

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI**

DOKTORA TEZİ

**HEVC İÇİN HIZLI HAREKET KESTİRİMİ VE MOD SEÇİMİ
YAKLAŞIMLARI**

RAMAZAN DUVAR

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

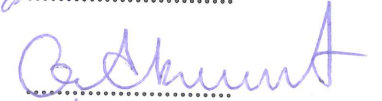
HEVC İÇİN HIZLI HAREKET KESTİRİMİ VE MOD SEÇİMİ
YAKLAŞIMLARI

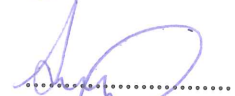
RAMAZAN DUVAR

Prof. Dr. Oğuzhan URHAN
Danışman, Kocaeli Üniversitesi
Doç. Dr. Gökhan Bora ESMER
Jüri Üyesi, Marmara Üniversitesi
Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Aysun TAŞYAPI ÇELEBİ
Jüri Üyesi, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Gökçen ÇETİNEL
Jüri Üyesi, Sakarya Üniversitesi


.....


.....


.....


.....


.....

Tezin Savunulduğu Tarih: 21.01.2019

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, HEVC için hareket kestirimi ve mod seçimi yaklaşımları geliştirilerek HEVC referans yazılımına uygulanmıştır. Yapılan çalışmaların bu alanda çalışanlara yararlı olması dilerim.

Lisans hayatımdan itibaren beraber çalıştığım, çalışmalarımın her aşamasında ilgi ve destek gösteren, kendisinden çok şey öğrendiğim değerli hocam Prof. Dr. Oğuzhan URHAN'a teşekkürlerimi sunarım.

Hayatım boyunca benim için hiçbir fedakarlıktan kaçınmayan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen babam Ahmet DUVAR'a ve annem Hidayet DUVAR'a teşekkür ederim.

Doktora tez izleme komitemde yer alan Dr. Öğretim Üyesi Orhan AKBULUT'a değerli yardım ve katkıları için teşekkür ederim.

Uzun süredir beraber çalıştığım, birçok zorluğu paylaştığım dostum Ayhan KÜÇÜKMANİSA'ya ayrıca teşekkür ederim.

Bu çalışma TÜBİTAK tarafından 2211-C kapsamında desteklenmiştir.

Ocak – 2019

Ramazan DUVAR

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	vi
SİMGELEr DİZİNİ VE KISALTMALAR	viii
ÖZET.....	ix
ABSTRACT	x
GİRİŞ	1
1. VİDEO KODLAMA	3
1.1. Giriş.....	3
1.2. Video Kodlama Temel Döngüsü	4
1.3. Kalite Ölçütleri.....	7
1.4. Video Kodlama Standartları	9
1.4.1. H.261	10
1.4.2. MPEG-1	10
1.4.3. H.262/MPEG-2	11
1.4.4. H.263	11
1.4.5. MPEG-4	11
1.4.6. H.264/AVC (Advanced Video Coding).....	11
2. HEVC	13
2.1. Genel Bakış.....	13
2.2. HEVC’de Ağaç Yapısının Oluşturulmasına Detaylı Bir Bakış	16
2.2.2. Hareket kestirimi	19
2.2.3. Maliyet hesabı	22
2.3. HEVC için Test Konfigürasyonları	23
2.4. HEVC Zaman Analizi.....	24
3. LİTERATÜRDEKİ ÇALIŞMALAR	26
3.1. Hareket Kestirimi	26
3.1.1. Arama noktalarının azaltılması	27
3.1.2. Uyumlama kriterinin basitleştirilmesi.....	30
3.1.3. Bit-derinliğinin azaltılması	30
3.1.4. Tahminsel arama	36
3.1.5. Hiyerarşik arama	37
3.1.6. Hızlı tam arama	37
3.1.7. Karma yöntemler.....	38
3.2. Mod Seçimi	41
4. TEZ ÇALIŞMASI KAPSAMINDA GELİŞTİRİLMİŞ OLAN HAREKET KESTİRİMİ VE MOD SEÇİMİ YÖNTEMLERİ.....	44
4.1. Ekran İçerik Görüntüleri İçin Uyarlamalı Bit Düzlemi Seçme Tabanlı Hareket Kestirimi.....	44
4.1.1. Giriş.....	44

4.1.2. Önerilen yöntem.....	45
4.1.3. Deneysel sonuçlar	47
4.1.4. Sonuçlar.....	51
4.2. HEVC Üzerinde Düşük Bit Derinliği Temelli Hareket Kestirim Yöntemlerinin Başarımlarının Değerlendirilmesi.....	52
4.2.1. Giriş.....	52
4.2.2. Düşük bit derinliği temelli hareket kestirim yöntemlerinin HEVC'ye uygulanması	52
4.2.3. Deneysel sonuçlar	54
4.2.3.1. Düşük bit gösterim temelli yaklaşımların HEVC üzerindeki tam arama hareket kestirim başarımı	54
4.2.3.2. 1BT temelli hareket kestirim yaklaşımlarının HEVC üzerindeki seyrek arama hareket kestirim başarımı	57
4.2.4. Sonuçlar.....	61
4.3. HEVC için Seçmeli Gray Kodlama Temelli Hızlı Hareket Kestirimi	62
4.3.1. Giriş.....	62
4.3.2. Blok hareket kestirimi için verilerin uyumlanması	62
4.3.3. Hızlı tam arama yaklaşımı	64
4.3.4. Seyrek Arama Yaklaşımı	66
4.3.5. Deneysel sonuçlar	67
4.3.6. Sonuçlar.....	69
4.4. HEVC için Blok Benzerliğine Dayalı Mod Seçimi	70
4.4.1. Giriş.....	70
4.4.2. Önerilen Yöntem	70
4.4.2.1. Uygun CU derinliğinin belirlenmesi (CUD)	73
4.4.2.2. Uygun PU modunun belirlenmesi (PUM).....	74
4.4.3. Deneysel sonuçlar	74
4.4.4. Sonuçlar.....	78
5. SONUÇLAR VE ÖNERİLER	79
6. KAYNAKLAR.....	81
KİŞİSEL YAYINLAR VE ESERLER	87
ÖZGEÇMİŞ	89

