sayısal videolar içinde standart resim ve metin arama teknikleri kullanılmadığından video indeksleme konusu ilgiler artarak literatürde yer alan birçok çalışma yapılmıştır. Sayısal video indeksleme yöntemi videoları bölümlendirerek bölümlere ait özet bilgilerin çıkarılması ile uygulanmaktadır. Bu bağlamda doğru video bölümlendirme video indekslemenin temelini oluşturan en önemli kısım olarak ortaya çıkmıştır.

Bu çalışmada sıkıştırılmamış videolar üzerine çalışan bölümlendirme algoritmaları incelenerek performans karşılaştırması yapılmıştır. Algoritmaların pratik uygulamalarını mümkün kılacak bir arayüz tasarımı yapılmıştır. Algoritmaların performans karşılaştırmalarını otomatik olarak yaparak kullanıcıya sayısal değerler ve grafikler halinde sonuçları verecek bir arayüz tasarlanmıştır. Ayrıca mevcut video bölümlendirme algoritmalarından daha başarılı bölümlendirme sonuçları elde eden Filtrelenmiş Video Histogram Karşılaştırması algoritması tasarlanmıştır.

STATICAL ANALYS TO SUPPLY CHAIN MANAGEMENT

SUMMARY

Key Words: Video indexing, Video segmentation, Rapid access to video databases, Semantic retrieval of video

Today, the use of digital videos instead of analog videos has become most popular in terms of easily recordable and low-cost storage. Therefore, huge amount of digital video archives have been came out dramatically. In order to access this video database very fast, the semantic search in a video has been an important demand. While the standard picture and text searching methods cannot be used in a digital video, the video indexing has become a popular interest, and lots of research have been came out. Digital video indexing methods are implemented by segmenting the videos into sub-scenes and extracting the main data related to these sub-scenes. In this sense, correct video segmenting has been most important part of the video indexing.

In this study, video segmenting algorithms about uncompressed videos have been investigated and compared with their performance analysis. An interface software has been designed for making the practical applications of these segmenting algorithms possible. Also different interface software has been designed for comparing the performance analysis of the segmenting algorithms automatically and giving the results to the user in graphically and numerically. Added to these, an algorithm called Filtered Video Histogram Comparison has been implemented which has more successful segmenting results than existing video segmenting algorithms.

BÖLÜM 1. GİRİŞ

Çağımızda bilgisayarların, elektronik cihazların gelişmesi ve herkes tarafından edinilebilecek kadar düşük maliyetlere ulaşması, internet'in hayatımızın bir vazgeçilmezi haline gelmesi gibi sebeplerle hemen hemen bütün bilgi türleri dijital formatta saklanmaktadır. Son yıllarda tüm bilgi türleri gibi videolarda analog formatlar yerine dijital formatlarda saklanmaktadır. Bu bağlamda çeşitli ortamlarda içinde on binlerce saatlik kayıtlar bulunan büyük video veritabanları ortaya çıkmıştır. Bu denli büyük veritabanlarında istenilen videoya hızlı erişim ve video içinde anlamsal arama yapmak ihtiyaç haline gelmiştir. Ancak dijital videolara hızlı erişim ve anlamsal arama için standart dijital bilgi erişim teknikleri kullanılamamaktadır.

Sayısal videolar için hızlı erişimi mümkün kılmak için video indeksleme fikri ortaya atılmıştır. Video indeksleme sayısal video içinde videonun tamamında eşleştirme ve arama yerine videonun özetinin çıkarılması ile arama kriterlerini düşürerek, yapılan aramaların özet içinde yapılması ve özete ait video parçasına erişme mantığıyla çalışmaktadır. Video özetini elde etmek için video içinde benzer alanların gruplanarak her grup için tek bir kriter çıkarılması gerekmektedir. Bu amaçla video içinde içeriğin aynı kaldığı ardışıl kareler tespit edilerek videolar bölümlendirilmektedir. Video indeksi hazırlanırken önce video, bölümlendirme algoritmaları ile bölümlerine ayrıştırılmakta ardından ayrıştırılan bölümler için özet bilgiler çıkartılarak bir veritabanında saklanmaktadır.

Yapılan tez çalışmasında sıkıştırılmamış video bileşenleri üzerine çalışan bölümlendirme algoritmaları üzerinde durulmuştur. Tez çalışmasında mevcut bölümlendirme algoritmalarının uygulanması ile performans karşılaştırmaları yapılmıştır. Bölümlendirme algoritmalarının kullanılmasını ve otomatik performans karşılaştırmasının yapılmasını sağlayacak arayüzler hazırlanmıştır.

Tez çalışması sırasında mevcut bölümlendirme algoritmalarının dezavantajlarını kapatacak yeni bir video bölümlendirme algoritması geliştirilmiştir. Geliştirilen algoritma diğer algoritmalar ile karşılaştırılarak değerlendirme ve yorumlar teze eklenmiştir.

Tez içinde konuyu hakkında bilgiler verilmeden önce Bölüm 1’de sayısal resim ve video hakkında temel bilgiler verilmiştir.

2. Bölümde video veritabanları, sayısal videolarda arama ve hızlı erişim ve video indeksleme hakkında genel bilgiler verildikten sonra video bölümlenme üzerine durulmuştur. Video bölümünün açıklanması ardından video bölümlenme hakkında genel bilgiler, video bölümlenme algoritmalarının karşılaştıkları zorluklar ve video bölüm türleri hakkında bilgiler verilmiştir.

3. Bölümde literatürde yer alan sıkıştırılmamış video bileşenleri üzerine çalışan bölümlenme algoritmaları anlatılmıştır.

4. Bölümde Tez çalışması sırasında geliştirilen Filtrelenmiş Video Histogram Karşılaştırması yöntemi detaylarıyla anlatılmıştır. Önerilen yöntem genel özelliklerinden bahsedilerek değerlendirilmesi sunulmuştur.

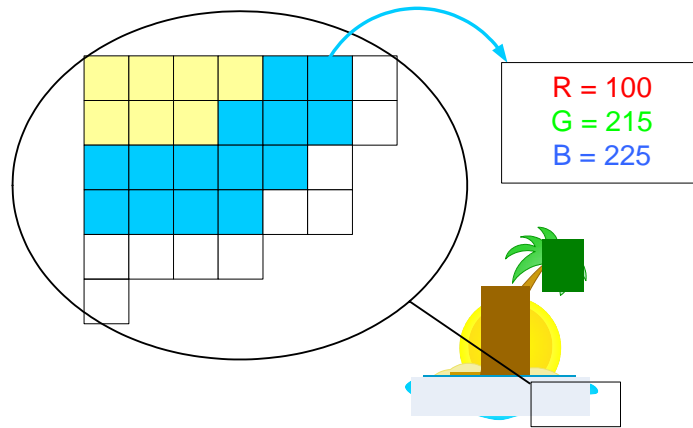
5. Bölümde geliştirilen arayüzler çalışma mantıkları ile anlatılarak kullanımları hakkında bilgiler verilmiştir. Video bölümlendirme algoritmalarının performans karşılaştırmaları grafik olarak sunulmuş yöntemler hakkında değerlendirme ve yorumlara yer verilmiştir.

BÖLÜM 2. SAYISAL GÖRÜNTÜ VE VİDEO

Günümüzde bilgisayarların ve multimedya özelliklerine sahip elektronik cihazların artması ile görüntülerin sayısal formatta saklanması ihtiyacı artmıştır. Ve yine analog görüntüler ve analog görüntü kayıtları yerini tamamen sayısal görüntülere bırakmıştır. Günümüzde analog olarak kullanıcıya sunulan görüntüler bile sayısal video formatında hazırlanır devamında analog görüntüye çevrilerek kullanıcıya sunulmaktadır.

2.1. Sayısal Görüntü Temelleri

Bütün sayısal görüntüleme ortamlarında kullanılan görüntü alanları (televizyonlar, monitörler. Projeksiyon cihazları vs.) küçük noktaların bir araya gelmesi ile oluşturulmaktadır. Ekranda oluşturulan görüntüler bu küçük noktaların her birinin ilgili renge boyanması ile oluşturulur. Bu küçük noktalara piksel denilir ve her piksel sadece bir renk alabilir.



Şekil 2.1 Örnek bir sayısal resim ve piksel yapısı

Şekil 2.1 de örnek bir sayısal resim görünmektedir. Şekilde resmin seçilen belli bir alanına ait piksel yapısı yakınlaştırılarak sunulmuştur. Piksel yapısı incelendiğinde her bir pikselin tek bir renk aldığı görülmektedir. Pikseller ekranda çok küçük noktacıklardır ekrana bakan insan odaklandığı piksel alanını bütün bir resim olarak görecektir. Sayısal resimleme tekniğinde uygulanan yöntem gerçek görüntünün küçük noktacıklarla ifade edilmesidir.

Gerçek görüntüyü temsil edecek nokta sayısı ne kadar fazla olursa sayısal görüntü gerçek görüntüye o kadar yakınlaşacaktır. Gerçek görüntü sayısal görüntüye çevrilirken kullanılacak piksel sayısı yeterli seçilmezse içeriğinde farklı renkler bulunan(özellikle görüntüde nesne kenarlarının bulunduğu) küçük alanlar tek bir renk ile ifade edileceğinden net bir görüntü oluşturulamaz. Sayısal görüntüye bakan insan görüntüyü bulanık, nesne kenarlarının net algılanamadığı bir resim olarak görür.

Sayısal görüntüde, görüntü alanında yatay ve dikey olarak kaç piksel kullanıldığı bilgisine çözünürlük bilgisi denir. Örneğin bir resim için 640*480 çözünürlükte bir resim ifadesi kullanıldığında; bu resim alanının dikey olarak 480, yatay olarak 640 piksel kullanılarak oluşturulduğu ($640 \times 480 = 307200$ piksel içerdiği) anlaşılır. O halde bir sayısal görüntü için çözünürlük ne kadar yüksekse gerçek görüntüye o kadar yakın bir görüntüdür (o kadar nettir) denilebilir.

Sayısal görüntü bilgisi içinde bulunan bilgi, piksellerin adreslerine göre renk koyuluk değerleridir. Piksellerin renk koyuluk değerlerini ifade etmek için üç ana rengin (kırmızı, yeşil, mavi) ne oranlarda birbirlerine karıştırıldığı bilgisi verilir. Bu yaklaşım tüm gerçek renkleri üç ana rengin değişik oranlarda karıştırılarak elde edilebileceği bilgisine dayanarak ortaya çıkmıştır. Sayısal görüntüde her bir piksel için üç renk bileşeninin koyuluk değerlerinin verilmesi rengin oluşturulması için yeterlidir. Örneğin beyazı elde etmek için kırmızı, yeşil, mavi renklerinin hepsi tam koyulukla karıştırılması gerekmektedir. Bu tanımlama bir standart haline gelmiştir ve RGB resim kodlama olarak adlandırılmaktadır.

RGB resim kodlama her renk koyuluk deęerinin nitelendirileceęi en byk sayı, renk kalitesini belirler. rneęin renkleri belirtmek iin 4 bit seilirse; her renk iin en koyu ve en aık deęeri arası 16 ($(1111)_2 = (15)_{10}$) deęer ile ifade edilecektir ki bu da renkleri ayrıntılı ifade etmekten uzaktır. Aslında her rengi belirtmek iin ne kadar az bit kullanılırsa o resim dosyasının boyutu o oranda dşecek ve hafızada o kadar az yer kaplayacaktır. RGB kodlama teknięinde her renk iin 1 byte (8 bit) kullanılması standart haline gelmiřtir, bu durumda bir pikseli ifade etmek iin 3byte hafıza alanına ihtiya vardır.

RGB kodlama 24 bit olarak yapılırsa her renk bileřeni 256 koyuluk deęeri ile nitelendirilecektir. Bu durumda RGB renkler iin 0 deęeri ilgili rengin piksel ierięinde hi bulunmadıęını belirtir, 255 deęeri ise ilgili rengin piksel ierięinde tam koyulukta bulunduęunu belirtir. rneęin RGB 0,0,0 deęerindeki piksel siyah, RGB 255,255,255 deęerindeki piksel beyaz renktedir. řekil 2.1' de RGB kodları gsterilen mavi renkli piksel 100 kırmızı aęırlıęına, 215 yeřil aęırlıęına, 225 mavi aęırlıęına sahiptir.

2.2. Sayısal Video

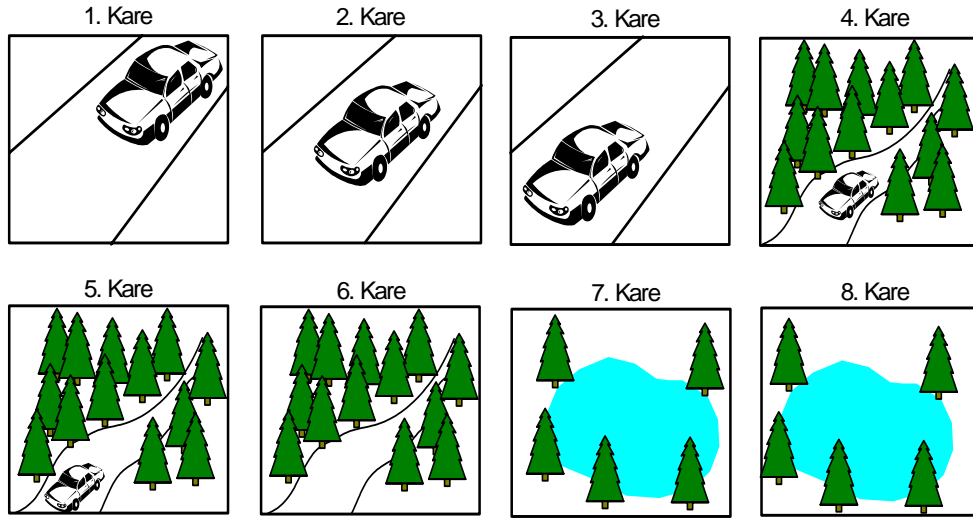
Bilgisayarların ve dijital cihazlarının yaygınlařması, aynı zamanda dijital haberleřmenin analog haberleřmeden daha kolay uygulanabilir hale gelmesi ve yaygınlařması ile tm bilgi trlerinde olduęu gibi video bilgisinin de dijital ortama aktarılması gereksinimi ortaya ıkmıřtır.

Sayısal video insan gznn sahip olduęu zaafiyetlerden yararlanarak uygulanmaktadır. İnsan gz saniyede 15-20 kereden hızlı deęiřen grntleri hareketli grnt olarak algılamaktadır. En hassas insan gz bile saniyede 20 deęiřimin zerini algılayamamaktadır. Sayısal video resim erevelerinin saniyede 15 kareden hızlı deęiřimlerle gsterilmesi ile uygulanmaktadır. Sayısal video, video sahnelerinin saniyede 15~30 kare olarak fotoęraflanması ile elde edilir.

Sayısal videonun saniyede ka kare ekilerek kayıt edildięi bilgisine fps (saniyedeki kare sayısı) denilmektedir. Sayısal video hangi fps hızı ile kaydedildi ise o hız ile

ekranda gösterilmelidir. Aksi takdirde video olduğundan daha hızlı veya daha yavaş sunulmuş olacaktır.

Yukarıda anlatılanlardan da anlaşıldığı gibi sayısal video bilgisi ard arda gelen sayısal resimlerden oluşan sayısal bir bilgidir. Eğer video içerisinde ses kaydı var ise resim kareleri ile birlikte ses bilgilerini de içermektedir.



Şekil 2.2. Örnek bir videoya ait ardışık 8 kare

Sayısal videoların kullanılmaya başlamasından itibaren sayısal video teknolojileri çok fazla gelişmiştir. Bu süreç içinde birçok sayısal video standardı geliştirilmiştir. Geliştirilen standartların birçoğu farklı sıkıştırma yöntemlerinin uygulanması ile daha az yer kaplayarak daha kaliteli videonun mümkün kılınması üzerine yoğunlaşmıştır. Ayrıca yarı iletken teknolojilerinin ve bilgisayarların gelişimine paralel olarak ilk sayısal videolara göre çok daha kaliteli sayısal video kaydedilip saklanma imkanı ortaya çıkmıştır.

BÖLÜM 3. VIDEO VERİTABANLARI VE ERİŞİM

Günümüzde sayısal video'nun analog videonun yerini alması ile birlikte sayısal video kayıt eden kameraların, sayısal video oynatıcıların ve bilgisayarların maliyetlerinin düşmesi nedeniyle bütün videolar sayısal video formatında saklanmaktadır. Özellikle güvenlik görüntü kayıtları, haber görüntü kayıtları arşivlenerek saklanmaktadır. Büyük sistemlerde binlerce saatlik video kayıtları tutulmaktadır.

Sayısal video kayıtlarının bu kadar arttığı bir ortamda sayısal videolar içinde arama yapılması ile hızlı erişim gereksinimi ortaya çıkmıştır. Örneğin güvenlik videolarının tutulduğu bir arşivde bir şahsa, bir araca ait görüntü kaydının bulunup bulunmadığı sorgulanması gerekmektedir. Haber videoları içerisinde bir konu ile alakalı eski haberlere ulaşmak istenmesi de sık karşılaşılan bir durumdur. Günümüz sayısal video teknolojilerinde bu tarz bir arama yapılması mümkün değildir. Bu tarz bir aramayı ancak insan faktörü ile yapmak mümkündür. Video arşivini hiyerarşik bir sınıflandırma ile saklandığı durumlarda bile böyle bir arama büyük bir zaman kaybı oluşturmaktadır.

Yukarıda sayılan gereksinimler sayısal video üzerinde çalışan araştırmacıları sayısal video içerisinde anlamsal aramayı mümkün kılan yöntemler geliştirmeye teşvik etmiştir. Bu maksatla son yıllarda sayısal video veritabanlarına hızlı erişim teknikleri birçok araştırmacının odağı olmuştur. Halen birçok üniversitede kurulan araştırma grupları ve sayısal teknolojiler üzerinde ar-ge çalışmaları yapan kuruluşlar tarafından bu konu üzerine araştırmalar sürmektedir.

Yapılan çalışmalar birçok yeni tekniğin geliştirilmesi ile neticelenmiştir. Fakat halen geliştirilen teknikler bu konuda pratik uygulamayı mümkün kılmamaktadır. Geliştirilen teknikler arasında içeriği belli bir türde olan videolarda başarılı sonuç

veren teknikler olsada; içeriğinde farklı videoların bulunduğu veritabanlarında kullanılabilir ortak bir teknik henüz geliştirilememiştir. Bu bilgiler ışığında sayısal video veritabanları içerisinde anlamsal aramayı mümkün kılan hızlı erişim teknikleri üzerine önümüzdeki yıllarda birçok araştırmanın yapılacağı açıktır.

3.1. Video Veritabanında Arama Yapmanın Farklılıkları

Sayısal video içerisinde aramanın mümkün olup olmadığı düşünüldüğünde ilk akla gelen videonun ardışıl resimlerden oluştuğu ve resim tanıma ve karşılaştırma tekniklerinin bu amaçla kullanılabilirliği. Fakat bu fikir pratikte uygulanabilir bir yöntem değildir. Çünkü sayısal videonun boyutuna bağlı olarak içinde milyonlarca resim bulunabilir. Örneğin 30fps hızında kaydedilmiş 1saatlik videoda 1080000 (30fr×60sn×60dk) adet resim bulunmaktadır. Bir saatlik video için bile bu resimlerin tamamının karşılaştırılması hızlı bir bilgisayarda bile saatler almaktadır. Bu karşılaştırma işleminin binlerce saatlik bir video veritabanında yapılması çok uzun bir zaman alacağı için kullanılabilirlikten uzaktır.

Sayısal video her ne kadar ardışıl resimlerden oluşsa da bu resimlerin hızlıca oynatılması ile hareketli bir görüntü ortaya çıkmakta ve bu görüntü bir anlam ifade etmektedir. Örneğin hareket eden bir arabaya ait bir video'nun kareleri ayrı ayrı incelenirse araba resmi olarak algılanır. Fakat video olarak oynatıldığında trafikte seyir eden bir araba anlamını vermektedir. Eğer bu video içinde araba resmi arandığı senaryoda her ne kadar yavaşta olsa standart resim karşılaştırma teknikleri kullanılırsa araba algılanabilir. Fakat video içinde trafikte hareket eden araba tanımlaması aranırsa bir sonuç elde edilemeyecektir. Bu yüzden sayısal video içinde anlamsal olarak bütünlük ifade eden kısımlara ait kareler birlikte değerlendirilmelidir.

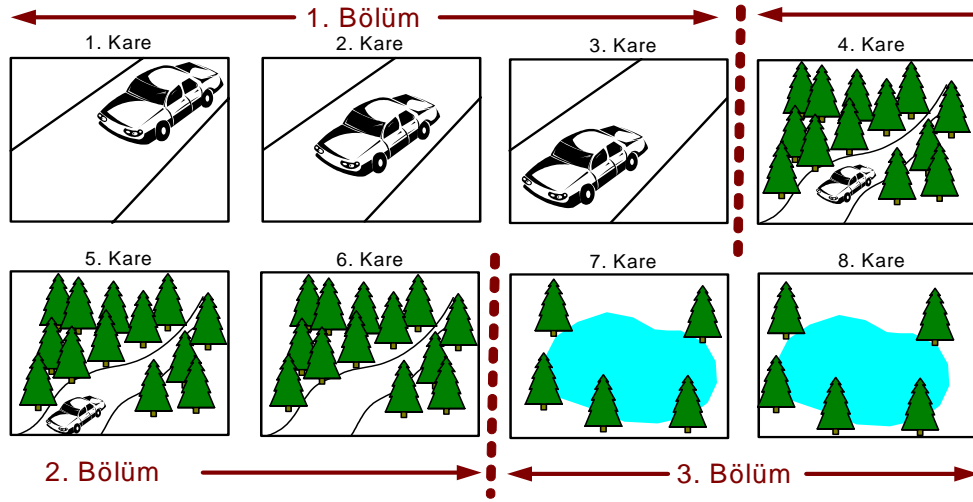
Sayısal video bilgisi bilindiği gibi içinde sadece görüntü öğeleri barındırmaz görüntü ile birlikte ses öğeleri barındırır. Bu yüzden bazı anlamsal arama türlerinde ses bilgisinin görüntü bilgisi ile birlikte değerlendirilmesi gerekmektedir. Örneğin tezahürat yapan bir kalabalık araması yapıldığında ses bilgisinin değerlendirilmesi gerekecektir.

3.2. Video İndeksleme

Yukarıda da anlatıldığı gibi standart arama ve erişim teknikleri video indeksleme için kullanılamaz. Bu yüzden video erişim için özel teknikler kullanılmalıdır. Video erişim ile ilgilenen araştırmacılar, videonun tamamında arama yapmak yerine videonun önceden işlenmesiyle elde edilen video indeksinde aramanın ve uygun bulunan sonuçların referans ettiği video parçasına erişmenin daha uygun olacağı fikrini ortaya atmışlardır. Ardından gelen çalışmalar bu fikir üzerine gelişmiş ve bu mantık video erişim için temel yapı taşı olmuştur.

Video indeksleme nedir neden gereklidir sorusunun cevabını biraz daha net açıklamak için bir örneğe başvurmak faydalı olacaktır. Örneğin C programlama ile alakalı bir kitapta printf fonksiyonunun kitabın neresinde olduğunu bulmak için yapılması gereken, kitabın içindekiler veya indeks kısmında printf'i tarayarak bulunan öğelerin referans edildiği sayfaları direk açılmasıdır. Kitabın içindekiler ve indeks kısmı kullanılmadan bütün sayfaların tek tek aranan öğe için taranması bir hayli vakit kaybı olacaktır. Arama yapılacak bir videoda bütün bilgilerin her seferinde taranması, kitapta sayfaların tek tek taranması ile erişim de olduğu gibi erişimi zorlaştıracaktır. Bu yüzden video içinde arama yapmak ve erişimi mümkün kılmak için kitapta olduğu gibi videonun indeksini çıkarmak ve özel bir veritabanında saklamak uygun olacaktır. Video veritabanına eklenen her video, bir dizi işlemde geçirilerek indeksi elde edilerek saklanmalıdır. Arama yapıldığında aranan öğe indekste taranacak eşleşme bulunursa referans ettiği video parçası kullanıcıya sunulacaktır.

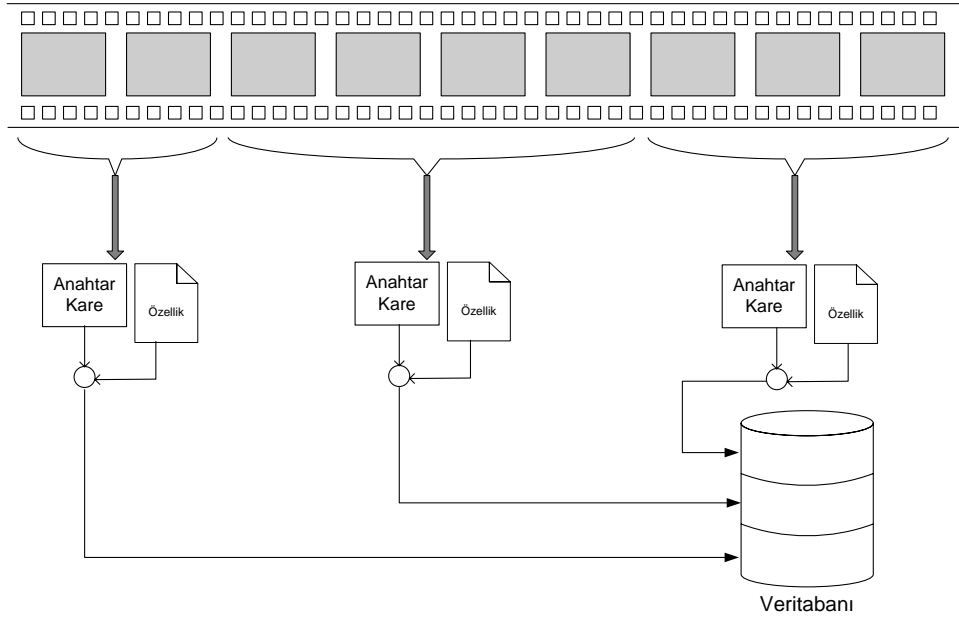
3.3. Video Bölümler



Şekil 3.1. Örnek bir video ve videoya ait bölümler

Video’da yapılan arama neticesinde kullanıcıya arattığı kriterleri sağlayan video parçasının veya parçalarının tamamı sunulmalıdır. Örneğin bir şahısa ait video görüntüleri aratılacaksa o şahsın görüntüye girdiği andan çıktığı ana kadar olan bütün parçanın kullanıcıya sonuç olarak verilmesi gerekmektedir. Aynı şekilde trafikte seyreden araba tanımlaması arandığında videoda trafikte seyreden araba görüntüsünün başladığı andan bittiği ana kadar video parçası sonuç olarak elde edilmelidir. Bu bağlamda video, bulunan neticelerde bölümün tamamına ulaşılabilmesi için ayrıştırmalı ve bölüm hakkında özet bilgiler oluşturulan indekse kaydedilmelidir (Şekil 3.3).

Video erişimlerinin indeksleme sayesinde kolay ve hızlı hale gelmesinin ana sebebi eşleştirilmesi gereken bilgi sayısının özetlenerek azaltılmasıdır. Videolar içinde aynı görsel içeriğe sahip birçok kare ard arda gelmektedir (video sahneleri). Bilindiği üzere filmler sahnelere ayrılarak kaydedilirler ve görsel sahne içeriği hemen hemen aynı kalır. Sahne ayrılıkları sadece filmler için geçerli değildir. Örneğin bir haber programı videosunda içerik genellikle stüdyonun gösterilerek spikerin sunum yapması, ardından haber kaydının gösterilmesi ve haber kaydı ardından tekrar spikerin diğer bir haber için sunum yapması şeklinde gelişmektedir. Bu şekilde bir video örneğinde ayrıştırılması gereken bölümler spikerin görüntüde kaldığı kareler dizisi ve habere ait görüntünün kareler dizisidir.



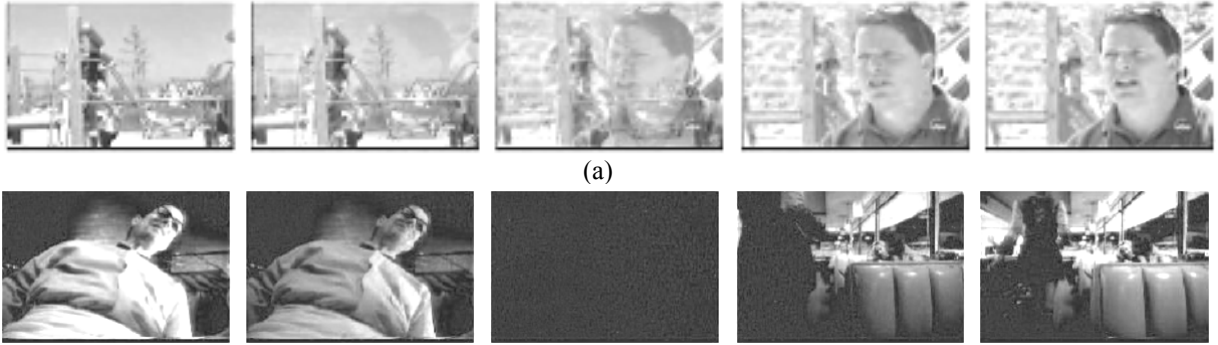
Şekil 3.2. Bir videonun bölümlendirilerek özet bilgilerin bölümler referans edilerek veritabanına kaydedilmesi

Yukarıda belirtilen sebeplerden dolayı video indekslemenin en önemli parçası video bölümlendirmektir. Video bölümlendirme indekse eklenecek bilgileri azaltma hedefine, benzer öğeleri gruplandırarak tek bilgi ile onları nitelendirme imkânını sunarak katkıda bulunmaktadır. Aynı zamanda arama neticesinde gösterilecek bölümlerin önceden tespit edilerek hazırlanmasını sağlamaktadır.

Literatürde yer alan video veritabanları ve erişim konusundaki çalışmaların birçoğu video bölümlenmeyi konu edinmektedir. Bu çalışmaların temel hedefleri doğru bölüm geçişlerini algılayarak bölümlendirme yapmaktır.

Video bölüm geçişleri ani ve kademeli bölüm geçişi olarak iki türe ayrılmaktadır. Ani bölüm geçişi tanımlaması bir önceki kare ile bir sonraki karenin birbirinden farklı olduğu durumlar için kullanılır. Ani bölüm geçişleri genellikle film çekimlerinde olduğu gibi kameranın kaydının durdurulması ve farklı bir görüntü için yeniden başlatılması ile oluşmaktadır. Kademeli bölüm geçişleri ise görüntü içeriğinin kademeli olarak değiştirilmesi ile oluşmaktadır. Bu tarz geçişler sinema efektlerinin iki bölümü birleştirmek için kullanılmasıyla oluşmaktadır. Kademeli geçişlere en iyi örnekler bölümün solarak geçmesi (fade effect) ve bölümün eriyerek geçmesidir (dissolve effect). Solma efekti siyah bir resimden başlayarak parlaklık

değerinin gittikçe artmasıyla diğer bölüm başlangıcının görüntüye girmesi ile oluşturulur. Erime efekti ise bir karenin diğerinin üzerine ilkinin karartılarak ikincisinin parlaklaştırılarak yerleştirilmesi ile elde edilir.



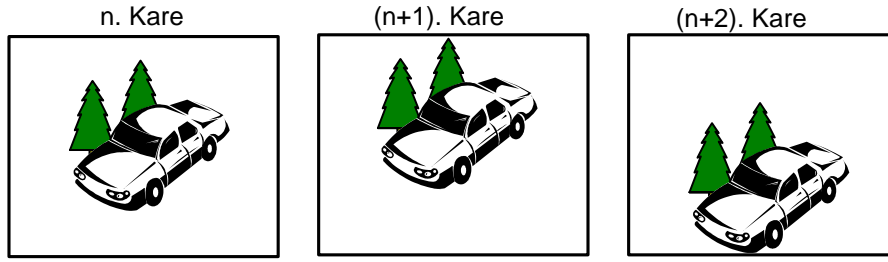
Şekil 3.3..a Örnek solma ve erime efektleri 3.4.b erime efekti solma efekti

(b)

Kademeli bölüm geçişlerini algılamak ani bölüm geçişlerini algılamaktan daha zordur. Bu algılamaların kamera ve obje hareketinden ayrıştırılmaları gerekmektedir. Özellikle parlaklık hareketlerini içeren bölüm geçişlerini algılamak diğer geçiş türlerine göre daha zordur.

3.4. Video Bölümlendirme Algoritmanın Karşılaştıkları Zorluklar

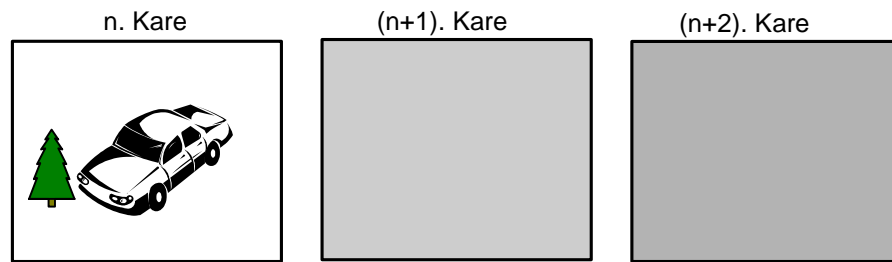
Video bölümlendirme algoritmaları genellikle ardışıl kareler arası görsel değişim oranlarını elde ederek değişimin oranının yüksek olduğu anlarda bölümler arası bir geçiş olduğu mantığına dayanarak bölüm algılaması yapmaktadırlar. Bölümlendirme algoritmaları videonun 1. karesinden başlayarak tarama yaparlar ve arka arkaya gelen video karelerini değerlendirirler. Bu değerlendirme sırasında sahne içeriğinin aynı kaldığı fakat sayısal anlamda bir farkın bulunduğu anları bölüm geçişi gibi algılayabilmektedirler. Böyle senaryolarda video gerçekte olduğundan daha fazla bölüme bölünecek ve indeksleme doğru yapılamayacaktır. Sahne içeriğinin değişmeden aynı kalıp sayısal anlamda fark olduğu durumlara en belirgin örnekler kamera-obje hareketleri ve parlaklık değişimleridir.



Şekil 3.4. Ardışıl karelerde oluşan bir kamera hareketi

Kamera hareketi Şekil 3.5’de görüldüğü gibi videonun çekildiği esnada kameradan veya kameramandan kaynaklanan hareketlerden dolayı görsel içeriğin aşağı-yukarı, sağa-sola hareket etmesidir. Bu durumda anlamsal olarak bölüm devam ettiği halde sayısal olarak fark değerlendirmesi yapıldığında yüksek bir fark ortaya çıkmaktadır. Çünkü bir önceki kare ile bir sonraki karede görüntü tamamen konum değiştirdiğinden bu karelerde örtüşen piksellerin değerleri tamamen değişmiş olacaktır.

Objeye hareketi, bölüm görsel içeriği içinde bulunan objelerin ve arka planın değişmediği fakat bir veya birden fazla objenin hareket ettiği durumlarda ortaya çıkmaktadır. Objeye hareketleri anlamsal olarak farklılık olmadığı halde ardışıl küçük alanlarda piksel değerlerinin değişmesi ile neticelenecektir. Bölümlendirme algoritmalarını bu küçük farklılıkları tolere edecek şekilde bölümlendirme yapmaları gerekmektedir.



Şekil 3.5. Ardışıl karelerde oluşan parlaklık değişimi

Parlaklık değişimi, Şekil 3.6’da görüldüğü gibi ardışıl kareler arası görsel içeriğin değişmeden kaldığı kare parlaklığının değişik olarak resimlenmesi ile ortaya çıkmaktadır. Parlaklık değişimi genellikle kameradan kaynaklanmaktadır. Kameraya düşen ışık miktarının değişimi ile sahne daha parlak veya daha karanlık

resimlenmektedir. Genellikle kamera hareketleri ile parlaklık deęişimleri aynı anda ortaya çıkmaktadır kamera hareket ettirildiğinden ışık kaynağı ile kamera merceęi arasındaki açı deęiştiiğinden görüntü daha aydınlık veya daha karanlık resimlenmektedir. Video bölümlendirme algoritmaları parlaklık deęişimlerinden etkilenmeyecek şekilde tasarlanırken yapı olarak parlaklık deęişimine benzeyen solma efektini algılayamayacak bir yapıda tasarlanmamalıdır. İdeal bölümlendirme algoritması parlaklık deęişimi ile solma efektini birbirinden ayırabilmelidir.