ŞEKİLLER DİZİNİ

Şekil 1.1.	Temel bir video kodlama döngüsü	5
Şekil 1.2.	Temel bir GOP yapısı.....	7
Şekil 1.3.	Örnek bir bit oranı-PSNR grafiği	8
Şekil 1.4.	Geçmişten günümüze video kodlama standartları	10
Şekil 2.1.	Tipik bir HEVC kodlayıcının blok şeması.....	13
Şekil 2.2.	HEVC'deki hareket kestirimi ile ağaç yapısı oluşturulması işlemlerinin akış şeması	17
Şekil 2.3.	HEVC'de derinlik belirleme ve mod seçimi işlemleri	17
Şekil 2.4.	Örnek PU modları	18
Şekil 2.5.	Örnek bir CTU yapısı.....	18
Şekil 2.6.	BasketballPass dizisi için belirlenmiş olan ağaç yapıları.....	19
Şekil 2.7.	TZS algoritmasının akış şeması	21
Şekil 2.8.	HEVC'de TZS için kullanılan elmas örüntü	21
Şekil 2.9.	RA konfigürasyonu için örnek bir GOP yapısı	24
Şekil 2.10.	LDB konfigürasyonu için örnek bir GOP yapısı.....	24
Şekil 3.1.	Hızlı hareket kestirim yaklaşımları	27
Şekil 3.2.	TDSL için örnek bir arama senaryosu.....	28
Şekil 3.3.	TSS yöntemi için örnek bir arama senaryosu.....	28
Şekil 3.4.	DS'de kullanılan arama örüntüsü.....	29
Şekil 3.5.	DS ile gerçekleştirilmiş örnek bir hareket kestirimi.....	29
Şekil 3.6.	a) "Big Buck Bunny" dizisinden örnek bir görüntü b) (3.2) ile süzgeçlenmiş görüntü c) (3.3) ile dönüşümü gerçekleştirilmiş görüntü	31
Şekil 3.7.	SGC yönteminde önerilmiş olan şablon.....	34
Şekil 3.8.	Big Buck Bunny a) Orjinal çerçeve b) Bloklar için seçilmiş bit düzlemleriyle oluşturulmuş çerçeve c) Bloklar için seçilmiş bit düzlemleri.....	37
Şekil 4.1.	4. 5. 6. ve 7. bit düzlemlerini kullanan şablon	46
Şekil 4.2.	Big Buck Bunny ve Google Maps dizileri için Gray-kodlanmış bit düzlemleri GC7, GC6, GC5, GC4 a) - d)Big Bug Bunny and e) - h)Google Maps.....	47
Şekil 4.3.	a) "Big Buck Bunny" dizisinden alınmış örnek bir görüntü b) Görüntüdeki bloklar için seçilmiş modlar.....	47
Şekil 4.4.	Test dizilerinden alınmış örnek görüntüler a) Big Buck Bunny b) Elephants Dream c) Counter Strike d)Pro Evolution Soccer e) Google Maps	49
Şekil 4.5.	Hareket kestirimi performansı/zaman grafiği	51
Şekil 4.6.	HEVC'deki düşük bit gösterim temelli hareket kestiriminin akış şeması.....	53
Şekil 4.7.	Düşük bit temelli hareket kestirim yöntemlerinin oran- bozunum performansı a) Kimono dizisi b) Kimono dizisi yakından görünüm c) ParkScene dizisi d) ParkScene dizisi	

	yakından görünüm e) PeopleOnStreet dizisi f) PeopleOnStreet dizisi yakından görünüm	57
Şekil 4.8.	1BT temelli seyrek arama yöntemlerinin oran-bozunum performansı a) Kimono dizisi b) Kimono dizisi yakından görünüm c) ParkScene dizisi d) ParkScene dizisi yakından görünüm e) PeopleOnStreet dizisi f) PeopleOnStreet dizisi yakından görünüm.....	60
Şekil 4.9.	HEVC referans yazılımı üzerinde piksel gruplama işlemi.....	63
Şekil 4.10.	Örnek bir piksel gruplama işlemi	64
Şekil 4.11.	Referans çerçevedeki verilerin dönüştürülmesi	65
Şekil 4.12.	Seyrek arama yaklaşımının blok şeması	67
Şekil 4.13.	Taramalı arama işlemi için örnek bir şablon.....	67
Şekil 4.14.	BasketballPass dizisinin bir çerçevesi için kodlayıcı tarafından oluşturulmuş ağaç yapısı	71
Şekil 4.15.	BasketballPass dizisinden alınmış örnek ağaç yapıları	71
Şekil 4.16.	İki parça içeren PU modları	72
Şekil 4.17.	Geliştirilmiş olan algoritmanın blok şeması.....	73
Şekil 4.18.	BasketballPass dizisinden alınmış örnek ağaç yapıları için S değerleri.....	74
Şekil 4.19.	CUD için eşik değerinin belirlenmesi a)Kimono b)ParkScene.....	75
Şekil 4.20.	PUM için eşik değerinin belirlenmesi a)Kimono b)ParkScene	76
Şekil 4.21.	CUD yöntemi için elde edilmiş olan $\Delta B-\Delta T$ grafiği a)ParkScene b)KristenAndSara.....	77

TABLolar DİZİNİ

Tablo 2.1. Performans değeriendirilmesinde kullanılan görüntü dizileri ve özellikleri.....	25
Tablo 2.2. HEVC zaman analizi.....	25
Tablo 4.1. Önerilen yöntemin literatürdeki düşük bit gösterimi temelli yöntemler karşılaştırılması (PSNR)	49
Tablo 4.2. Önerilen yöntemin ve literatürdeki yöntemlerin piksel başına düşen işlem yükleri.....	50
Tablo 4.3. Temel yapılandırma parametreleri	54
Tablo 4.4. Farklı QP'lerde düşük bit temelli hareket kestirim yöntemlerinin Kimono dizisi için ortalama PSNR (dB) ve bit-oran değeri.....	55
Tablo 4.5. Farklı QP'lerde düşük bit temelli hareket kestirim yöntemlerinin ParkScene dizisi için ortalama PSNR (dB) ve bit-oran değeri	55
Tablo 4.6. Farklı QP'lerde düşük bit temelli hareket kestirim yöntemlerinin PeopleOnStreet dizisi için ortalama PSNR (dB) ve bit-oran değeri	55
Tablo 4.7. SGC, 1BT ve C-1BT yöntemlerinin HM ile karşılaştırılması (LDB)	56
Tablo 4.8. Tam arama temelli 1BT, C1BT ve SGC yöntemlerinin HEVC üzerindeki çalışma zamanları (saat)	56
Tablo 4.9. Farklı QP'lerde bir bit temelli hareket kestirim yöntemlerinin Kimono dizisi için ortalama PSNR (dB) ve bit-oran değeri	58
Tablo 4.10. Farklı QP'lerde bir bit temelli hareket kestirim yöntemlerinin ParkScene dizisi için ortalama PSNR (dB) ve bit-oran değeri.....	58
Tablo 4.11. Farklı QP'lerde bir bit temelli hareket kestirim yöntemlerinin PeopleOnStreet dizisi için ortalama PSNR (dB) ve bit-oran değeri.....	59
Tablo 4.12. SGC, 1BT ve C-1BT yöntemlerinin HM ile karşılaştırılması (LDB)	59
Tablo 4.13. 1BT temelli seyrek arama yöntemlerinin HEVC üzerindeki çalışma süreleri (saat) (QP = 27).....	61
Tablo 4.14. Geliştirilmiş olan hızlı tam arama yaklaşımı ile HM'deki tam arama yaklaşımının sonuçlarının karşılaştırılması	68
Tablo 4.15. Geliştirilmiş olan seyrek arama yaklaşımı ile HM'deki TZS yaklaşımının sonuçlarının karşılaştırılması.....	68
Tablo 4.16. Geliştirilmiş olan hızlı tam arama yaklaşımı ile seyrek arama yaklaşımının sonuçlarının karşılaştırılması.....	69
Tablo 4.17. CUD yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB).....	76

Tablo 4.18. PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB).....	77
Tablo 4.19. CUD + PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB).....	77
Tablo 4.20. CUD + PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (RA)	77



SİMGELER DİZİNİ VE KISALTMALAR

Kısaltmalar

HEVC	: High Efficiency Video Coding (Yüksek Verimli Video Kodlama)
1BT	: 1 Bit Transform (1 Bit Dönüşümü)
C-1BT	: Constrained 1 Bit Transform (Kısıtlanmış 1 Bit Dönüşümü)
SGC	: Selective Gray Coded (Seçmeli Gray Kodlanmış)
SIMD	: Single Instruction, Multiple Data (Tekli Komut, Çoklu Veri)
SAD	: Sum of Absolute Differences (Mutlak Farklar Toplamı)
VCEG	: Video Coding Experts Group (Video Kodlama Uzmanları Grubu)
MPEG	: Moving Pictures Experts Group (Hareketli Resimler Uzmanları Grubu)
JCT-VC	: Joint Collaborative Team on Video Coding (Video Kodlama İşbirliği Takımı)
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
GOP	: Group of Pictures (Resim Grubu)
QP	: Quantization parameter (Kuantalama Katsayısı)
CTU	: Coding Tree Unit (Kodlama Ağaç Ünitesi)
CU	: Coding Unit (Kodlama Ünitesi)
PU	: Prediction Unit (Kestirim Ünitesi)
HK	: Hareket Kestirimi
TZS	: Test Zone Search (Bölgesel Test Araması)
PSNR	: Peak Signal to Noise Ratio (Tepe İşaret Gürültü Oranı)
NNMP	: Number of Non-Matching Points (Uyumsuz Nokta Sayısı)
LD	: Low Delay (Düşük Gecikme)
RA	: Random Access (Rastgele Erişim)