3.5. Anahtar Kare ve Özellik Çıkartımı

İndekslenerek video bölümlere ayrıldıktan sonra bu bölümleri nitelendirecek ve indekse eklenecek özelliklerin çıkarılması gerekmektedir. Bölümle alakalı çıkarılması gereken özellikler kullanılan indeksleme türüne göre farklılıklar gösterebilmektedir. Örneğin sadece görsel aramayı mümkün kılacak bir indeksleme yapılacaksa bölümdeki tüm görsel öğeleri ifade edecek bir anahtar kare elde edilmelidir. Eğer anlamsal aramayı mümkün kılacak bir indeksleme yapılacaksa anahtar kareler ile birlikte anlamsal özellikler çıkarılmalı ve indekse kaydedilmelidir.



Şekil 3.6. Bir video bölümüne ait örnek bir anahtar kare

Bölümle alakalı indekse eklenmesi gereken vazgeçilmez öğe anahtar karedir. Anahtar kare bölüm içinde bir video karesi olarak seçilecekse bölümde bulunan karelerin tamamına en yakın özelliklere sahip kare kullanılmalıdır. Aksi taktirde video içinde bulunan bir obje aratıldığında bile yakın özellikte olan bir anahtar kare bulunmadığından erişim sağlanamayacaktır. Eğer daha kesin neticelere ulaşmak amacıyla bölümde bulunan kareler kullanılarak yeni bir anahtar kare oluşturulacaksa

hareket eden bölgelerin hareket süreçlerinin orta noktası baz alınarak anahtar kare oluşturulmalıdır. Bölüm içinde hareket eden bölgeler için daha da kesin neticeler için hareket orta noktaları görsel anahattı oluşturmalı, hareketin diğer kısımları düşük ağırlıklarla anahtar kareye kopyalanmalıdır. Anahtar kare hazırlama işlemi indeks içinde arama yapılacak eşleştirme yönteminin hassasiyetleri dikkate alınarak yapılmalıdır. Bir video bölümü için hazırlanmış örnek bir anahtar kare Şekil 3.7' de görülmektedir.

BÖLÜM 4. VIDEO BÖLÜMLENDİRME ALGORİTMALARI

Sıkıştırılmamış videolarda çalışan bölümlendirme algoritmaları genel olarak dayandıkları tekniğe göre ardışıl kareler arasında bir benzerlik oranı elde ederler. Arka arkaya gelen iki karenin birbirine benzerliği düşükse orada bir bölüm ayrımı olduğunu kabul ederek videoları bölümlendirme yoluna gitmektedirler. Bu tarz tekniklerde ardışıl kare benzerlikleri için hangi değerlerin üzerinin bir bölüm geçişi olarak algılanacağı bir eşik değeri ile belirlenmektedir. Bu yüzden bu tarz algoritmalara literatürde eşik tabanlı yaklaşım algoritmaları da denmektedir[5].

4.1. Piksel Karşılaştırması

Piksel karşılaştırma yöntemi arka arkaya gelen karelerde örtüşen piksellerin parlaklık ve renk değerlerini karşılaştırarak bölüm geçişi algılama işlemi yapmaktadır. Bu yöntem şablon eşleştirme olarak da adlandırılmaktadır.

Bölüm geçişlerini algılamanın en standart yolu ardışıl karelerde örtüşen piksel farklılıklarının toplamını hesaplamak ve onu bir eşik değeri ile karşılaştırmaktır.

$$D_{(i,i+1)} = \frac{\sum_{x=1}^X \sum_{y=1}^Y |P_i(x,y) - P_{i+1}(x,y)|}{X.Y} \quad (\text{siyah beyaz videolar})$$
$$D_{(i,i+1)} = \frac{\sum_{x=1}^X \sum_{y=1}^Y \sum_c |P_i(x,y,c) - P_{i+1}(x,y,c)|}{X.Y} \quad (\text{renkli videolar}) \quad (1)$$

Denklem (2.1) ve denklem (2.2) de i ve $i+1$ $X \cdot Y$ boyutunda iki ardışıl kareyi temsil etmektedir. Siyah beyaz videolar için $P_i(x,y)$ i karesinde (x,y) koordinatına sahip pikselin parlaklık değerini, renkli videolar için c renk bileşenlerini ifade etmektedir (R,G,B renk sistemi için $c \in \{R,G,B\}$), $P_i(x,y,c)$ i karesinde (x,y) koordinatına

sahip pikselin c ile ifade edilen renk bileşenin renk değerini ifade etmektedir. Denklemlerde farkların mutlak değer ifadesi içine alınmasının amacı ise piksellerde yer alan pozitif ve negatif değişimlerin birbirine eklenerek toplam değişimin daha kesin bir ifade ile elde edilebilmesidir.

Mutlak değer ifadesi kullanılmadığı örnek bir video $i+1$. karede i . kareye göre piksellerin yarısının değerleri artmış yarısının değerleri azalmış ise bu iki ardışıl kare arasındaki toplam piksel değişim değeri 0' a yakın çıkacak ve bu iki kare birbirine benzer olarak algılanacaktır. Oysa ki bu iki ardışıl karede bütün pikseller değiştiği için büyük bir farklılık vardır.

Formül sonucunda elde edilen kare benzerlik değeri önceden belirlenmiş bir eşik değerinden büyük olduğu durumda bir bölüm geçişi algılanacak bir önceki bölüm geçişinden bu yana geçen bütün kareler yeni algılanan bölüme dahil edilecektir. Bu metodun en önemli eksiği küçük bir alandaki büyük bir renk değişimi ile büyük bir alanda küçük renk değişimlerinin birbirinden ayıramamasıdır. Örneğin küçük bir alandaki piksel değerleri hızlı büyük bir değişime gittiğinde yanlış bölüm algılanacaktır. Bu yüzden piksel karşılaştırma tabanlı yöntemler obje ve kamera hareketlerine çok duyarlıdır.

Bir eşik değerinin üzerinde değişen piksellerin sayısını hesaplamak ve değişen piksel sayısının 2. eşik değerinin üzerinde olduğu durumda bölüm geçişi algılamak uygun bir yaklaşımdır.

$$DP(i, i + 1, x, y) = \begin{cases} 1 & |P_i(x, y) - P_{i+1}(x, y)| > T_1 \\ 0 & |P_i(x, y) - P_{i+1}(x, y)| \leq T_1 \end{cases}$$

$$D(i, i + 1) = \frac{\sum_{x=1}^X \sum_{y=1}^Y DP(i, i + 1, x, y)}{X \cdot Y} \quad (2)$$

Değişen piksel sayısı $D(i, i + 1)$ T_2 eşik değerinden büyük ise bölüm geçişi algılanır.

Bu yaklaşımda ise küçük kare farklılıkları duyarlılık dışında bırakılmıştır. Bu yöntem hala kamera ve obje hareketine duyarlıdır. Örneğin kameranın küçük bir hareketiyle çok fazla piksel değerinin değiştiğini algılayarak burada bir bölüm geçişi

algılanabilir. Oysa ki böyle bir durumda oluşan sadece görüntünün birkaç piksel hareketidir sahne içeriğinde hiçbir değişiklik olmamıştır. Ayrıca bu yöntem parlaklık değişimine en duyarlı yöntemdir. Kameradan veya sahneye gelen ışık miktarından kaynaklanan bir parlaklık değişimi olduğunda bütün piksel değerleri değişeceğinden bölüm geçişi algılanır fakat böyle bir örnekte sahne içeriği değişmemiştir ve yanlış bir algılama yapılmıştır.

4.2. Blok Tabanlı Piksel Karşılaştırması

Şablon eşleştirme için bütün resim ile ilgilenmek yerine blok tabanlı yaklaşımlar kamera ve obje hareketlerine duyarlılığı azaltmak için videoları bloklara bölerek yerel bileşenlerle ilgilenirler. Bu yöntemde her video karesi b adet bloğa bölünür ve bir sonraki karede eşleşen kendisine eşleşen blok ile karşılaştırılır.

$$D(i, i + 1) = \sum_{k=1}^b c_k \cdot DP(i, i + 1, k) \quad (3)$$

Genelde i ve $i+1$ kareler arasında benzerlik, denklem 2.3 de olduğu gibi ölçülmektedir. Burada c_k k bloğu için önceden belirlenmiş bir katsayıdır ve $DP(i, i + 1, k)$ i ve $i+1$ karelerinde k . bloklar için eşleşme değeridir.

KASTURİ R., JAİN R., tarafından örtüşen bloklar, benzerlik oranı ile karşılaştırılmıştır[6] :

$$\lambda_k = \frac{\left[\frac{\sigma_{k,i} + \sigma_{k,i+1}}{2} + \left(\frac{\mu_{k,i} + \mu_{k,i+1}}{2} \right)^2 \right]^2}{\sigma_{k,i} \cdot \sigma_{k,i+1}} \quad (4)$$

Burada $\mu_{k,i} + \mu_{k,i+1}$ değerleri ardışıl i ve $i+1$ karelerinde k . blok için ortalama parlaklık değerleridir ve $\sigma_{k,i} \cdot \sigma_{k,i+1}$ denklem 2.3 ile verilen bloklara ait değişim değerleridir.

$$DP(i, i + 1, k) = \begin{cases} 1 & \lambda_k > T_1 \\ 0 & \lambda_k \leq T_1 \end{cases} \quad (5)$$

Benzerlik oranları t_1 eşik değerinden fazla blokların değerleri sayılarak bunların yeteri kadar büyük olduğu değerler bölüm geçişi olarak algılanmaktadır.

Şablon eşleştirme ile karşılaştırıldığında bu yöntem küçük hareketlere daha az duyarlıdır fakat karmaşık formüllerinden dolayı hesaplama zamanı çok yüksektir. Ek bir dezavantaj olarak, farklı oldukları halde aynı benzerlik oranına sahip bloklar arasında bir geçiş algılanmasıdır[5].

Blok Tabanlı Pksel Karşılaştırma yöntemlerinin tamamı piksel karşılaştırma yöntemine göre daha az kamera hareketine duyarlıdır fakat parlaklık değişim duyarlılıkları piksel karşılaştırma yönteminde olduğu gibi yüksektir.

4.3. Histogram Tabanlı Yöntemler

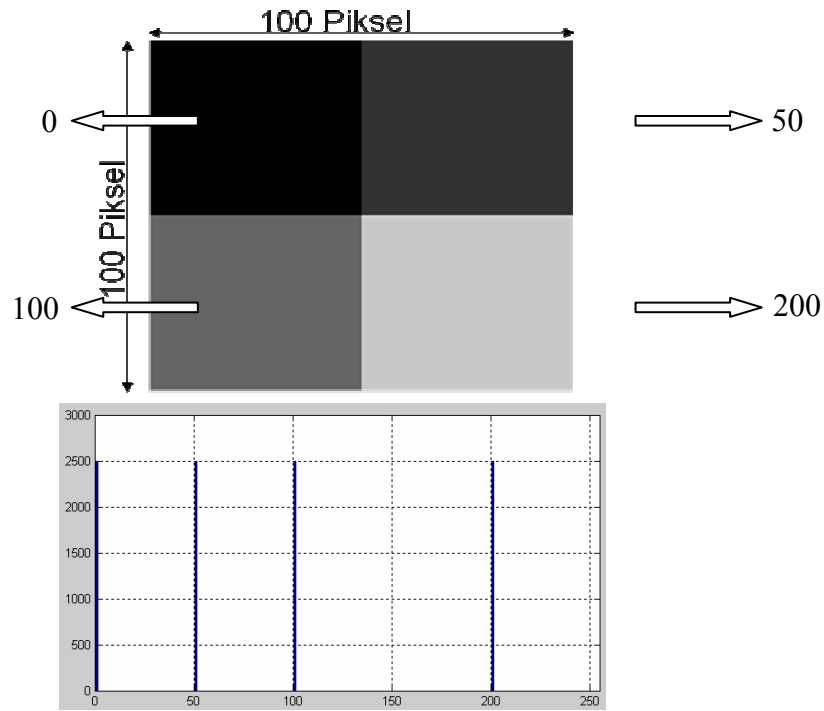
Video bölümlendirme algoritmalarında kamera obje hareketlerine duyarlılığı düşürmek için histogram tabanlı karşılaştırma yöntemleri ortaya atılmıştır. Histogram karşılaştırmasına dayanan yöntemlerin altında yatan ana fikir değişmeyen arka plana ve değişmeyen objelere (hareketli veya hareketsiz) sahip ardışıl iki karenin histogramlarında çok fazla bir fark oluşmayacağıdır. [5] Bununla birlikte histogram resim döndürülme ve resimleme açısının değişmesine duyarsızdır. Bu temellere dayanarak literatürde ardışıl karelerin histogramlarının karşılaştırılmasına dayanan bir çok çalışma yapılmıştır.

Histogram bir sinyal içeriğinde bulunan frekans bileşenleri hakkında bilgi veren bir grafikdir. Analog bir işaret için histogram analog işaretin hangi frekans değerinde bileşenleri olduğunu her bir frekans değeri için kaç adet bileşen olduğu bilgisini verir. Analog işaretler için işaretin histogramı sayesinde işaret bileşenlerinin hangi frekanslarda yoğunlaştığı bilgisine ulaşılır.

Sayısal resimlerde ise histogram grafikleri resmin renk bileşenleri hakkında bilgi verir. Sayısal videolarda histogramlar için frekansların yerini renk koyuluk değerleri almaktadır. Resim karesi için 0-255 arası renk ağırlık değerlerine ait kaç tane piksel değeri olduğu konusunda bilgi sunmaktadır. Örneğin bir resme ait histogram grafiği 255 değerine yakın bir bölgede yoğunlaşıyorsa o resmin parlak bir resim olduğu neredeyse beyaz bir resim olduğu yorumu yapılabilir. Eğer histogram 0 değerine yakın bölgede yoğunlaşıyorsa resmin karanlık bir resim olduğu neredeyse siyah renk bir resim olduğu yorumu yapılabilir. Histogram grafiği farklı koyuluk değerlerine dağılmışsa resmin çok renk içeren dolayısıyla farklı objeler içeren bir resim olduğu yorumu yapılabilir.

Şekil 4.1'de 100×100 piksel boyutlarına sahip örnek bir siyah-beyaz resim ve resme ait histogram grafiği görülmektedir. Şekildeki resimde toplam 10000 adet piksel vardır ve bunların $1/4$ 'ü 0 koyuluk değerine, $1/4$ 'ü 50 koyuluk değerine, $1/4$ 'ü 100 koyuluk değerine, $1/4$ 'ü 200 koyuluk değerine sahiptir yani resimde 0,50,100,200 koyuluk değerlerine ait 2500'er adet piksel yer almaktadır. Histogram grafiği incelendiğinde bahsedilen koyuluk değerlerinde bulunan piksel sayılarının histogram grafiğinde yer aldığı görülmektedir.

Renkli resimlere ait histogram grafikleri her bir renk değeri için elde edilmeli ve beraber değerlendirilmelidir. Sayısal resimlerde R,G,B, bileşenleri yer aldığından üç adet ayrı histogram grafiği (kırmızı, yeşil ve mavi histogramları) elde edilmelidir.



Şekil 4.1. 100×100 piksel boyutuna sahip örnek bir siyah-beyaz bir resim ve resme ait gri koyuluk değer histogramı

4.3.1. Genel histogram karşılaştırması

Histogram karşılaştırma yönteminde piksel tabanlı yöntemlerde olduğu gibi ardışıl kareler arası benzerlik değerleri ortaya çıkarılmaktadır ve benzerlik değerinin önceden belirlenen bir eşik değeri üzerinde olan kareler arasında bir bölüm geçişi algılama işlemi yapılmaktadır. Denklem 1'de yer alan yaklaşımın histogramlara uyarlanarak parlaklık değerleri yerine siyah beyaz videoların grilik değer histogramlarının karşılaştırılması üzerine çalışmalar yapılmıştır[7,8,9]. Yapılan bu çalışmalarda kullanılan genel formül denklem 6 da yer almaktadır.

$$D(i, i + 1) = \sum_{j=1}^m |H_i(j) - H_{i+1}(j)| \quad (6)$$

Denklem 6’da yer alan $H_i(j)$ i . karenin j koyuluk değerine ait histogram bileşenini ifade etmektedir, j koyuluk değerini, n ise videonun içerdiği en üst koyuluk değerini ifade etmektedir.

Renkli videolar üzerine yapılan ve başarılı sonuçlar elde edilen bir çalışma Zhang, Kankanhallı ve Smoliar tarafından sunulmuştur[5].

$$D(i, i + 1, c) = \sum_{j=1}^n |H_{(i,c)}(j) - H_{(i+1,c)}(j)| \quad (7)$$

Renkli videolar için gerçekleştirilen bu yaklaşım siyah beyaz videolar için kullanılan denklem 6’nın uyarlanması ile elde edilmiştir. Denklem 7’de c renk bileşenlerini (R,G,B renk sistemi için $c \in \{R, G, B\}$) ifade etmektedir. Yöntem sonucunda her bir kare için 3 adet histogram değeri elde edilir. Kareler arası benzerlikleri ifade etmek için ardışıl karelerin her bir renk bileşenleri için histogram farkları hesaplanır ve bunların ortalaması alınarak her kare için tek bir benzerlik değeri elde edilir elde edilen bu benzerlik değeri eşik değerinin üzerinde ise bölüm geçişi algılanır.

Şekil 4.2a’da görüldüğü gibi ardışıl iki kareye ait histogram grafikleri farklılıklar göstermektedir. Ardışıl karelere ait 3 renk bileşenin histogramları karşılaştırılarak kare benzerlik değeri elde edilir. Şekil 4.2b’de denklem 7 toplam formülü içinde ki formülle elde edilen r,g,b histogram farkları görülmektedir.



Şekil 4.2. a. Örnek bir videodan alınmış ardışıl iki kare bu karelere ait Kırmızı, Yeşil, Mavi Renklere ait Histogram grafikleri - b. Karelerin histogram farkları

Bölüm geçişi olduğu durumda ardışıl kareler arası fark değerini arttırmak için birçok yazar χ^2 testini kullanmayı uygun görmüşlerdir[7].

$$D(i, i + 1, c) = \sum_{j=1}^n \frac{|H_{(i,c)}(j) - H_{(i+1,c)}(j)|^2}{H_{(i+1,c)}(j)} \quad (8)$$

Fakat Zhang, Kankanhallı ve Smoliar tarafından yapılan çalışmada [9] χ^2 testinin sadece bölüm geçişi esnasındaki fark değerini yükseltmekle kalmayıp obje ve kamera hareketi olduğu anlarda da fark değerini yükselttiğini sunulmuştur.