HEVC İÇİN HIZLI HAREKET KESTİRİMİ VE MOD SEÇİMİ YAKLAŞIMLARI

ÖZET

Günümüzde, sayısal video kaydedebilen veya oynatabilen cihazlara olan talebin yoğun bir şekilde artmasıyla birlikte sayısal videoların etkin bir şekilde saklanması ve iletilmesinin önemi de artmaktadır. Bu doğrultuda verimli bir sıkıştırma sağlayan ve hesapsal karmaşıklığı az olan yöntemlere ihtiyaç duyulmaktadır. Bu tez çalışmasının odak noktası da en güncel video kodlama standardı olan HEVC için etkin hareket kestirimi ve mod seçimi yöntemleri geliştirilmesidir.

Bu tez çalışması kapsamında ekran içerik görüntüleri için düşük bit gösterimi temelli bir hareket kestirim yöntemi geliştirilmiş, literatürdeki düşük bit gösterimi temelli hareket kestirim yöntemleri HEVC referans yazılımına uygulanmış ve bu yöntemler için hızlı tam arama ve seyrek arama yaklaşımları geliştirilmiştir. Bununla birlikte, HEVC'de önemli bir işlem yüküne sahip mod seçim işlemlerini hızlandıran blok benzerliğine dayalı bir yöntem önerilmiştir.

DeneySEL sonuçlar, tez kapsamında önerilen yöntemlerin literatürdeki yöntemlere göre daha iyi bit oran/bozunum ve hız dengesi sağladığını göstermektedir.

Anahtar Kelimeler: Hareket Kestirimi, HEVC, Mod Seçimi, Video Kodlama.

FAST MOTION ESTIMATION AND MODE DECISION APPROACHES FOR HEVC

ABSTRACT

Today, with the increasing demand for devices capable of recording or playing digital video, the importance of effective storage and transmission of digital videos is also increasing. In this respect, there is a need for methods that provide an efficient compression and low computational complexity. The focus of this thesis is to develop motion estimation and mode decision methods for the most recent video coding standard HEVC.

Within the scope of this thesis, a motion estimation method based on low bit representation for screen content images was developed. Low bit representation based motion estimation methods in the literature were applied to HEVC reference software and fast full search and sparse search methods were developed for these methods. Additionally, a method based on block similarity has been proposed to accelerate the mode decision process in HEVC.

Experimental results show that the methods proposed in the thesis provide better bit rate/distortion and speed balance compared to the methods in the literature.

Keywords: Motion Estimation, HEVC, Mode Decision, Video Coding.

GİRİŞ

Sayısal bir video birbirini takip eden görüntü çerçevelerinin bir araya gelmesiyle oluşturulmuş ayrık zamanlı bir veri olarak tanımlanabilir. Teknolojideki ilerlemelerle birlikte günümüzde akıllı telefon, tablet, bilgisayar, kamera, televizyon vb. sayısal video kaydedebilen veya oynatabilen cihazlara olan talep de yoğun bir şekilde artmaktadır.

Sayısal video işaretlerinin ham olarak depolanması durumunda yüksek hafıza birimlerine, gerçek zamanlı olarak iletilmesi durumunda ise çok yüksek bant genişliğine ihtiyaç duyulmaktadır. Bu sebepten dolayı, genel bir ifadeyle, video işaretleri içindeki birbirine benzeyen ve artıklık adı verilen uzamsal, zamansal ve istatistiksel bilgilerden faydalanılarak sıkıştırılmaktadır.

Video sıkıştırma/kodlama işlemleri için standartları belirlemek amacıyla uzun yıllardır çeşitli çalışmalar gerçekleştirilmektedir. Bu amaçla ITU-T tarafından kurulan VCEG (Video Coding Experts Group) tarafından 1984 yılında ilk sayısal video kodlama standartlarından biri olan H.261 [1] ilan edilmiştir. ISO/IEC tarafından ise MPEG (Moving Pictures Experts Group) kurulmuş ve 1993 yılında MPEG-1 [2] duyurulmuştur. Günümüze kadar başka grupların da destekleriyle çeşitli video kodlama standartları geliştirilmiştir. 2013 yılında ise VCEG ve MPEG'in bir araya gelmesiyle oluşturulan JCT-VC (Joint Collaborative Team on Video Coding) tarafından HEVC (High Efficiency Video Coding) [3] standardı ilan edilmiştir. Günümüzde kullanılan en güncel video kodlama standardı HEVC olmakla birlikte MPEG tarafından FVC (Future Video Coding) isimli yeni bir kodlama standardı ile ilgili çeşitli çalışmalara başlanmıştır.