Histogram tabanlı karşılaştırma yönteminde kamera ve obje hareketine duyarlılık piksel tabanlı yöntemlere göre daha düşüktür. Bu iyileştirme histogramın temel özelliğinden kaynaklanmaktadır. Genel histogram tabanlı yöntemde parlaklık değişimine duyarlılık yüksektir. Çünkü ardışıl kareler arasında küçük bir parlaklık değişimi resim histogramını bulunduğu yerden sağa veya sola taşımaktadır, bu durumda iki karenin fark değeri büyük çıkmaktadır. Bazı durumlarda bu fark değeri gerçek bölüm geçişinin olduğu karelerden daha yüksek çıkmaktadır, bu yüzden parlaklık değişimi olan bir videoda genel histogram karşılaştırma yöntemi yanlış bölümler algılamaktadır.

4.3.2. HSV ve YIQ histogram karşılaştırması

HSV ve YIQ, RGB' den farklı birer renk kodlama tekniğidir. HSV (Hue, Saturation, Value) kodlama tekniği RGB kodlama tekniğinin bir nonlinear transformudur. HSV kodlama da H ve S bileşenleri renk bilgisini ifade etmektedir, V bileşeni ise parlaklık değerini ifade etmektedir. HSV kodlama tekniği HSB (Hue, Saturation, Bright) olarak da bilinmektedir. YIQ kodlama tekniği NTSC televizyon görüntüsü kodlama için kullanılmaktadır ve HSV tekniğine benzer olarak Y bileşeni parlaklığı ifade etmektedir, I ve Q bileşenleri ise renk bilgisini ifade etmektedir.

HSV ve YIQ histogram karşılaştırma yöntemlerinde parlaklık bileşenleri karşılaştırma dışında tutulmaktadır. Bu yaklaşım parlaklık değişimine duyarlılığı azaltmaktadır. Bu iyileştirme HSV ve YIQ kodlama tekniklerinin sağladığı bir özelliktir çünkü bu iki kodlama tekniğinde de parlaklık bileşeni renk bileşenlerden ayrı olarak ifade edilmektedir. Bu teknikleri kullanan basit yaklaşımda denklem 7'de gösterilen formül küçük bir farkla kullanılmaktadır. HSV kodlama tekniği için $c \in \{H, S\}$, YIQ kodlama tekniği için $c \in \{I, Q\}$ alınmaktadır.

$$D(l, l + 1) = 1 - \frac{\sum_{j=1}^n \min (H_i(j) - H_{i+1}(j))}{\sum_{j=1}^n \max (H_i(j) - H_{i+1}(j))} \quad (9)$$

Denklem 9 ile verilen formül HSV ve YIQ histogram karşılaştırma tekniklerinde kullanılan diğer bir formüldür. Denklem 9 ile verilen formül farklı histogramlara ait iki kareye uygulandığında sonuç 1 e yakın çıkmaktadır. Bu formül kullanıldığında 1'e yakın geçiş değerleri bir bölüm geçişi olarak algılanmaktadır.

$$D(l, l + 1) = \sum_{j=1}^n \sum_{k \in N(k)} W(k) \cdot (H_i(j) - H_i(k)) \quad (10)$$

HSV ve YIQ histogram karşılaştırma teknikleri için 1 boyutlu veya 2 boyutlu histogram farkları almak yerine denklem 10'da yer alan formül önerilmektedir. Denklem 10'da $N(k)$ histogram j elemanına yakın elemanlardandır ve j elemanın komşuluğu olarak ifade edilir ve $W(k)$ ilgili komşuya ait ağırlık değeridir. Bu yaklaşımda genellikle 3×3 veya 3 komşuluk ilişkisi kullanılmaktadır.

HSV ve YIQ'dan farklı olarak $L^*u^*v^*$, $L^*a^*b^*$ ve Munsell uzayı renk kodlama teknikleri ile yapılmış histogram karşılaştırma teknikleri sunulmuştur [10]. Farklı kodlama teknikleri üzerine çalışan bu yöntemler R,G,B, kodlama tekniğine göre daha başarılı sonuçlar vermektedirler. Özellikle parlaklık değişimine duyarlılıkları oldukça azdır. Fakat dijital videolar R,G,B, kodlama tekniği ile kodlanmaktadır. Farklı bir kodlama tekniği üzerinde çalışılması gerektiğinde R,G,B, kodlama tekniğinden diğer kodlama tekniklerine geçmek fazladan büyük bir işlem yükünü beraberinde getirmektedir.

4.3.3. Blok tabanlı histogram karşılaştırması

Histogram tabanlı tekniklerin basit, obje ve kamera hareketlerine duyarlılığı az olan teknikler olduklarından yukarıda bahsedilmektedir. Fakat bu teknikler uzamsal bilgiyi önemsemediklerinden içerikleri farklı ama birbirlerine yakın histogramlara sahip iki kare arasında karşılaştırma yaptıklarında yanılmaktadırlar. Blok tabanlı

piksel karşılaştırma tekniği uzamsal bilgiyi dikkate almaktadır. Histogram tabanlı tekniklerin ve blok tabanlı tekniğin avantajlarını bir araya getirmek için blok tabanlı histogram teknikleri öne sürülmüştür. Bu iki avantajın bir araya getirilmesi ile kamera ve obje hareketinden kaynaklanan yanlış algılamalar ortadan kaldırılacak ve uzamsal bilgide dikkate alınarak daha kesin bölüm algılamaları yapılacaktır.

$$DP(i, i + 1, c, k) = \sum_{j=1}^n |H_{(i,c)}(j, k) - H_{(i+1,c)}(j, k)| \quad (11)$$

$$D(i, i + 1) = \sum_{k=1}^b DP(i, i + 1, k) \quad (12)$$

Ardışıl i ve $i+1$ karelerinin fark değeri denklem 11 ve denklem 12 kullanılarak elde edilir. Burada $H_{(i,c)}(j, k)$ i . kareye ait k adresine sahip bloğa ait c renk bileşen histogramını, b ise toplam blok sayısını ifade etmektedir.

Swanberg ve arkadaşları tarafından blok tabanlı histogram karşılaştırması tekniğinin kullanıldığı çalışmada blokların R,G,B histogram değerlerini karşılaştırmak için aşağıdaki eşitlik kullanılmıştır [11] :

$$DP(i, i + 1, k) = \sum_{c \in \{R, G, B\}} \sum_{l=1}^n \frac{(H_i^c(l) - H_{i+1}^c(l))^2}{H_i^c(l) + H_{i+1}^c(l)} \quad (13)$$

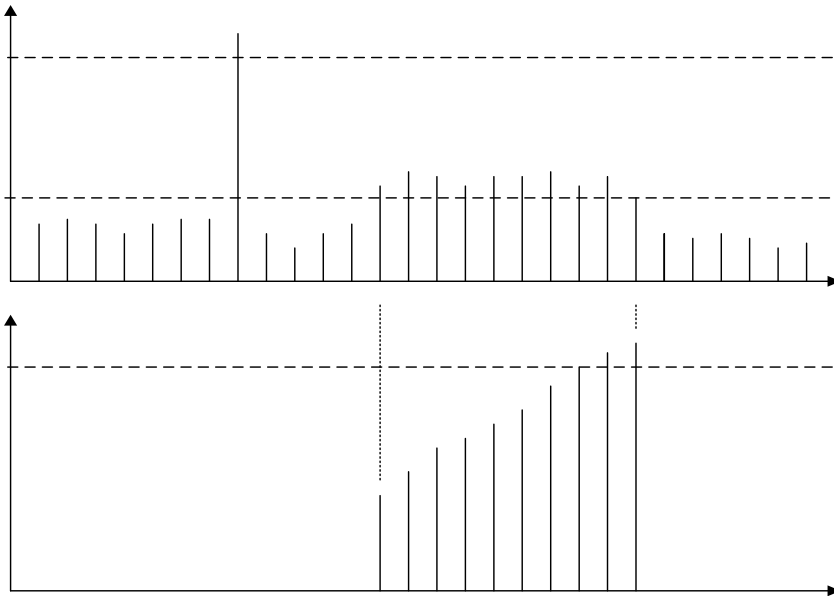
Denklem 13 ile elde edilen sonuçlar c_k değeri bütün bloklar için $1/b$ seçilerek denklem 3 de kullanılarak kareler arası farklılık değeri elde edilmektedir.

4.4. Kademeli Bölüm Geçişleri İçin İkiz Karşılaştırma Tekniği

Buraya kadar anlatılan teknikler bölüm geçişi olduğu ana ait karelerin histogram farklarının çok büyük olduğunu ve bölüm geçişlerin bir eşik değeri ile karşılaştırılarak algılanabildiği fikrine dayanmaktadırlar. Bu yüzden birçok tek eşik bağımlı teknik kademeli bölüm geçişlerini algılama için başarısız kalmaktadır.

Kademeli geçişin bulunduğu bir bölüm içinde kareler arası farklar normal bölüm içindeki kareler arası farklardan yüksektir. Fakat kademeli geçiş sırasındaki farklar bölüm geçişi için belirlenen farktan düşüktür. Bölüm geçişini belirlemek için kullanılan eşik değerinin düşürülmesi hassasiyeti çok arttıracak ve video içindeki kamera hareketinden ve parlaklık değişiminden kaynaklanan küçük değişimleri bir bölüm olarak algılayacaktır bu yüzden video gerçekte olduğundan daha fazla bölüme ayrılacaktır. Bu yüzden eşik değerini düşürmek uygun olmayacaktır. İkiz karşılaştırma (twin-comparison) [9] metodu iki eşik değeri kullanarak kademeli geçişe ait birikmiş kare farklarını dikkate alarak bölüm geçişi algılamaktadır.

İkiz karşılaştırma tekniğinde bölüm geçişlerini algılamak için iki defa tarama yapılır yapılan ilk taramada T_h eşik değerinin üzerinde kalan kare farkları bölüm geçişi olarak algılanır. İkinci taramada esnasında T_l eşik değeri kademeli geçişin potansiyel başlangıç karesi F_s yi algılamak için kullanılır. İkinci taramada belirlenen F_s ve sonrası kareler ait değerler eklenerek tarama devam ettirilir eğer fark değerleri T_l altına inene kadar toplam fark değeri T_h değerinin üzerine çıkarsa kademeli geçiş algılanmış olur ve bölüm oluşturulmaktadır. Eğer F_s karesi ile başlayarak eklenen fark değeri T_h değerine ulaşmadan kare farkları T_l altına inerse burada bir bölüm geçişi olmadığı algılanır ve bir sonraki F_s ve kademeli bölüm taraması için devam edilir.



Şekil 4.3. İkiz Karşılaştırma tekniği tarafından kademeli geçişlerin algılanması

Bazı durumlarda kademeli bölüm geçişleri sırasında ardışıl kare farklarının T_l değerinin altına düştüğü durumlar ile karşılaşılabilir. Bu tarz senaryolarda yanlış algılamayı engellemek için tarama işlemine önceden belirlenen bir tolerans değeri ilave etmek faydalı olmaktadır.

Yukarıda anlatılanlardan da anlaşıldığı gibi ikiz karşılaştırma tekniği basit mantıkla işleyen ve başarılı bir tekniktir. İkiz karşılaştırma yöntemi hem kademeli hem de ani geçişleri algılayabilmektedir. Boreczky ve Rowe birçok bölümlendirme algoritmasının karşılaştırmasını yaptıkları çalışmalarında ikiz karşılaştırma tekniğinin basit ve çok doğru çalışan bir teknik olduğu sonucuna varmışlardır[12].

BÖLÜM 5. FİLTRELENMİŞ VIDEO HİSTOGRAM KARŞILAŞTIRMASI

5.1. Giriş

Sayısal video veritabanlarına hızlı erişim gereksinimlerini karşılamak için video indekslemenin kullanılmasının yararları açıktır. Bir video için çıkarılan indeks ne kadar doğru ve öz bilgiler içeriyorsa videoya erişim o denli hızlı ve etkili olacaktır.

Videolar için mümkün olan en yüksek başarıma sahip indeksi elde etmek doğru ve hızlı video bölümlendirmeden geçmektedir. İşlenen videoda bölümlerin doğru algılanamadığı durumda elde edilen indeksin kullanılması aramalar neticesinde yanlış erişim yapılmasına hatta bazı durumlarda bölümün tamamına erişilememesi ile neticelenecektir. Bu sebeple video indeksleme üzerine yapılan çalışmaların büyük çoğunluğu video indeksleme üzerine yoğunlaşmıştır. Yapılan çalışmaların sonucunda birçok algoritma geliştirilmiştir fakat halen bütün video türlerinde kullanılacak bir algoritma sağlanamamıştır.

Bir önceki bölümde anlatılan sıkıştırılmamış video öğeleri ile bölümlendirme yapan algoritmaların eksiklikleri olduğu ve istenen performansı sağlayamadığı açıktır. Bu algoritmaların en büyük eksikliği parlaklık değişimi ve kamera hareketlerine gösterdikleri yüksek duyarlılıktır. Bu duyarlılıklarından ötürü bu yöntemler görsel içeriğin değişmediği parlaklık değişimi ve kamera hareketlerini fazladan bölüm geçişi olarak algılamaktadır. Bu durumda video için bulunan bölüm sayısı gerçek bölüm sayısından daha fazla olmakta ve başarılı bir indeksleme yapılamamaktadır.

Önerilen bölümlendirme yöntemi sıkıştırılmamış video bölümlendirme algoritmalarının yukarıda bahsedilen eksikliklerini gidermek amacı ile geliştirilmiştir. Bir önceki

bölümde de bahsedildiği gibi sıkıştırılmamış video bölümlendirme algoritmalarının hemen hepsi parlaklık değişimine çok duyarlıdır. Önerilen yöntem parlaklık değişimi ve kamera hareketlerine duyarlılığı düşürebilmek için bölüm geçişlerinin sahnede bulunan objelerin değişmesi ile gerçekleştiği ve renk bileşenleri yerine objeler ile ilgilenmenin başarılı olacağı fikri üzerine geliştirilmiştir.

Sahnede bulunan objeleri ayırtmak ve değişimlerini izlemek için yapay zeka, yapay sinir ağları gibi akıllı tekniklerin kullanılması doğru sonuçlar verecektir. Fakat bu gibi teknikleri bütün videoya uygulamak çok ağır bir işlem yükünü beraberinde getirmektedir. Ayrıca bu tekniklerin eğitim aşamalarında birçok tür videoyu kapsayacak örneklerle eğitilmesi gerekmektedir. Sınırlı örnek ile eğitim yapıldığı durumlarda algoritmanın her tür video için kullanılması doğru sonuçlar vermeyecektir.

5.2. Filtrelenmiş Video Histogram Karşılaştırması

Filtrelenmiş Video Histogram Karşılaştırması yöntemi bölümlendirme işlemini diğer çoğu sıkıştırılmamış video bölümlendirme algoritması gibi histogram karşılaştırma tekniğini kullanmaktadır. Obje değişimine dayalı video bölümlendirme işleminin hızlı ve kullanışlı bir yapıda kullanılabilmesi için resim filtreleme tekniklerinden faydalanılmıştır. Bölümlendirilecek videonun tamamı öncelikle resim filtreleme teknikleri ile filtrelendikten sonra histogram karşılaştırmasına tabi tutulmuştur.

Sık kullanılan resim filtreleri arasında yer alan sobel resim filtresi filtrelenen resim içinde renk değişimlerinin keskin olduğu bölgeleri, değerlerini siyah veya beyaz (filtre ağırlıklarına göre belirlenmektedir) renklerine ve renk değişiminin bulunmadığı düz alanları diğer renge yakınlaştırarak ön plana çıkarmaktadır. Daha kısa bir tabirle sobel filtresi resimler içindeki obje kenarlarını belirginleştirmektedir.



Şekil 5.1. a. örnek bir video karesi - b.video karesinin sobel filtresi ile filtrelenmesi -c.video karesinin farklı ağırlıklara sahip sobel filtresi ile filtrelenmesi

Şekil 5.1’de görüldüğü gibi sobel filtresi ile filtrelenen resimde obje kenarları belirginleşmektedir. Filtre ağırlıklarının değiştirilmesi ile sobel filtresinin hassasiyeti arttırılabilmektedir. Şekil 5.1b’de gösterilen filtreleme işleminde kullanılan filtre

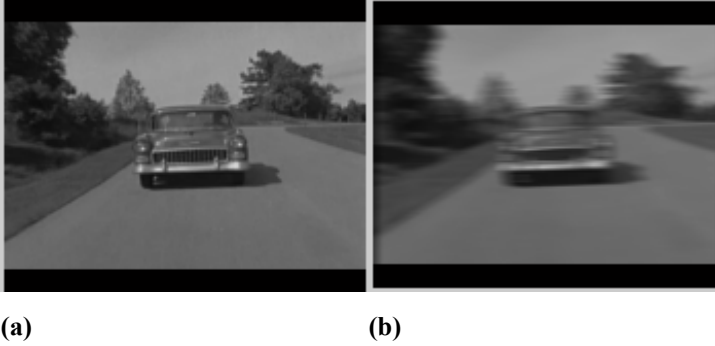
$$\begin{array}{ccc} 1 & 2 & 1 \\ 1 & 0 & -0.5 \\ -0.5 & -1 & -0.5 \end{array} \text{ ağırlıklarına sahiptir. Örneğin ağırlık değerlerinin } \begin{array}{ccc} 1 & 2 & 1 \\ 1 & 0 & -1 \\ -1 & 2 & -1 \end{array}$$

şeklinde değiştirilmesi ile Şekil 5.1c’de görülen obje kenarları beyaz, düz alanların ise siyah renk olduğu sonuç elde edilecektir.

Önerilen yöntemde videonun tek başına sobel filtresi ile filtrelenmesi istenilen performansı elde etmek mümkün olmayacaktır. Histogram karşılaştırması yapıldığında ardışıl kareler için obje kenarlarının ve düz alanların kaç piksel kapladıkları karşılaştırılacaktır. Bu şekilde yapılacak karşılaştırmada benzer objelere sahip bölüm geçişinin bulunduğu ardışıl karelerde fark değeri düşük çıkacak ve bölüm geçişleri algılanamayacaktır. Ayrıca kenarlar çok keskin bir şekilde ortaya çıktığından objenin küçük hareketleri ile obje arkasına gelen arka rengin değişmesi gibi sebeplerle obje kenarlarında büyük farklılıklar ortaya çıkacaktır. Bu öngörünün doğruluğunu test etmek için yapılan denemeler sonucunda aynı sonuca ulaşılmıştır.