Bu tez çalışmasının odak noktaları mevcut kodlama verimliği korunarak hesapsal karmaşıklığı düşüren yöntemlerin önerilmesi ve mevcut hesapsal karmaşıklık seviyesinde kalınarak kodlama verimliliğini arttıran yeni yaklaşımların önerilmesidir. Bu çerçevede bu tez çalışmasında, uygun bir bit-oran /bozunum ve hız dengesi sağlayan çeşitli hareket kestirimi ve mod seçimi algoritmaları geliştirilmiştir. Bu

algoritmalar hem MATLAB™, hem de en güncel video kodlama standardı olan HEVC'nin referans yazılımı HM16.7 üzerinde gerçekleştirilmiştir. Bu tez çalışmasının literatüre olan özgün katkıları aşağıda verilmiştir:

- Ekran içerik görüntüleri için tasarlanmış düşük bit-derinliği temelli özgün bir hareket kestirim yöntem geliştirilmesi,
- Bir-bit dönüşümü (1BT) [4], kısıtlanmış bir-bit dönüşümü (C-1BT) [5] ve seçici gray kodlama (SGC) [6] temelli hareket kestirimi yöntemlerinin ilk defa HEVC referans yazılımına uygulanması ve başarımlarının değerlendirilmesi,
- HEVC referans yazılımı üzerinde en iyi başarımları veren düşük bir gösterimi temelli yöntem olan SGC yöntemini temel alan ve SIMD (Single instruction, multiple data) kullanarak paralel işlem yapan yeni ve özgün bir hızlı hareket kestirimi algoritması geliştirilmesi ve yine HEVC referans yazılımı üzerinde uygulanması,
- Blok benzerliğine/homojenliğe dayalı bir özgün bir mod seçimi yaklaşımının geliştirilmesi ve HEVC referans yazılımı üzerinde uygulanması.

Bu tez çalışmasının birinci bölümünde genel bir video kodlayıcı şeması ile birlikte video kodlamadaki temel kavramlar ve geçmişten günümüze video kodlama standartları ile ilgili bilgiler verilmiştir.

İkinci bölümde HEVC standardı ile bilgiler verilmiştir. Bu bölümde özellikle HEVC'nin önceki video kodlama standartlarından farkları üzerinde durulmuş, HEVC'deki hareket kestirimi ve mod seçimi işlemleri irdelenmiştir.

Sonraki bölümde literatürdeki hareket kestirimi ve mod seçimi yöntemleri hakkında bilgiler verilmiş, yöntemler ayrıntılı bir şekilde incelenmiştir. Dördüncü bölümde ise bu tez çalışması kapsamında geliştirilmiş olan yöntemler ve yapılmış olan uygulamalar detaylı bir şekilde anlatılmıştır.

Bu tez çalışmasının beşinci ve son bölümünde ise tez kapsamında yapılmış olan çalışmalar hakkında genel bir değerlendirme yapılmış olup, gelecekte yapılabilecek olan çalışmalar hakkında öneriler sunulmuştur.

1. VİDEO KODLAMA

1.1. Giriş

Sayısal bir video işareti birbirini takip eden görüntü çerçevelerinden oluşmaktadır. Her bir pikselin 24 bitle ifade edildiği, 1920×1080 (Full HD) piksel çerçeve boyutuna sahip ham bir görüntüyü kaydedebilmek için bellekte yaklaşık 5.93 MB alana ihtiyaç duyulmaktadır. Aynı çözünürlük altında, 24 çerçeve/saniye değeri ile oluşturulmuş bir görüntü dizisi ele alındığında, saniyede kaydedilmesi/iletilmesi gereken veri miktarı yaklaşık 142.32 MB'dır. Bu verinin iletimi için ise yaklaşık 1138 Mbps (yaklaşık 1Gbps) veri aktarım hızı sağlayabilecek bir kanal gerekmektedir. Dünya genelinde ortalama internet hızının birkaç 10 Mbps olduğu düşünüldüğünde, bu tip bir verinin gerçek zamanlı olarak iletilemeyeceği açıktır. Bu örnek üzerinden yola çıkıldığında, video işaretlerinin ham veri olarak depolanmasının ve iletilmesinin mümkün olmadığı açıkça görülmektedir. Bu problemlerin önüne geçebilmek için video işaretleri sıkıştırılarak boyutları azaltılmaktadır. Bu bağlamda, video kodlama, video işaretinin çeşitli özelliklerinden faydalanılarak işaretin sıkıştırılmasına karşılık gelmektedir. Tipik bir video kodlayıcıda, kayıplı sıkıştırma algoritmalarının kullanıldığı hibrit bir yaklaşım benimsenmektedir. Video kodlamada görsel kaliteden olabildiğince ödün vermeyecek şekilde bir sıkıştırma oranı seçilmesi birçok uygulamada tercih edilmektedir.

Video işaretlerinde birbirine benzeyen çok fazla miktarda veri bulunmaktadır. Birbirlerine benzeyen bu veriler artıklık olarak isimlendirilmektedir. Video sıkıştırma işlemlerindeki temel ölçüt artıklık verilerinin ortaya çıkarılması ve bu artıklıkların giderilmesidir. Video işaretindeki artıklık tipleri:

- Algısal,
- Uzamsal,
- Zamansal ve
- İstatistiksel olarak gruplanabilmektedir.