Yukarıda bahsedilen sobel filtresinin tek başına kullanıldığında oluşacak olumsuzlukları gidermek amacıyla sobel filtreleme işleminden önce obje kenarlarının daha yumuşatılarak geniş alana yayılması ve yakın piksellerin belli ağırlıklarla birbirleri üzerine kopyalanması ön görülmüştür. Bu sayede sobel filtresi ile elde edilen keskin kenar değerleri yumuşatılacak ve kenarlara yakın alanların (arkaplan ve

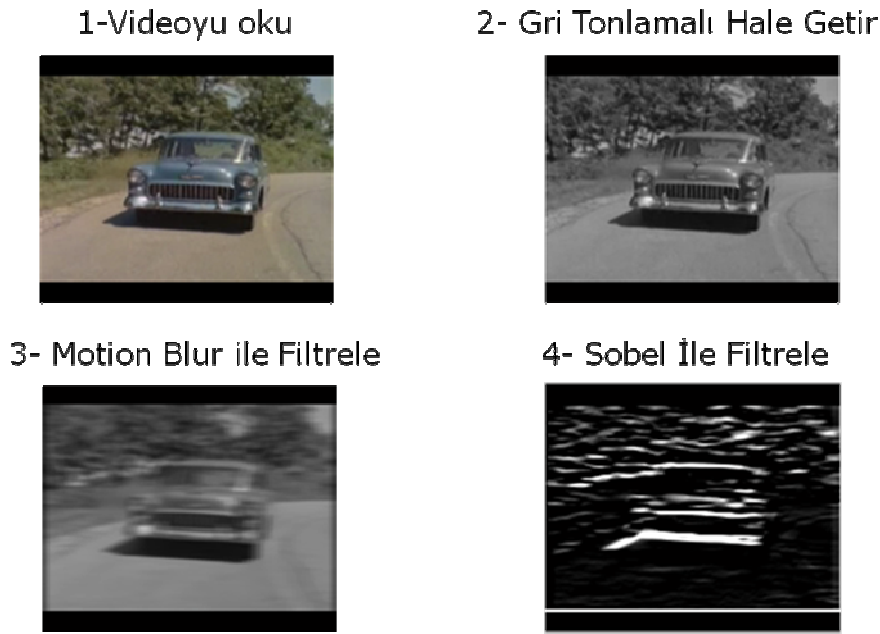
obje kenar bölgelerinin) etkileri de incelenmiş olacaktır. Bu gereksinimleri karşılamak amacı ile motion blur filtresinden faydalanılmıştır.



Şekil 5.2. a. Örnek bir video karesi - b. Video karesinin motion blur filtresi ile filtrelenmesi

Motion Blur filtresi resim içinde komşu piksel değerlerini birbirlerine kopyalayarak görüntünün bulanıklaştırılması için kullanılmaktadır. Şekil 5.2’de görüldüğü gibi Motion Blur filtresinin uygulanması ile resim bulanıklaştırılmaktadır. Şekil 5.2’de kullanılan filtre ağırlıkları $[0.05 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.05]$ şeklindedir bu filtre ağırlıklarının değiştirilmesi ile resim daha bulanık hale getirilebilir başka bir deyişle sonuç için kaç adet komşu piksel kullanılacağı belirlenebilmektedir.

Önerilen yöntemde yukarıda anlatılan öngörülerin ışığında videonun filtrelenmesinde Motion Blur ve Sobel filtreleri kullanılmıştır. Yöntemin obje kenarlarının değişimini takip etmesi sebebiyle videolar siyah beyaz videoya çevrilerek işlenmiştir. Filtrelenmiş Video Histogram Karşılaştırması yönteminin filtreleme safhasının adımları şekil 5.3’de görülmektedir.



Şekil 5.3. Filtrelenmiş video histogram karşılaştırması filtreleme adımları

Filtrelenmiş Video Histogram Karşılaştırması yönteminde seçilen sobel filtresi ağırlıkları düz alanları siyah renge yaklaştıracak, obje kenarlarını ise beyaz renge

yaklaştıracak şekilde $\begin{matrix} 2 & 3 & 2 \\ 1 & 0 & -1 \\ -2 & -3 & -2 \end{matrix}$ olarak seçilmiştir. Elde edilen filtrelenmiş

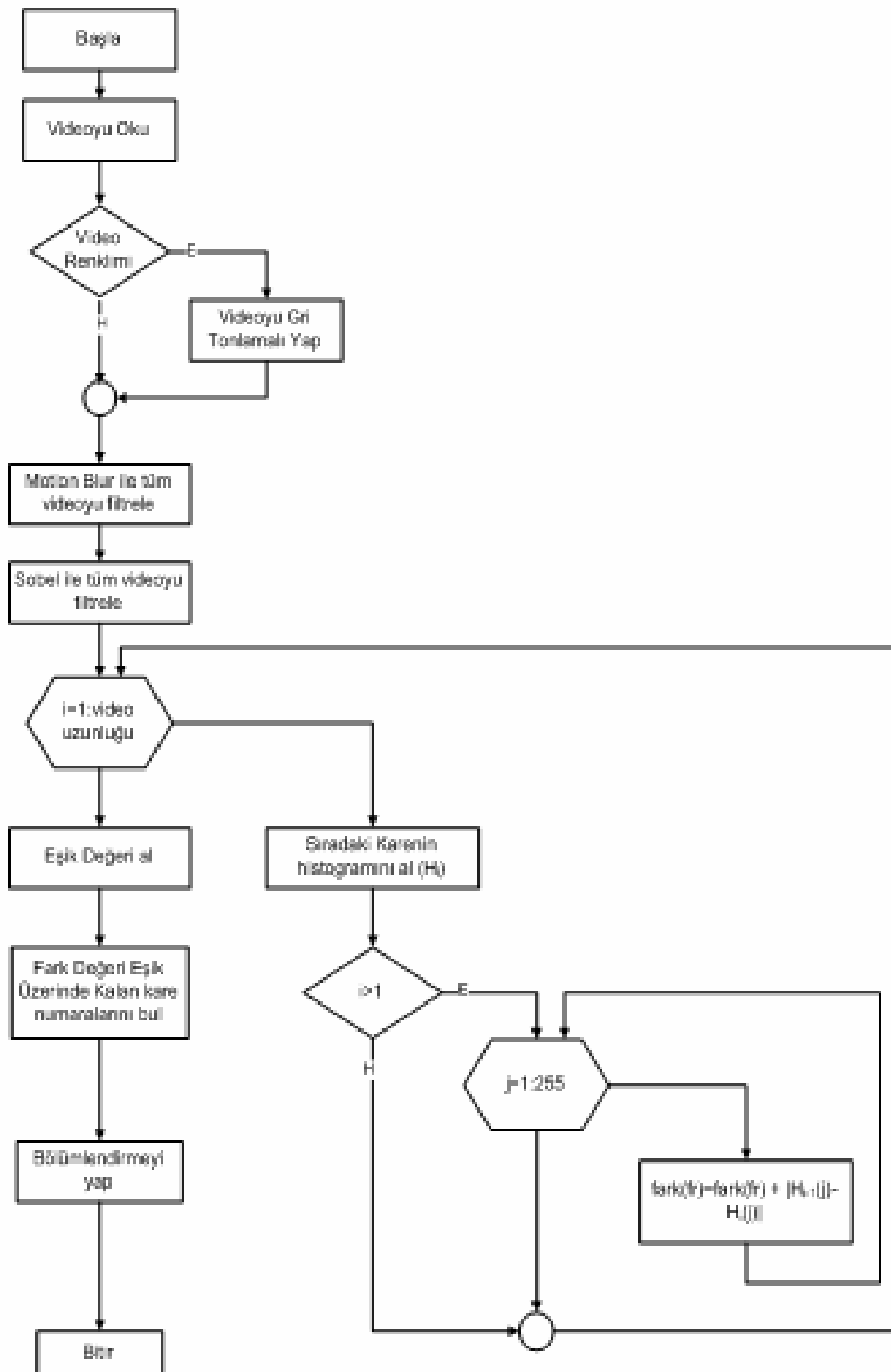
ardışıl karelerin fark değerlerini elde etmede standart histogram karşılaştırma formüllerinin kullanılması durumunda yöntemin ilgilenmediği düz alanlarda karşılaştırma kriterine dahil edilerek yanlış sonuçların artmasına sebep verecektir. Bu eksikliği gidermek amacı ile standart histogram karşılaştırma formülünde siyah rengi temsil eden 0 değeri devre dışı bırakılarak Filtrelenmiş Video Histogram Karşılaştırması yönteminde kullanılacak formül (Denklem 14) elde edilmiştir. Denklem 14'te verilen formül, sayısal videolarda 8 bit renk derinliğine sahip (koyuluk değerleri 0-255 ; $n=256$) videolarda kullanılacak formüldür.

$$D(i, i + 1) = \sum_{j=1}^{255} |H_i(j) - H_{i+1}(j)| \quad (14)$$

Filtrelenmiş Video Histogram Karşılaştırması yöntemi çalışma adımları sırasıyla özetlenecek olursa :

1. Videonun gri tonlamalı hale getirilmesi
2. Videonun Motion Blur filtresi ile filtrelenmesi
3. Motion Blur ile filtrelenen videonun Sobel filtresi ile filtrelenmesi
4. Elde edilen videonun siyah bileşenler (0 ağırlık değerine sahip pikseller) ihmal edilerek histogram karşılaştırması ile ardışıl kareler fark değerlerinin elde edilmesi
5. Fark değerleri içinde eşik değeri üzerinde kalan karelerin bölüm geçişi olarak belirlenmesi
6. Bulunan bölüm geçişleri kullanılarak bölümlerin oluşturulması

Şekil 5.4'de Filtrelenmiş Video Histogram Karşılaştırması algoritmasına ait akış diyagramı yer almaktadır. Diğer algoritmalarından farklı olarak eşik değeri ardışıl kareler fark değerlerinin bulunmasından sonra işleme alınmıştır. Bunun sebebi kullanıcının eşik değerini değiştirdiğinde tekrar uzun bir hesaplama sürecini beklemeden değişimleri gözlemleyebilme olanağı tanımaktır. Fark değerlerinin hesaplanması ağır bir işlem yüküdür ve uzun zaman almaktadır eşik değerinin fark değerlerine bir etkisi olmadığından fark değerleri bir kez hesaplandıktan sonra eşik değeri değiştirilerek bölümlendirme sonuçları hızlıca gözlemlenebilmektedir.



Şekil 5.4. Filtrelenmiş Video Histogram Karşılaştırması algoritmasına ait akış diyagramı

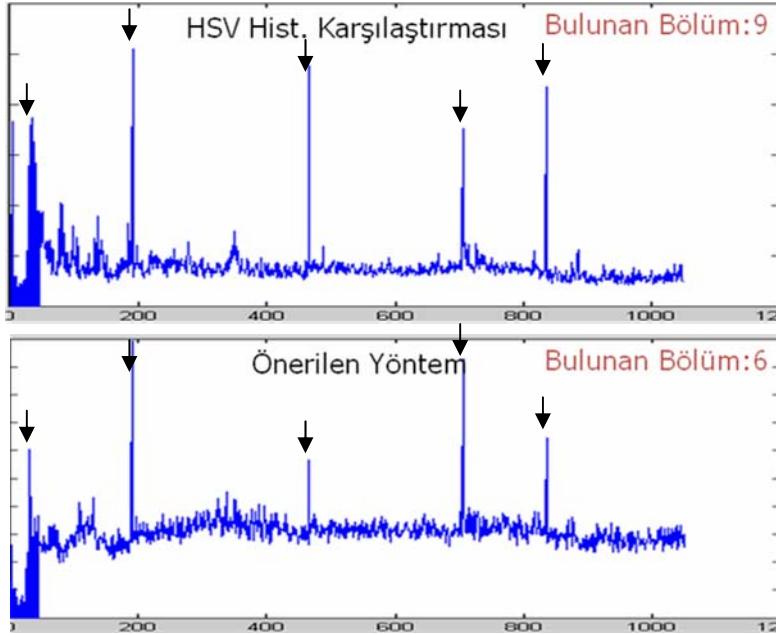
Filtrelenmiş Video Histogram Karşılaştırması obje kenarlarını izleyerek parlaklık değişimlerine duyarlılığı azaltmaktadır. Erime ve solma efektlerinde şekil kademeli olarak görüntüden çıkarıldığı için kenarları da şekiller ile birlikte kaybolmaktadır. Bu sebeple yöntem erime ve solma efektlerini algılamak için kullanılabilir. Erime ve solma efektinin hızlı uygulandığı bölüm geçişleri için yöntem direkt olarak kullanılabilir. Ekeftlerin yavaş uygulandığı bölüm geçişlerini algılamak için Filtrelenmiş Video Histogram Karşılaştırmasının ikiz karşılaştırma yöntemi ile birlikte kullanılması çok başarılı sonuçlar elde edilmesini sağlayacaktır.

5.3. Değerlendirme ve Karşılaştırmalar

Önerilen yöntemin uygulanması ile elde edilen neticeler değerlendirildiğinde beklendiği gibi Filtrelenmiş Video Histogram Karşılaştırması yönteminin kamera hareketleri ve parlaklık değişimine duyarlılığının, diğer sıkıştırılmamış video bileşenleri üzerine çalışan yöntemlere göre çok daha düşük olduğu tespit edilmiştir. Yapılan denemelerde özellikle parlaklık değişimlerine duyarlılığın çok düşürüldüğü ve kameradan kaynaklanan parlaklık değişimleri içeren videolarda çok daha başarılı bölümlendirme yapılabildiği gözlemlenmiştir.

Şekil 5.5’de obje ve kamera hareketleri içeren örnek bir video (TRECVID web sayfasından temin edilen chevrolet.avi) için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri grafiksel olarak görülmektedir. HSV histogram karşılaştırması standart yöntemler içinde obje parlaklık değişimine en az duyarlı yöntemlerden olduğu için karşılaştırma için seçilmiştir. Şekil üzerinde oklar ile gösterilen noktalar gerçek bölüm geçişleridir. Algoritmalar tarafından gerçek bölüm geçişleri dışında elde edilen yüksek fark değerlerinin en az olması istenmektedir. Kullanılan video 1 ila 200. kareler arası sürekli obje hareketleri içermektedir. Grafik üzerinde görüldüğü üzere HSV Histogram karşılaştırması 1-200 kareler arası yüksek değerler elde etmekte, video başlangıcında yer alan parlaklık değişimini ve ilk bölüm geçişi etrafında aslında bölüm geçişi olmadığı halde fazladan kareleri bölüm geçişi olarak algılamaktadır.

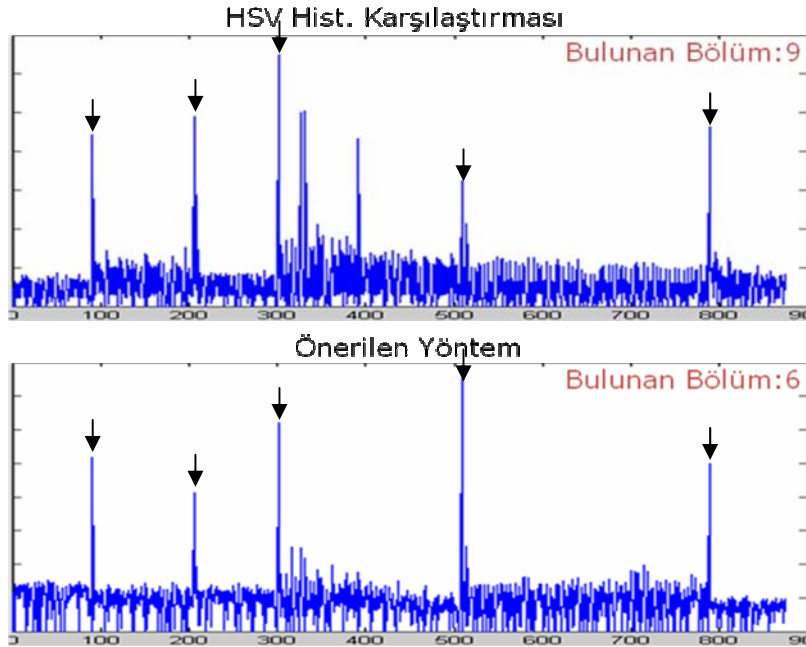
Şekil 5.5’de yer alan sonuçları elde etmek için kullanılan video gerçekte 6 bölüm içermektedir. Filtrelenmiş Video Histogram Karşılaştırması yönteminin bu bölümlerin tamamını doğru bir şekilde algılayabildiği gözlemlenmektedir.



Şekil 5.5. Objeye ve kamera hareketleri içeren örnek bir video için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri

Şekil 5.6’da parlaklık değişimi ve kamera hareketlerinin yoğun olduğu örnek bir video için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri grafik olarak görülmektedir. Örnekte kullanılan video, özellikle kalitesiz bir kamera ile kamera ve parlaklık değişimlerini ihtiva edecek şekilde kaydedilmiştir. Örnek video 300-515. kareleri arasında bulunan 4. bölümünde şiddetli parlaklık değişimi ve kamera hareketleri içermektedir. HSV Histogram Karşılaştırması yönteminin 4. Bölüm içerisinde parlaklık değişiminin olduğu kare fark değerlerini 5. bölüm başlangıç değerinden daha yüksek elde ettiği gözlemlenmektedir. Ayrıca HSV Histogram Karşılaştırması yönteminin 5. Bölüm başlangıcından hemen sonra yer alan kamera hareketini bölüm geçiş değerine yakın elde ettiği gözlemlenmektedir.

Şekil 5.6’da görülen sonuçlar için kullanılan video gerçekte 6 bölüm içermektedir. Filtrelenmiş Video Histogram Karşılaştırması yönteminin bu bölümlerin tamamını doğru bir şekilde algılayabildiği gözlemlenmektedir.



Şekil 5.6. Parlaklık değişimi ve kamera hareketlerinin yoğun olduğu örnek bir video için HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması tarafından elde edilen kare fark değerleri

Yukarıda yer alan iki örnek video için elde edilen sonuçlar incelendiğinde Filtrelenmiş Video Histogram Karşılaştırması yönteminin parlaklık değişimleri, obje ve kamera hareketlerine duyarlılığının diğer yöntemlere göre çok daha düşük olduğu gözlemlenmektedir.

Ayrıca Filtrelenmiş video histogram karşılaştırması yöntemi üç renk bileşeni yerine gri tonlamalı piksel değerlerini kullanması ve düz alanların fark değeri elde edilirken ihmal edilmesi sebebiyle genel olarak daha düşük fark değerleri elde ederek düşük eşik değerleri ile başarılı bölümlendirme yapabilmektedir. Fark değerlerinin ve eşik değerinin daha düşük sayılarda olması sebebiyle filtrelenmiş video histogram karşılaştırması bellek kullanımını dikkate değer bir şekilde düşürmektedir. Daha düşük sayı aralıklarında karşılaştırma yapılması sebebiyle bölüm geçişlerinin bulunması daha hızlı yapılabilmektedir.