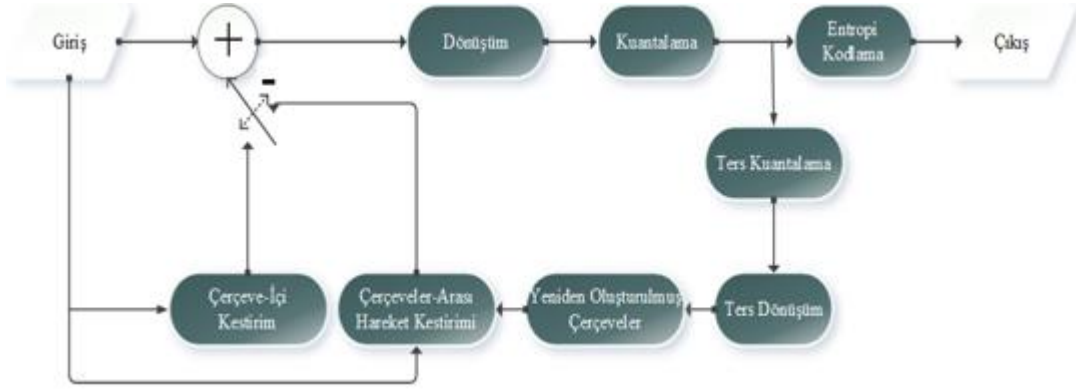
Algısal artıklıklara en belirgin örnek olarak renk uzayları verilebilir. İnsan gözü ışıklılık bileşenlerine renk bileşenlerine göre daha duyarlı olduğu için RGB uzayı yerine YCrCb uzayı kullanılması ve renk farkı bileşenlerinin (CrCb) daha düşük çözünürlükle kodlanması algısal bir artıklıktan faydalanmak içindir. Diğer yandan, uzamsal artıklıklar t anındaki bir çerçeve içinde bulunan ve birbirine benzeyen veriler olarak düşünülebilir. Uzamsal artıklık gideriminde çerçeve-içi kestirim yöntemleri ve dönüşüm kodlaması (transform coding) kullanılmaktadır. Zamansal artıklıklar ardışık görüntü çerçeveleri arasında benzerlik nedeniyle ortaya çıkmaktadır. Zamansal artıklığı gidermek için ise piksel/blok tabanlı hareket kestirimi yöntemleri kullanılmaktadır. Son olarak, istatistiksel artıklıklar ise verilerin entropisine bağlı olarak gerçekleştirilen kayıpsız kodlama yöntemleriyle giderilmektedir.

Zamansal artıklığın giderilmesi, video kodlamanın kodlama verimliliği ve hesapsal karmaşıklığı açısından önemli bir aşamasını oluşturmaktadır. Örneğin, HEVC standartının referans yazılımı olan HM'de, kullanılan görüntü dizilerine bağlı olmakla birlikte kodlama süresinin ortalama %70'i zamansal artıklığı gidermek için kullanılmış olan işlemlerde harcanmaktadır [7].

1.2. Video Kodlama Temel Döngüsü

Bugüne kadar geliştirilmiş olan her video kodlama standardı önceki standartlara göre yeni özelliklere sahip olmakla birlikte, tüm video kodlayıcıların temel yapıları birbirine benzemektedir. Şekil 1.1'de temel bir video kodlama döngüsü verilmiştir.

Aslında video kodlama standartları saklanacak veya gönderilecek olan verinin yapısının belirlenmesini amaçlamaktadır. Bununla birlikte kestirim, dönüşüm ve kodlama işlemleri geliştiricilerin insifiyatine bırakılmış durumdadır. Fakat referans yazılımlarda bu işlemler için de öneriler sunulmaktadır.



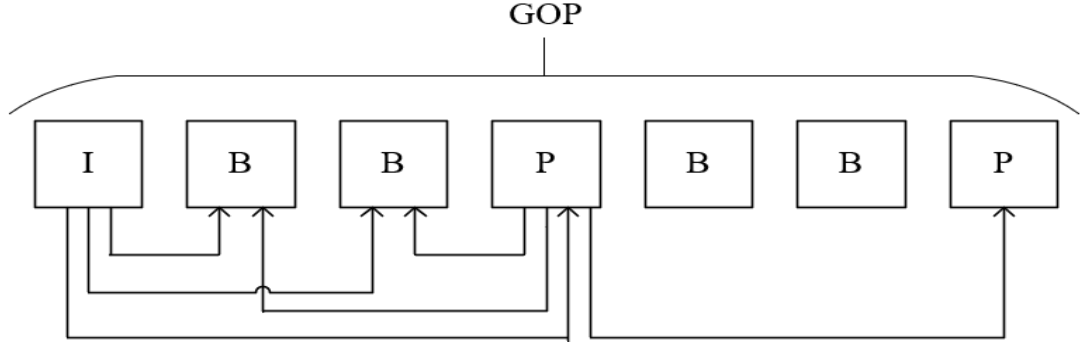
Şekil 1.1. Temel bir video kodlama döngüsü