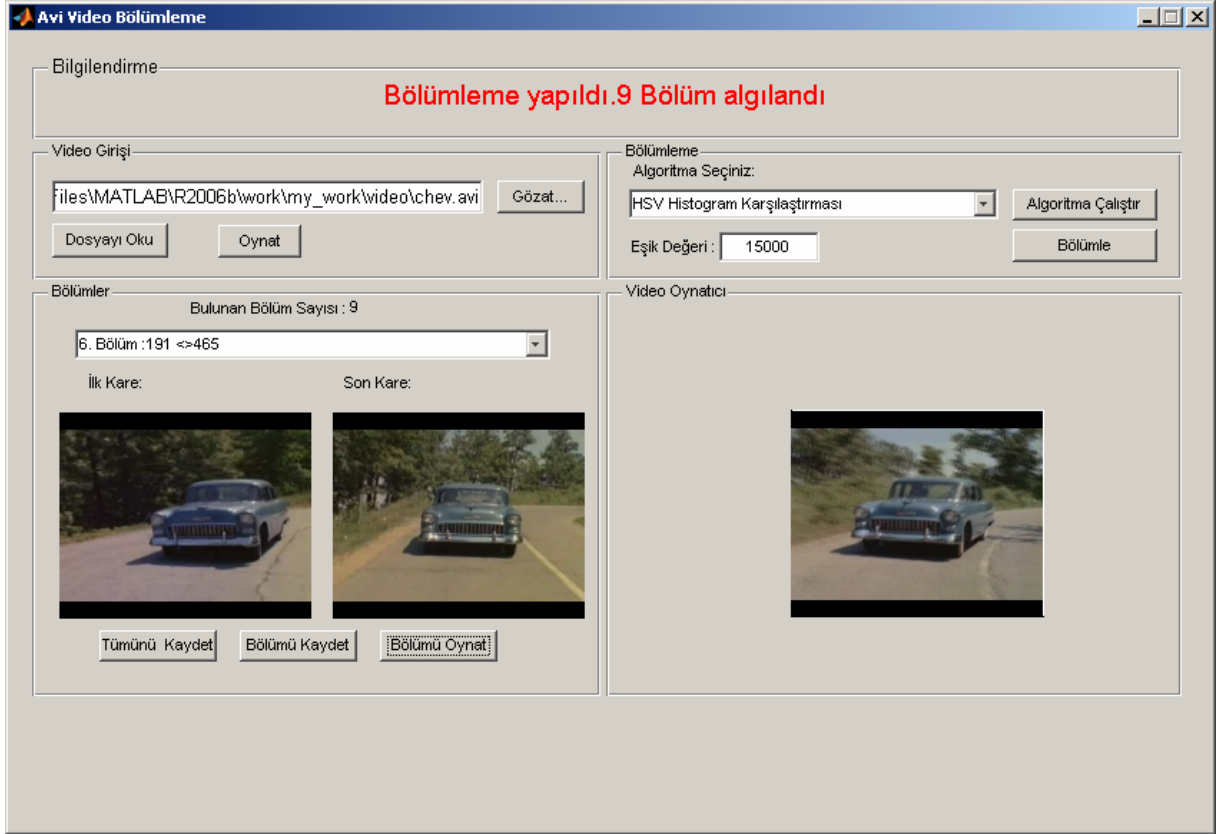
Filtrelenmiş Video Histogram Karşılaştırması hesaplama zamanı olarak değerlendirilecek olursa; filtreleme adımlarının hesaplama zamanını arttırdığı açıktır. Fakat video kare fark değerlerinin elde edildiği adımda siyah beyaz video üzerine çalışıldığı için Filtrelenmiş Video Histogram Karşılaştırması bu adımı tek aşamada tamamlamaktadır. Diğer yöntemler ise hesaplama adımını üç renk için tekrarlayıp ortalama değeri hesapladıklarından video kare fark değerlerini daha uzun zaman içinde elde etmektedirler.

BÖLÜM 6. VIDEO BÖLÜMLENDİRME ALGORİTMALARI UYGULAMA VE OTOMATİK PERFORMANS KARŞILAŞTIRMASI ARAYÜZLERİ

Yapılan çalışmada sıkıştırılmamış video bileşenleri üzerine çalışan bölümlendirme algoritmalarının kullanım imkanını ve otomatik performans karşılaştırması yapılmasını sağlayacak iki adet arayüz hazırlanmıştır. Hazırlanan arayüzler Matlab R2006b programlama ortamında hazırlanmıştır. Hazırlanan arayüzlerde Piksel Karşılaştırması, Blok Tabanlı Piksel Karşılaştırması, RGB Histogram Karşılaştırması, Blok Tabanlı Histogram Karşılaştırması, HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması yöntemlerinin kullanılmasına imkan tanınmaktadır.

6.1. Video Bölümlendirme Algoritmaları Uygulama Arayüzü

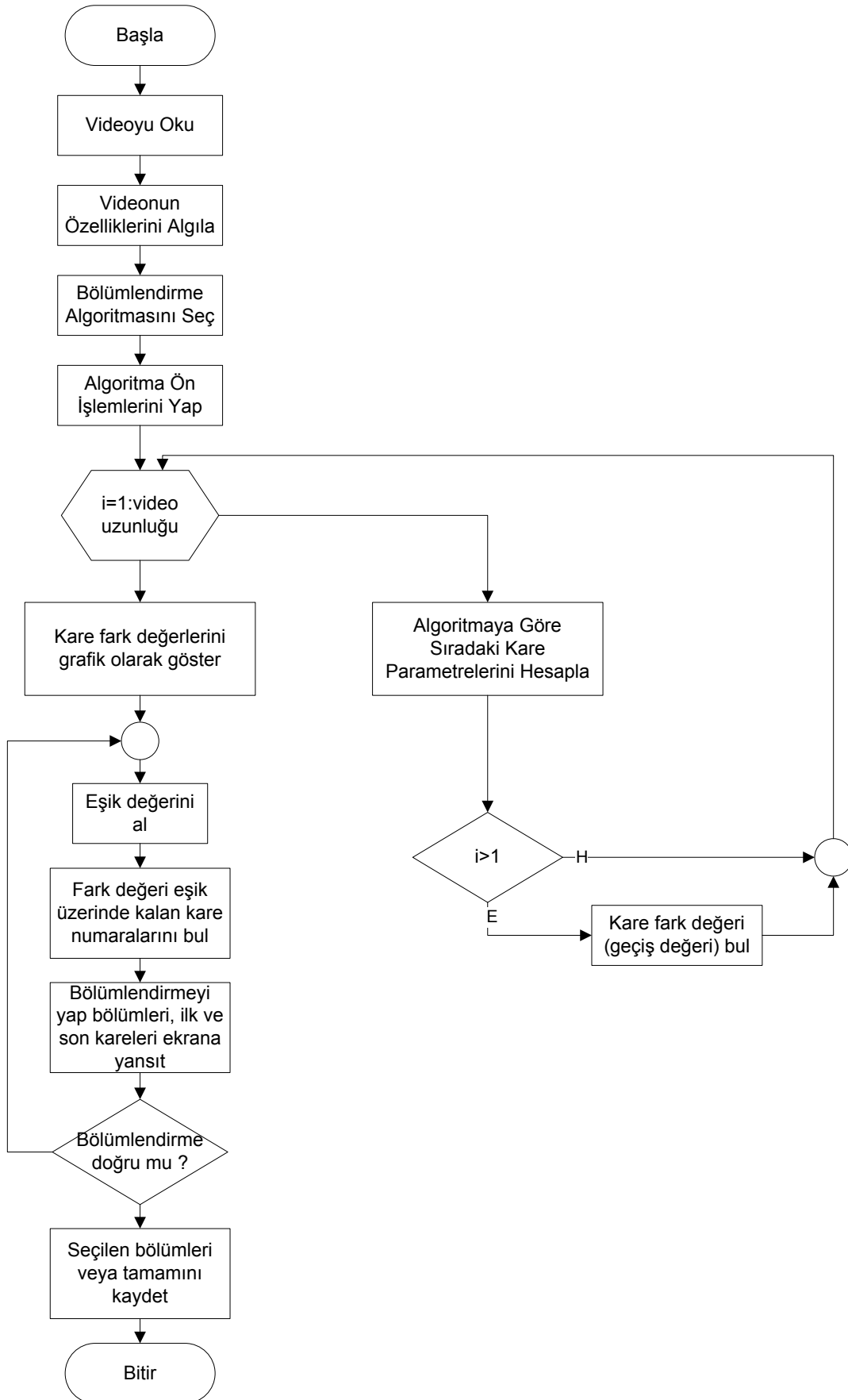
Video bölümlendirme algoritmalarının kullanımını kolaylaştırmak ayrıca bu alanda gelecekte çalışacak araştırmacıların konuya daha hızlı hakim olmalarını sağlayacak bir araç oluşturmak hedefiyle Video Bölümlendirme Algoritmaları Arayüzü oluşturulmuştur. Şekil 6.1’de hazırlanan Video Bölümlendirme Algoritmaları Arayüzü görülmektedir. Hazırlanan arayüz video bölümlendirme algoritmalarını kullanımını komut seviyesinden görsel seviyeye taşıyarak daha esnek bir kullanım ve sonuçları gözleme imkanı ile amaca uygun başarılı bir video bölümlendirme imkanı tanımaktadır.



Şekil 6.1. Video Bölümlendirme Algoritmaları Arayüzü

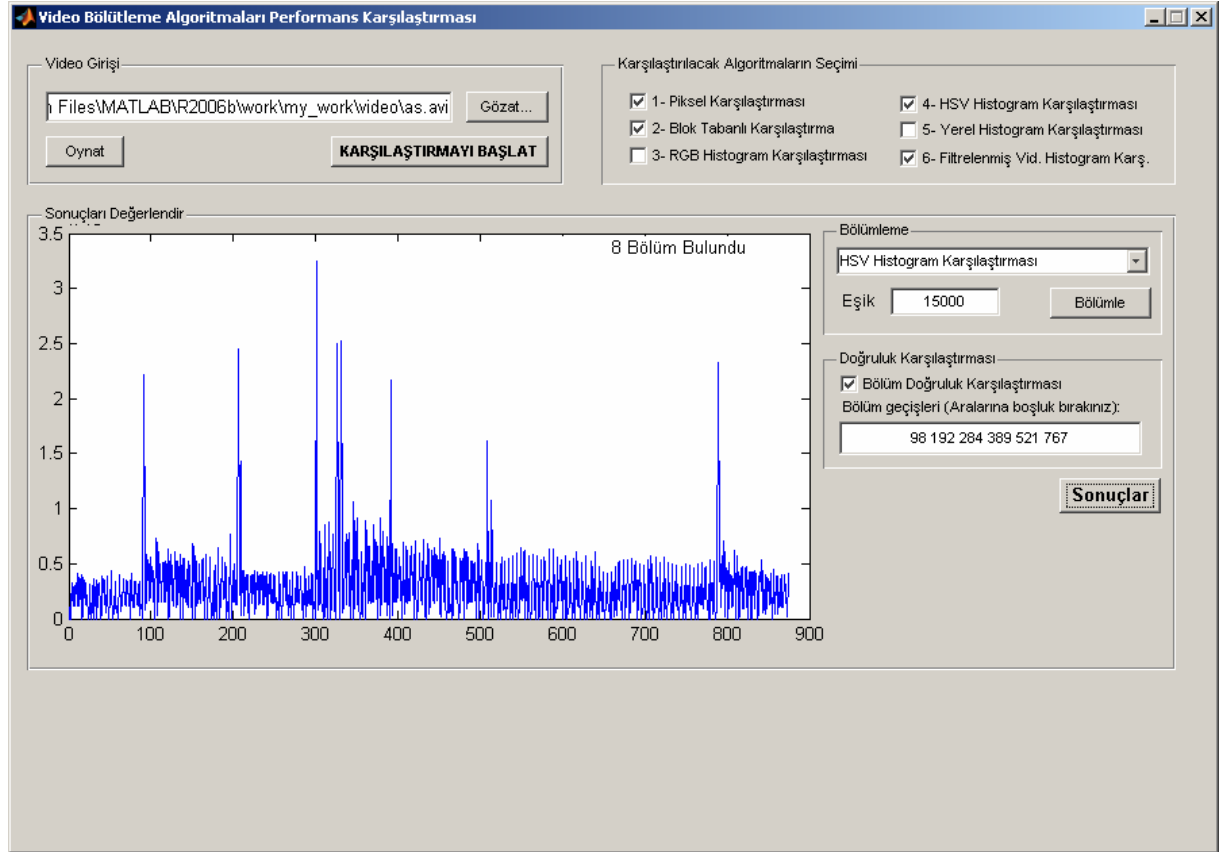
Video Bölümlendirme Algoritmaları Arayüzü'nün kullanımı kullanıcının video girişi yapması ile başlamaktadır. Arayüzün desteklediği dosya formatı sıkıştırılmamış avi uzantılı video dosyalarıdır. Video başarı ile okunduktan sonra kullanıcı tarafından bölümlendirme işleminde kullanılacak arayüz seçimi yapılarak Algoritma Çalıştır butonuna tıklanır. Bu noktadan sonra video seçilen algoritma tarafından işlenerek kareler arası geçiş değerleri hesaplanır. Kare geçiş değerleri hesaplamasının yapıldığı sırada ekranda bir durum çubuğu (progress bar) belirterek kullanıcıyı hesaplamanın ne kadarının tamamlandığı hakkında bilgilendirmektedir. Hesaplama sonucunda elde edilen kare geçiş değerleri x ekseninde kare numaraları y ekseninde ilgili kare fark değeri olacak şekilde grafik olarak ekrana yansıtılmaktadır. Fark değerlerini inceleyen kullanıcı tarafından eşik değeri girilerek bölümle düğmesine tıklanır. Bu adımda arayüz tarafından kare fark değerlerini taranarak eşik değeri üzerinde kalan kareler bölüm geçişi olarak algılanır ve bölümler oluşturulur. Oluşturulan bölüm sayısı ve bölümler açılır menüye (pop-up menu) yüklenmektedir. Kullanıcı tarafından seçilen bölüm ilk ve son kareleri ekranda gösterilir ve kullanıcı istediği

taktirde Bölümü Oynat düğmesine tıklayarak bölümü Video Oynat penceresinde izleyebilmektedir. Kullanıcı tarafından doğru bölümlendirme yapılamadığı kanaatine varılırsa eşik değeri değiştirilerek aynı yöntem için bölümlendirme veya farklı bir yöntem seçilerek bölümlendirme yapılabilmektedir. Doğru bölümler elde edildikten sonra istenirse seçilen bölüm belirlenen dosya yoluna kaydedilebilmektedir veya tüm bölümler belirlenen dosya yoluna otomatik isimlendirme ile kaydedilebilmektedir. Şekil 6.2’de Video Bölümlendirme Algoritmaları Arayüzü akış diyagramı ana hatları ile görülmektedir.



Şekil 6.2. Video Bölümlendirme Algoritmaları Arayüzü akış diyagramı

6.2. Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzleri



Şekil 6.3. Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü

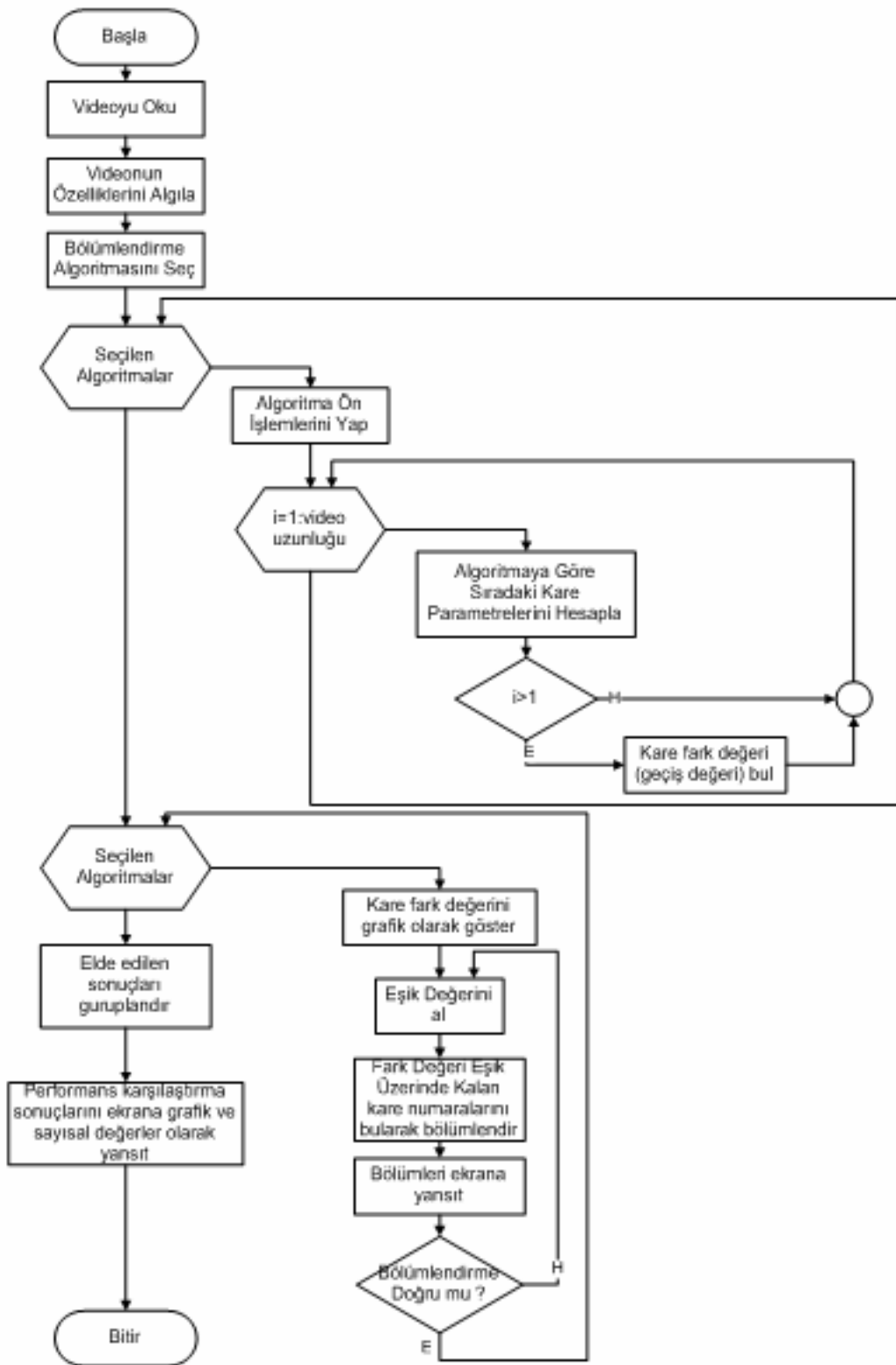
Video bölümlendirme algoritmaları üzerine yapılan çalışmalar neticesinde literatürde birçok video bölümlendirme algoritması yer almaktadır. Halen bütün video türlerinde aynı anda çalışarak başarılı bölümlendirme yapacak bir algoritma geliştirilememiştir. Literatürde geçen bütün yöntemlerin diğer yöntemlere göre avantaj ve dezavantajları vardır. Bu bağlamda video bölümlendirme algoritmalarının performans karşılaştırmalarının yapılarak diğer yöntemlere göre avantaj ve dezavantajlarının ortaya konması gelecekte bu konuda yapılacak çalışmalara ışık tutacaktır. Ayrıca algoritmaların başarımlarının video türleri arasında farklılıklar göstermesi nedeniyle performans karşılaştırması yapmak, kullanıcı tarafından bölümlendirilecek videoya en uygun yöntemin belirlenmesini sağlayacaktır.