Kestirim işlemleri genellikle uzamsal ve zamansal artıklıkları gidermek için gerçekleştirilmektedir. Uzamsal artıklıkların giderilmesi, genelde çerçeve-içi kestirim yöntemleri kullanıldıktan sonra JPEG’de (Joint Photographic Experts Group) kullanılan DCT veya benzeri yöntemlerle dönüşüm kodlama işleminin yapılmasıdır. Zamansal artıklıkların giderimi için ise çerçeveler-arası hareket kestirim yaklaşımları kullanılmaktadır. Bu yaklaşımlarda birçok standart ve uygulamada kodlanacak çerçeve bloklara bölünmekte ve bu bloklar ilgili zamanlardaki çerçevelerde aranarak, belirli değerlendirme kriterlerine göre en yakın olan blok bulunmaktadır. Bu işlem hareket kestirimi olarak adlandırılmaktadır. Video kodlayıcılarda genellikle blokların benzerliğini değerlendirmek üzere uyumlama kriteri olarak genellikle mutlak farkların toplamı (SAD - sum of absolute differences) yaklaşımı kullanılmaktadır. Bu tez çalışmasındaki odak noktalarından birisi de farklı bir uyumlama kriteri kullanmak olmakla birlikte, çerçeveler arası hareket kestirimi ile ilgili detaylardan sonraki bölümlerde bahsedilecektir. Aranılan bloğa en uyumlu blok bulunduğundan sonra ise hareket vektörleri kaydedilip, aranılan bloğa en yakın blokla aranılan blok arasındaki fark alınarak bu fark yine DCT veya benzeri bir yaklaşımla dönüşüm kodlamasına tabii tutulmaktadır. Çerçeve-içi ve çerçeveler arası kestirim sırasında genellikle YCrCb uzayında ışıklılığa karşılık gelen Y kanalı kullanılmaktadır. Çerçeveler arası hareket kestirimi işlemlerinde kodlayıcılar daha önce kodlamış oldukları çerçeveleri ters işlemleri yaparak yeniden oluşturmakta ve kullanılmaktadır. Kodlanmış olan video verisi çözücüde benzer şekilde geri oluşturulacağından hareket kestirimi aşamasında giriş çerçevesindeki veri yeniden oluşturulmuş çerçeveler içinde aranmaktadır. Dönüşüm işlemlerinde ise genellikle DCT veya benzeri bir dönüşüm kullanılarak veriler kuantalama işlemi için uygun bir hâle getirilmektedir. Kuantalama işlemi

sırasında da yüksek frekans vb. insan gözünün farketmeyeceği algısal artıklıkların giderilmesi işlemleri gerçekleştirilmektedir. Son aşamada ise hareket vektörleri ve dönüştürülmüş olan veriler sembol kodlanarak saklanacak/iletilecek veriye son hali verilmektedir. Kodlama işlemleri sırasında da istatistiksel artıklıkları gidermek amacıyla Huffman kodlama, aritmetik kodlama benzeri yaklaşımlar kullanılabilir.

Video kodlamada genellikle üç farklı çerçeve türü kullanılır. I çerçevesi (intra-frame), diğer çerçevelerden bağımsız olarak çerçevenin kendi içinde kodlanmasıyla oluşturulur. Bu kodlama çerçevenin genellikle JPEG gibi kodlanmasıyla gerçekleştirilir. Geleneksel video kodlama uygulamalarında görüntü kalitesini arttırmak ve diğer çerçevelerden bağımsız bir referans oluşturmak amacıyla belirli periyotlarla I çerçevesi kodlama işlemi gerçekleştirilebilmektedir. P çerçeveleri (predictive coded frame) çerçeveler-arası hareket kestirim yöntemleriyle kendilerinden önceki I veya P çerçeveleri kullanılarak kodlanmaktadır. B çerçeveleri de yine çerçeveler-arası hareket kestirim yöntemleriyle kendilerinden önceki veya sonraki I ve P çerçeveleri kullanılarak kodlanmaktadır. B çerçevelerinin birden fazla çerçeve kullanılarak kodlanıyor olması işlem yükünü arttırmakla birlikte, eşleme için daha uygun bir blok bulunmasını kolaylaştırmakta ve bu sayede blok eşlendikten sonra da yapılacak artıklık kodlama işlemleri için de avantaj sağlamaktadır. Bu durum da iletilecek veya saklanacak olan video dizisinin daha az bit ile temsil edilmesini sağlayarak, verimli bir kodlama işleminin gerçekleştirilmesinin yolunu açmaktadır. Fakat video kodlayıcıların bu vb. durumlarda hız ve kodlama verimliliği arasında bir seçim yapması gerekmektedir. Video kodlayıcılar hız ve sıkıştırılacak veri miktarı arasındaki dengeyi belirleyen karar mekanizmaları sayesinde bu tarz kararları verme yeteneklerine sahiptir. Bunlarla birlikte kestirim işlemleri blok bazlı yapıldığı için P veya B çerçeveleri I bloklar da içerebilir. Bu durumda yine video kodlayıcıların karar mekanizmaları devreye girmektedir [8].

I, P ve B çerçeve türlerinin çeşitlik kombinasyonlarda bir araya getirilmesiyle oluşturulan yapılara ise GOP (Group of pictures – resim grubu) ismi verilmektedir. Şekil 1.2’de örnek bir GOP yapısı verilmiştir. Geleneksel bir video kodlayıcı yapısında bloklar dilimleri, dilimler resimleri, resimler GOP’leri ve GOP’ler ise video dizisini oluşturmaktadır.



Şekil 1.2. Temel bir GOP yapısı

1.3. Kalite Ölçütleri

Video sıkıştırma işlemlerinde verimli bir sıkıştırma oranı elde edilmeli ve sıkıştırıldıktan sonra yeniden oluşturulan görüntüdeki bozulma etkisi hız/kalite ödünleşimi (trade-off) de dikkate alınarak mümkün olduğu aza indirilmiş olmalıdır.