Video bölümlendirme algoritmalarının performans karşılaştırmaları yapılmak istendiğinde ortak bir video için bütün algoritmalar ayrı ayrı çalıştırılmalı ve elde edilen sonuçlar kayıt altına alınmalıdır. Algoritmaların performans karşılaştırmalarının bir bilgisayar programı ile otomatik olarak gerçekleştirilmesinin çok faydalı bir gelişme olacağı açıktır. Bu amaçla geliştirilen Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü Şekil 6.3'de görülmektedir.

Geliştirilen arayüz sayesinde sıkıştırılmamış videolar üzerine çalışan algoritmalarından seçilen algoritmalar girilen video için karşılaştırılmakta ve sonuçlar kullanıcıya sayısal değerler ve grafikler olarak aktarılmaktadır. Geliştirilen arayüzde algoritmalarının performans karşılaştırma kriterleri şunlardır:

- Süre : Algoritmanın hesaplanma zamanı.
- Bulunan Bölüm Adedi : Algoritma tarafından bulunan bölüm adetleri; girilen video için gerçek bölüm değeri kadar bölüm bulunmalıdır.
- Bölüm Doğrulukları (isteğe bağlı) : Girilen videonun bölüm geçiş kare numaraları biliniyorsa arayüze girilerek algoritma tarafından bulunan bölümlerin doğruluğu kontrol edilebilmektedir. İdealde girilen bölüm geçişlerinin aynen bulunması istenmektedir (idealde %100).
- Fazladan Bulunan Bölüm Adedi (isteğe bağlı) : Algoritma tarafından bulunan bölüm sayısının gerçek bölüm sayısından ne kadar fazla olduğu. İdealde bütün bölümlerin aynen bulunması istenmektedir (idealde 0).

Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü akış diyagramı Şekil 6.4'de görülmektedir. Arayüzün kullanımı kullanıcı tarafından video girişinin yapılması ile başlar. Kullanıcı tarafından video girişi yapıldıkça diğer kontroller pasiftir, girişin yapılmasıyla diğer kontroller aktif olur. Video girişi ardından performans karşılaştırması yapılacak algoritmalar seçilir. Kullanıcı tarafından Karşılaştırmayı Başlat düğmesine tıklanması ile hesaplama süreci başlamaktadır. Hesaplama süreci içinde seçilen algoritmalar sıra ile çağrılarak videoya uygulanır. Her algoritmanın hesaplama zamanının ne kadar zaman aldığı ve kareler arası fark değerleri kaydedilerek diğer algoritmaya geçilir.



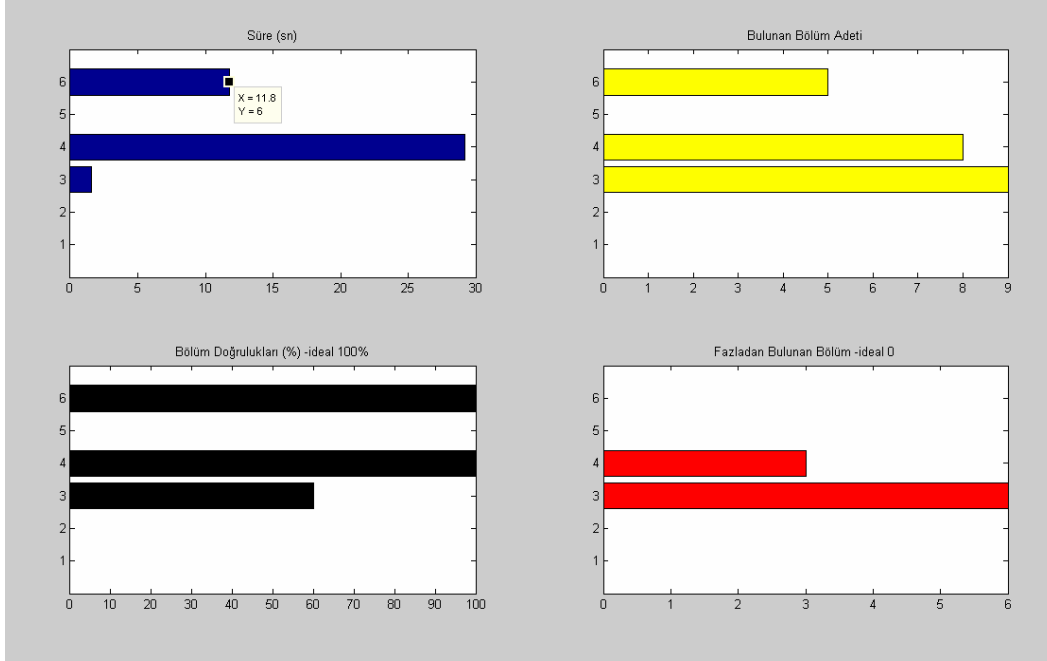
Şekil 6.4. Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü akış diyagramı

Hesaplama süreci içinde ekrana gelen bir durum çubuğu (progress bar) ile kullanıcı bilgilendirilir. Seçilen tüm algoritmalar bittiğinde kullanıcı bir mesajla uyarılır.

Hesaplama süreci ardından sonuçları gözlemek için bölümlene alanında bulunan açılır menüden algoritmalar seçilir. Açılır menüden seçilen algoritma tarafından elde edilen kare geçiş değerleri grafik olarak kullanıcıya sunulur. Kullanıcı grafiği inceledikten sonra eşik değeri girişini yaparak Bölümle düğmesine tıklar. Seçilen tüm algoritmalar için bölümle işlemi tamamlandıktan sonra Sonuçlar düğmesi aktif hale gelerek sonuçları gözleme imkanı sunar. Bu adımda sonuçlar düğmesine tıklanırsa sonuçlar grafiği açılacaktır fakat sonuçlar da sadece süre ve bulunan bölüm sayısı yer alacaktır.

Bölüm Doğrulukları ve Fazladan bulunan bölüm adedi karşılaştırması yapılmak isteniyorsa Doğruluk Karşılaştırması onay kutusu işaretlenir. Video gerçek bölüm geçiş kare numaraları biliniyorsa bölüm geçişleri metin alanına birer boşluk karakteri bırakarak yazılır. Eğer bölüm geçiş kare numaraları bilinmiyorsa video oynat düğmesine tıkladığı takdirde video oynatıcı ekrana gelerek kullanıcıya bölüm geçişlerini bulmak için yardımcı olacaktır. Kullanıcı tarafından bölüm geçiş kare numaraları ara yüze girildiği takdirde videoya ait gerçek bölüm sayısı program tarafından algılanmış olacaktır. Bölüm Doğrulukları onay kutusu işaretlendiği için sonuçlar alanına bölüm doğruluk oranı ve fazladan bulunan bölüm geçişleri dahil edilmektedir.

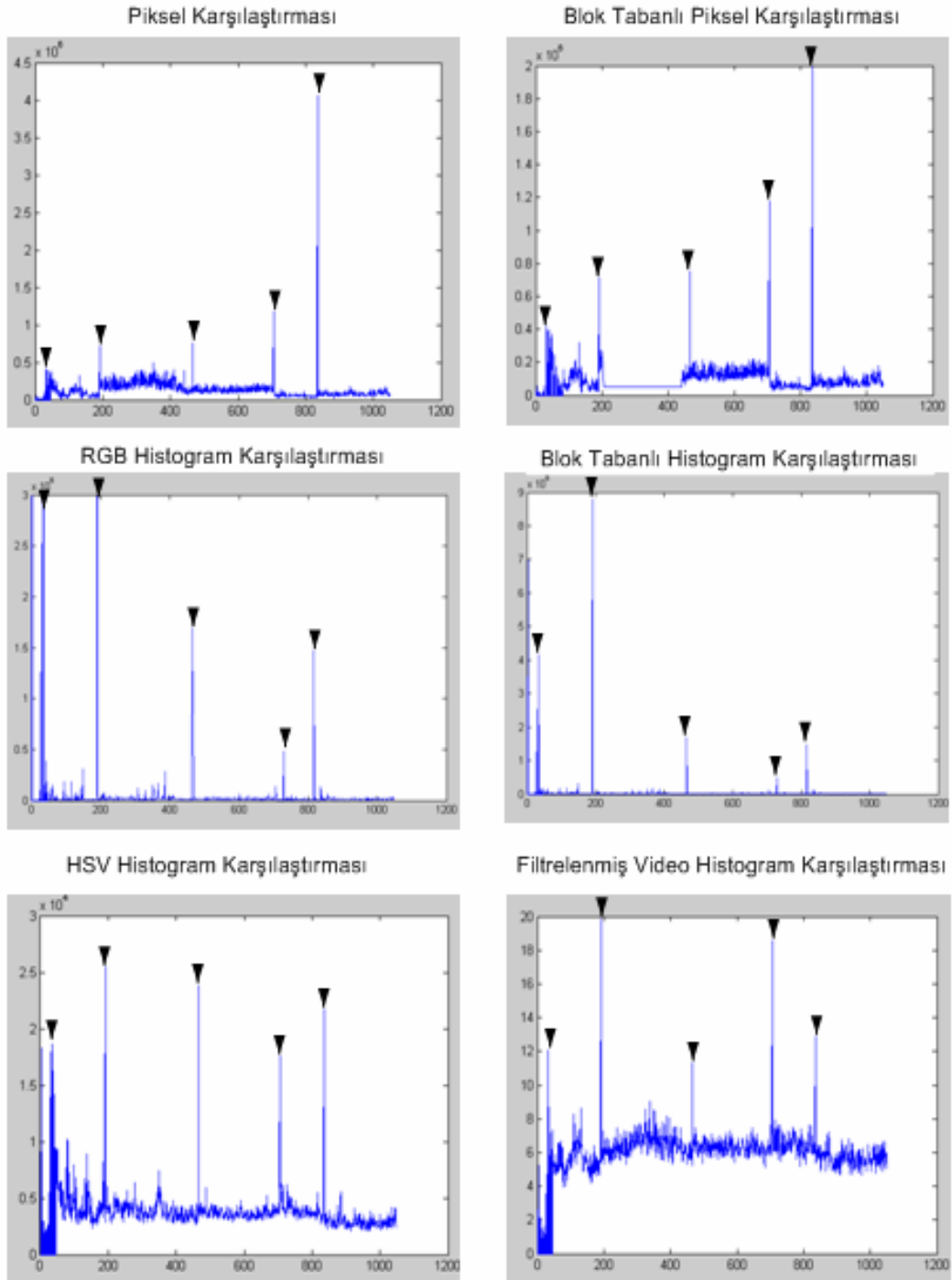
Arayüzde RGB Histogram Karşılaştırması, HSV Histogram Karşılaştırması ve Filtrelenmiş Video Histogram Karşılaştırması algoritmalarının seçilerek as.avi videosu için yapılan performans karşılaştırması grafiksel sonuçları görülmektedir. Karşılaştırma sonucunda 4 karşılaştırma kriteri için algoritmaların sonuçları gruplandırılarak kullanıcıya sunulmaktadır. Grafiklerde algoritmalar arayüzde seçim bölgesinde yer alan numaraları ile ifade edilmektedir. Karşılaştırma için seçilmeyen algoritmaların alanları grafikte boş bırakılmaktadır.



Şekil 6.5. Video Bölümlendirme Algoritmaları Performans Karşılaştırması Arayüzü grafiksel sonuçları

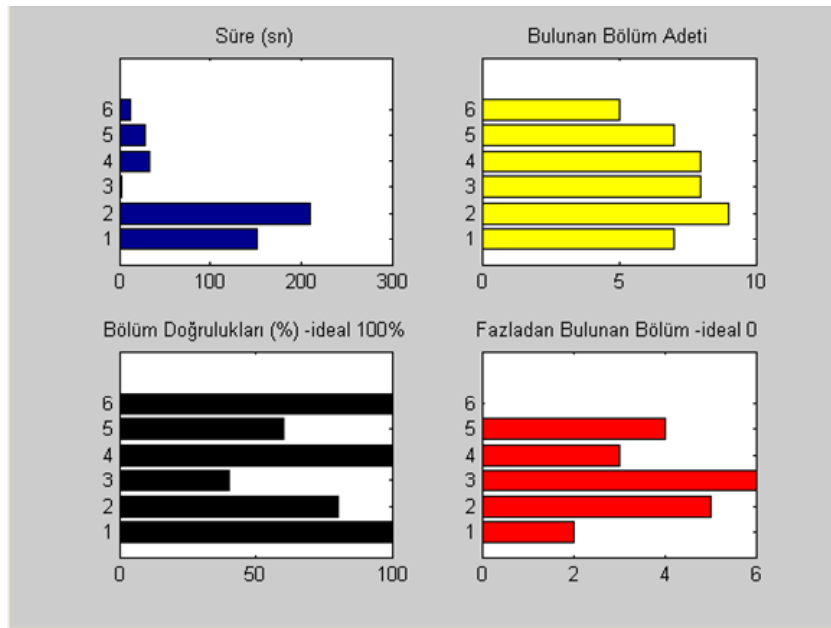
Grafik sonuçlarında sol üst grafik algoritmaların hesaplama sürelerini sn cinsinden göstermektedir. Sağ üst grafikte algoritma tarafından algılanan bölüm adetleri ifade edilmektedir. Bölüm doğruluk karşılaştırması onay kutusu seçilerek kullanıcı tarafından gerçek bölüm geçişleri (1. ve son kare numarası hariç) girildiği takdirde altta yer alan grafik alanları aktif olmaktadır. Sol alt grafikte elde edilen bölüm doğrulukları gösterilmektedir eğer algoritma tarafından bulunan bölüm geçişleri gerçek bölüm geçişlerinin birebir aynısı ise %100 doğruluk elde edilmekte aksi durumlarda ise gerçek bölüm geçişlerinin elde edilme oranı % olarak gösterilmektedir. Sağ alt grafikte ise algoritma tarafından elde edilen bölüm geçişlerinin kaç tanesinin gerçek bölüm geçişlerinden farklı olduğu adet olarak ifade edilmektedir. Yüksek kalitede bölümlendirme için istenen değerler; bölüm doğruluğunun %100, fazladan bulunan bölüm adedinin 0 olmasıdır. Grafiksel sonuçlar elde edildikten sonra kullanıcı tarafından kesin değerlerin görülmesi isteniyorsa ilgili noktaya tıklanarak Şekil 6.6 da 6. algoritma süre değerinde görüldüğü gibi bir bilgilendirme kutucuğu açılarak kullanıcıya bilgi sunmaktadır.

6.3. Video Bölümlendirme Algoritmaları Performans Karşılaştırmaları



Şekil 6.6. Chevrolet.avi videosu için bölümlendirme algoritmaları tarafından elde edilen kare geçiş değerleri

Video bölümlendirme algoritmalarının karşılıklı avantaj ve dezavantajlarını ortaya koymak amacıyla bölüm geçiş türleri ve görsel içerik açısından farklı bölümler içeren videolar için performans karşılaştırmalarına yer verilmiştir. Video bölümlendirme algoritmalarının performans karşılaştırmasının aynı video için ulaşılan değerlendirilmesi ile yapılmasının gerekliliği açıktır. Bu sebeple aynı video için bölümlendirme algoritmaları çalıştırılarak elde edilen kare geçiş değerleri şekillerde verilmiştir. Şekiller üzerinde oklar ile gösterilen değerler gerçek bölüm geçişleridir. Başarılı bir video bölümlendirme için bölüm geçişlerinin bölüm içi geçiş değerlerine göre yüksek değerli olması ve bölüm içlerinde en düşük değerle elde edilen gerçek geçiş değerinden büyük fark değeri olmaması gerekmektedir. Performans karşılaştırmalarını sayısal değerler ile grafik olarak ifade etmek için aynı videolar için tüm algoritmalar seçilerek Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü çalıştırılarak elde edilen sonuçlar şekillere eklenmiştir.



Şekil 6.7. Chevrolet.avi videosu için Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü tarafından elde edilen karşılaştırma grafiği

Şekil 6.7 de TRECVID web sayfasından elde edilen chevrolet.avi videosu için bölümlendirme algoritmalarının hesapladıkları kare geçiş değerleri grafikleri görülmektedir. Chevrolet.avi videosu 33,191,466,706,836 nolu karelerde geçişleri

bulunan 6 bölümden oluşmaktadır. Grafikler incelendiğinde video başlangıcında yer alan obje hareketlerinin bazı algoritmalar tarafından yüksek değerler elde edilmesine sebep verdiği görülmektedir. Chevrolet.avi videosu içindeki ilk bölüm geçişi solma efekti ile hazırlanmış bir kademeli bölüm geçişidir. Geçişin ardından gelen bölüm kademeli olarak görüntüye girerken aynı zamanda obje hareketleri barındırmaktadır. Grafikler incelendiğinde Filtrelenmiş Video Histogram Karşılaştırması haricindeki algoritmaların ilk bölüm geçişi civarında yüksek değerler elde ederek fazladan bölümlendirme yaptıkları görülmektedir.

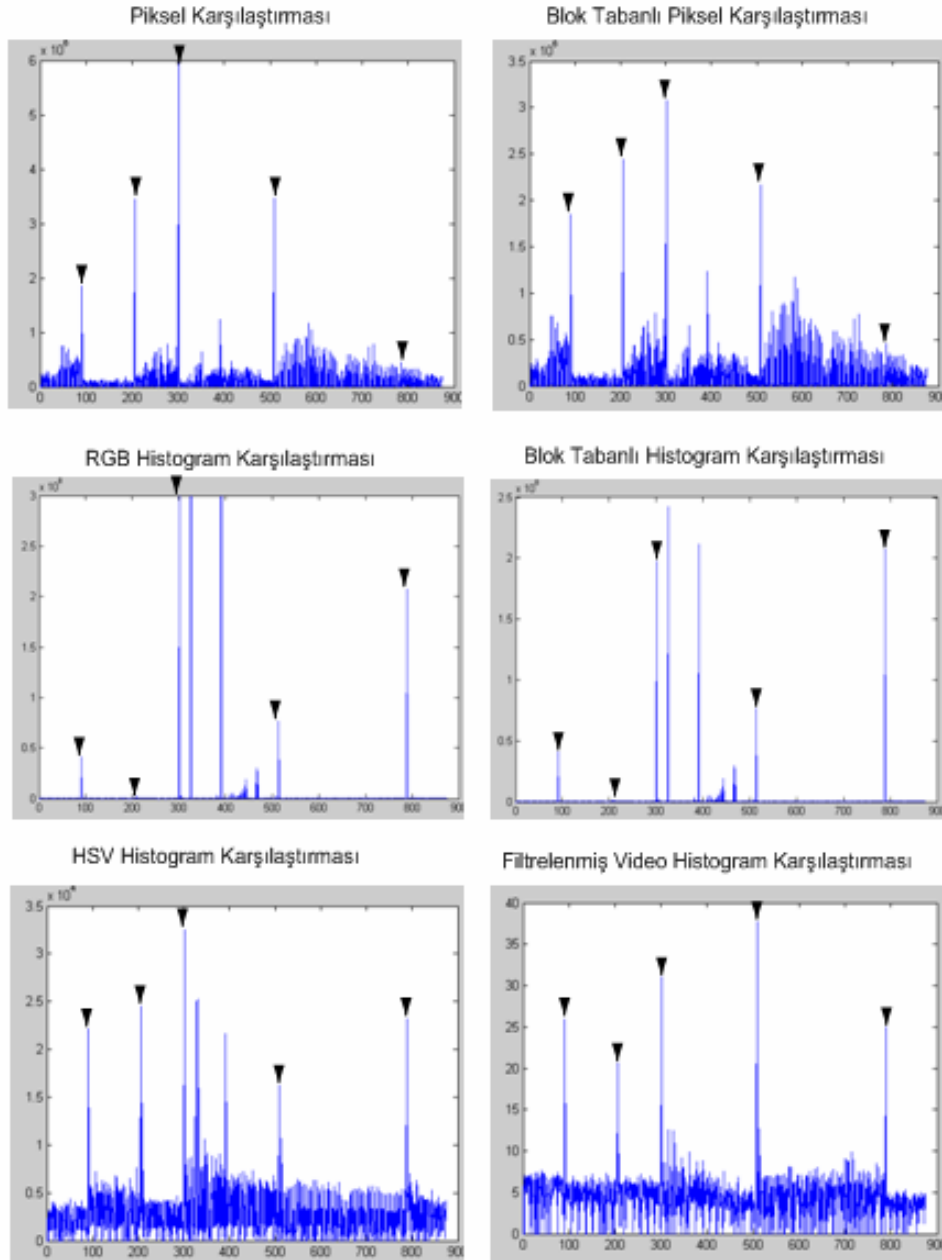
Performans karşılaştırma arayüzünün grafik çıktısında 4 kriter için sonuçlar görülmektedir. Ayrıca Tablo 6.1’de bütün algoritmalar için hesaplama süresi, bulunan bölüm adedi, gerçek bölüm geçişlerinin doğru bulunma oranı ve gerçek bölüm geçişleri haricinde bulunan bölüm geçiş sayıları verilmiştir.

Tablo 6.1. Chevrolet.avi videosu için algoritmaların elde ettikleri sayısal değerler Chevrolet.avi videosu için algoritmaların elde ettikleri sayısal değerler.

ALGORİTMA	Süre (sn)	Bölüm Adedi	Doğruluk (%)	Fazla Bölüm
Piksel Karşılaştırması	153.41	7	100	2
Blok Tabanlı Piksel Karşılaştırması	212.23	9	80	5
RGB Histogram Karşılaştırması	2.74	8	40	6
Blok Tabanlı Histogram Karşılaştırması	27,09	7	60	4
HSV Histogram Karşılaştırması	34.98	8	100	3
Filtrelenmiş Video Histogram Karşılaştırması	17.67	5	100	0

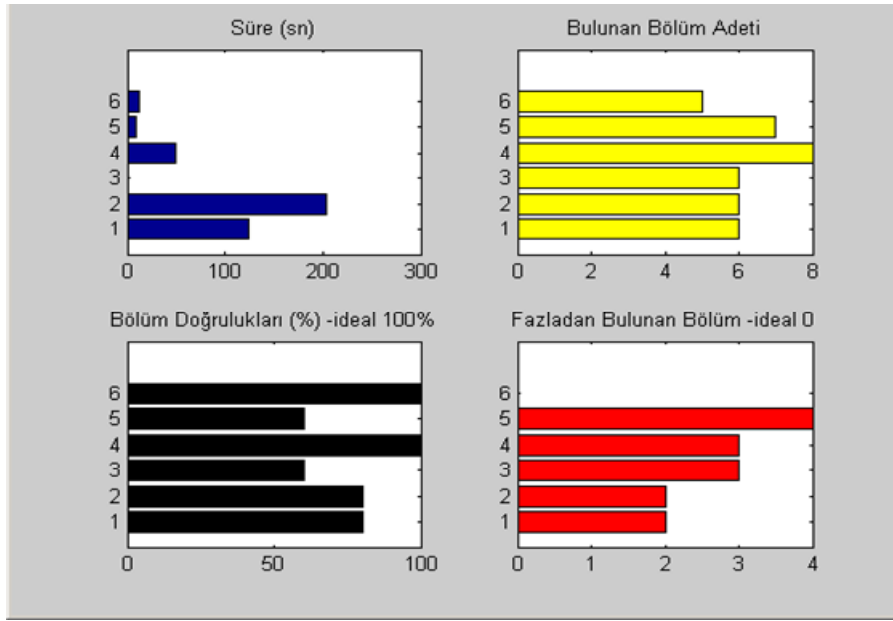
Bölüm geçişleri dışında çok fazla parlaklık değişimi ve kamera hareketi içerecek şekilde kaydedilmiş as.avi videosu için bölümlendirme algoritmalarının elde ettikleri kare geçiş değerleri Şekil 6.8’de görülmektedir. As.avi videosu 91,206,301,509,789 nolu karelerde bölüm geçişleri olan 6 bölümden oluşmaktadır. Grafikler incelendiğinde standart yöntemlerin bölüm içlerinde olan kareler için çok yüksek geçiş değerleri elde ettikleri gözlemlenmektedir. Özellikle 301-509 kareleri arasında bulunan 4. Bölüm içinde çok keskin parlaklık geçişleri bulunmaktadır. Grafiklerden

görüldüğü üzere 4. Bölüm içindeki parlaklık değişiminin bulunduğu kare fark değerleri bölüm geçişi olarak algılanabilecek kadar yüksektir. Özellikle histogram tabanlı yöntemlerde bu noktaların gerçek bölüm geçişlerinden bile yüksek olduğu gözlemlenmektedir. Histogram tabanlı yöntemlerdeki bu hatanın sebebi daha öncede bahsedildiği gibi parlaklık değişimleri ile resim histogramının sağa ve sola kaymasıdır.



Şekil 6.8. As.avi videosu için bölümlendirme algoritmaları tarafından elde edilen kare geçiş değerleri.

As. avi için Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü ile elde edilen sayısal değerler Tablo 6.2’de; bu değerlere ait grafik çıktısı ise Şekil 6.9’da görülmektedir. Sayısal değerler incelendiğinde Filtrenilmiş Video Histogram Karşılaştırması haricinde tüm yöntemlerin parlaklık değişimi ve kamera hareketlerinden dolayı gerçek değerler dışında bölüm geçişleri algılayarak fazla bölüm buldukları görülmektedir.



Şekil 6.9. As.avi videosu için Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü tarafından elde edilen karşılaştırma grafiği

Yukarıda bölümlendirme algoritmalarının doğru bölümlendirme doğruluğuna göre karşılaştırmaları yapılırken her yöntem için grafikler incelenerek en doğru sonuçları verecek eşik değerleri seçilmiştir. Eğer eşik değerleri benzer oranda seçilerek bir karşılaştırma yapılırsa Filtrenilmiş Video Histogram Karşılaştırmasının daha da başarılı sonuçlar elde edeceği açıktır. Diğer yöntemler tamamı çok büyük sayılarda kare fark değerleri elde ederek yüksek eşik değerlerine ihtiyaç duymaktadırlar.

Tablo 6.2 As.avi videosu için algoritmaların elde ettikleri sayısal değerler

ALGORİTMA	Süre (sn)	Bölüm Adedi	Doğruluk (%)	Fazla Bölüm
Piksel Karşılaştırması	123.73	6	80	2
Blok Tabanlı Piksel Karşılaştırması	201.18	6	80	2
RGB Histogram Karşılaştırması	1.61	6	60	3
Blok Tabanlı Histogram Karşılaştırması	9.32	7	60	4
HSV Histogram Karşılaştırması	48.71	8	100	3
Filtrelenmiş Video Histogram Karşılaştırması	11.65	5	100	0

Video bölümlendirme algoritmalarının başarılı bölümlendirme yapabilme gereksinimleri yanında hızlı çalışmaları gerekmektedir. Örneğin sürekli video eklenen dinamik bir video veritabanı için bölümlendirme işlemlerinin kısa sürede tamamlanması daha kısa süre içinde daha çok video eklenmesine imkan tayıacaktır. Yukarıda yer alan sonuçlar kaynak alınarak bölümlendirme algoritmaları hesaplama süresi kriterine göre karşılaştırılırsa genel olarak histogram tabanlı yöntemlerin piksel tabanlı yöntemlere göre daha kısa hesaplama süresine sahip olduğu gözlemlenmektedir. Bunun sebebi piksel tabanlı yöntemlerinde video içindeki toplam piksel sayısı kadar toplama, mutlak değer alma vs. işlemlerin yapılmasıdır ve toplam piksel sayısı kadar döngü adımına ihtiyaç duyulmasıdır. Histogram tabanlı yöntemlerde ise bu kadar yüksek işlem gereksinimi yaşanmamaktadır.

Bölümlendirme algoritmalarında blok tabanlı yöntemler diğer yöntemlere göre daha yavaş çalışmaktadır. Çünkü blok tabanlı yöntemler blokların belirlenmesi, bloklar için ayrı bir döngü kullanılması, hafızada saklanması gereken değişken miktarının artması, birden fazla karşılaştırma işlemi gibi gereksinimlerden dolayı daha uzun hesaplama sürelerine sahiptirler.

Histogram tabanlı yöntemler arasında RGB histogram karşılaştırması en hızlı çalışan yöntemdir çünkü kare geçiş değerlerini elde etmek için ön işlemlere gerek duymamaktadır. Sayısal videolar zaten RGB kodlama tekniğine göre kodlandığı için karşılaştırma işlemine direk başlanabilmektedir. HSV YIQ gibi farklı kodlama

teknikleri üzerine çalışan algoritmalar için RGB kodlanmış videonun ilgili kodlama tekniğine çevrilmesi fazladan bir işlem yükünü beraberinde getirmektedir. Filtrelenmiş video histogram karşılaştırması her ne kadar kare geçiş değerlerini elde etmek için üç renk bileşeni yerine gri kodlama ile ilgilense de videonun gri tonlamaya çevrilmesi ve filtre adımlarının gerçekleşmesi için ek bir işlem yükü gerektirmektedir.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında sıkıştırılmamış video bileşenleri üzerine çalışan bölümlendirme algoritmalarının uygulaması ve performans karşılaştırması yapılarak sıkıştırılmamış video bileşenleri üzerine çalışan yeni bir video bölümlendirme algoritması tasarımı gerçekleştirildi. Video bölümlendirme algoritmalarının tamamının uygulanabilmesi için bir arayüz hazırlandı. Video bölümlendirme algoritmalarının performans karşılaştırmalarını otomatikleştirerek daha hızlı ve etkili bir şekilde yapılmasını sağlamak amacı ile bir arayüz hazırlandı.

Tez çalışması sırasında hazırlanan tüm programlar ve arayüzler MATHWORKS Matlab r2006b programlama ortamında hazırlandı. Matlab r2006b programlama ortamının tercih edilmesinin sebebi teknik programlama gereksinimlerini karşılamak için bünyesinde birçok alt program barındırma özelliğiyle hedeflere hızlı şekilde ulaşma imkanı tanınmasıdır.

Video bölümlendirme algoritmalarının uygulama arayüzü algoritmaların kullanımını ve denenmesini kolaylaştırmaktadır. Bölümlendirme algoritmalarının tamamının aynı arayüz ile kullanılabilmesi, kullanım kolaylığı, komut satırı yerine görsel etkileşim ile kullanıcıya kolaylık sağlanması, bölümlerin ve sayısal değerlerin gözlemlenebilmesi, her adımda bölümlenecek videonun ve elde edilen bölümlerin hızlı ve kolay bir şekilde izlenme ve kolaylıkla kaydedilebilmesi tasarlanan arayüzün avantajları arasındadır. Arayüz, ileride video indeksleme ve video bölümlendirme üzerine yapılacak çalışmalarda Türkçe bir program olması ve istenildiği takdirde ek programların yapılmadan hızlıca yeni bir algoritmanın kolayca arayüze eklenebilmesi gibi özellikleri nedeniyle bir araç olacaktır.

Video Bölümlendirme Algoritmaları Otomatik Performans Karşılaştırması Arayüzü, algoritmaların performans karşılaştırmalarının otomatik olarak hızlı ve basit bir

kullanım yapısıyla yapılmasını sağlamaktadır. Ayrıca arayüz video bölümlendirme algoritmalarının performans karşılaştırmalarını süre, bölüm sayısı, doğruluk ve fazla algılanan bölüm sayısı kriterlerine göre kullanıcıya grafik olarak sunmaktadır. İleride geliştirilecek yeni algoritmaların kaynak kodlarına eklenmesi ile geliştirilen algoritmanın literatürde yer alan diğer algoritmalara göre kolayca performans değerlendirmesini yapılmasını sağlayacaktır.

Tez çalışması sırasında geliştirilen sıkıştırılmamış videolar bileşenleri üzerine çalışan Filtrelenmiş Video Histogram Karşılaştırması bölümlendirme algoritması literatürde aynı kategoride yer alan diğer algoritmalara göre daha doğru bölümlendirme sonuçları elde etmektedir. Özellikle gerçekte bölüm geçişi olmayan ve bölüm içinde içeriğin değişmediği durumlardaki kamera-obje hareketliliği ve parlaklık değişimlerine duyarlılığı dikkate değer seviyede düşürmektedir. Geliştirilen yöntem standart yöntemler yerine videonun filtrelenmesi ile elde edilen obje kenarlarını kullanarak hesaplama yapmaktadır. Geliştirilen yöntem sıkıştırılmış bileşenler üzerine çalışan algoritmalarla birleştirilerek pratikte kullanılabilir niteliktedir.

KAYNAKLAR

- [1] COTSACES C., NIKOLAÏDÌS N., PITAS I., Video shot detection and condensed representation – A review. *IEEE Signal Processing Magazine*, IEEE, 2006
- [2] XIONG Z., ZHOU X. S., TIAN Q., RUI Y., HUANG T. S., Semantic retrieval of video-review of research on video retrieval in meetings, movies and broadcast news, and sports. *IEEE Signal Processing Magazine*, IEEE, 2006
- [3] KOKARAM A., REA N., DAHYOT R., TEKALP A. M., BOUTHEMY P., GROS P., SEZAN I., Browsing Sports Video - Trends in sports-related indexing and retrieval work. *IEEE Signal Processing Magazine*, IEEE, 2006
- [4] ANDROUTSOS D., GUAN L., VENETSANOPOULOS A. N., Semantic Retrieval of Multimedia. *IEEE Signal Processing Magazine*, IEEE, 2006
- [5] KOPRÌNSKA I., CARRATO S., Temporal Video Segmentation : A Survey *Signal Processing: Image Communication*, Elsevier Science, 2001
- [6] KASTURÌ R., JAIN R., Dynamic Vision, in Computer Vision : Principles *IEEE Computer Society Press*, Washington DC,1991
- [7] NAGASAKA A., TANAKA Y., Automatic video indexing and full video search for object appearances, *Visual Database Systems II*, Elseiver, 1995
- [8] TONOMURA Y., Video Handling Based On Structured Information For Hypermedia Systems, *ACM Int. Conf. Multimedia Information Systems*, 1991
- [9] ZHANG H.J., LOW C.Y., SMOLIAR S.W., Video Parsing and Browsing Using Compressed Data, *SPIE Conf. Image and Video Processing II*, 1994
- [10] GARGI U., OSWALD S., KOSIBA S., DEVADIGA S., KASTURÌ R., Evaluation Of Video Sequence Indexing And Hiererchical Video Indexing, *SPIE Conf. Storage and Retrieval in Image and Video Databases* ,1995
- [11] SWANBERG D., SHU C. F., JAIN R., Knowledge guided parsing in video databases, *SPIE Conf.*, 1993

- [12] BORECZKY J. S., ROWE L. A., Comparison Of Video Shot Boundary Detection Techniques, *IS&T/SPIE Intern. Symposium Electronic Imaging*, 1996
- [13] WANG Y., OSTERMANN J., ZHANG Y. Q., Video processing and communications, *Prentice Hall*, 2001
- [14] <http://trecvid.nist.gov/>
- [15] <http://www.mathworks.com/matlabcentral/>

ÖZGEÇMİŞ

Bariş BORU, 1983 yılında Afyonkarahisar’da doğmuştur. İlk öğrenimini Hacıköy İlköğretim Okulunda, orta öğrenimini Edirne Keşan Anadolu Lisesi’nde tamamlamıştır. Lise öğrenimini Bağcılar Abdurrahman Nermin Bilimli Anadolu Meslek Lisesinde 2001 yılında tamamladıktan sonra Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi bölümü Elektronik Öğretmenliği programına girmiştir. 2005 yılında lisans eğitimini tamamlayarak Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi ana bilim dalında yüksek lisans eğitimine başlamıştır. Halen Aralık 2005’te göreve başladığı Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü’nde araştırma görevlisi olarak çalışmakta ve yüksek lisans öğrenimine devam etmektedir.