Video kodlama uygulamalarında kaliteyi ölçmek için genellikle PSNR (Peak Signal to Noise Ratio – Tepe İşaret Gürültü Oranı) metriği kullanılmaktadır. PSNR değeri giriş çerçevesi ve yeniden oluşturulan çerçeve arasındaki MSE (Mean Square Error - Ortalama Kareysel Hata) değerinin logaritması alınarak hesaplanmaktadır. Eşitlik (1.1)'de MSE hesabı, Eşitlik (1.2)'de ise PSNR hesabı verilmiştir.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (I_g(i,j) - I_\varphi(i,j))^2 \quad (1.1)$$

$$PSNR = 10 \times \log_{10} \left(\frac{R^2}{MSE} \right) \quad (1.2)$$

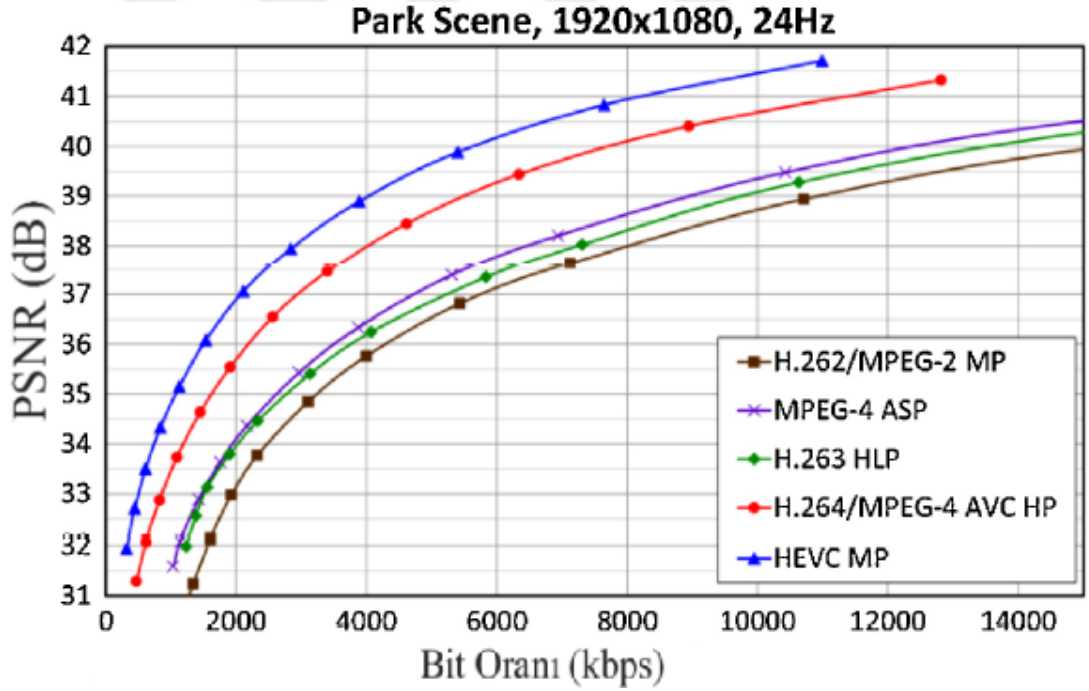
Eşitlik (1.1)'de H ve W değerleri çerçeve yükseklik ve genişliğini, I_g giriş çerçevesini, I_φ ise çıkış çerçevesini (sıkıştırılmış görüntüyü) temsil etmektedir. R değeri ise bir pikselin kaç bit ile ifade edildiğine bağlı olarak Eşitlik (1.3)'teki gibi hesaplanmaktadır.

$$R = 2^n - 1 \quad (1.3)$$

Bu denklemde n değeri bit sayısını ifade etmektedir. Örneğin 8 bit piksel değerlerinin kullanıldığı bir görüntü için R değeri 255 olarak hesaplanmaktadır. PSNR değerinin

yüksek olması çıkış görüntüsü ile giriş görüntüsü arasındaki benzerliğin fazla olduğunu, düşük olması ise benzerliğin az olduğu anlamına gelmektedir.

PSNR metriği iki görüntünün benzerliğinin karşılaştırılması sırasında mantıklı sonuçlar vermesine rağmen video kodlama işlemlerinde performans değerlendirmesi için genellikle tek başına yeterli değildir. Çünkü video kodlama işlemlerinde uygulamaya bağlı olarak işlem süresi veya bit oranı da önemli olabilmektedir. Örneğin bant genişliğinin sınırlı olduğu bir uygulama için düşük bir bit oranında görece olarak iyi PSNR değerleri gerekli olabilmektedir. Bu karşılaştırmaları yapmak amacıyla farklı bit oranlarındaki PSNR değerleri grafik üzerinde de incelenebilmektedir. Şekil 1.3'te farklı video kodlama standartları için elde edilen örnek bir bit oranı-PSNR grafiği verilmiştir. Bu grafik aynı zamanda literatürde oran-bozunum (rate-distortion) grafiği olarak da isimlendirilmektedir.



Şekil 1.3. Örnek bir bit oranı-PSNR grafiği [9]

Bu grafikler incelenirken aynı bit oranında elde edilen PSNR değeri veya aynı PSNR değerinde elde bit oranı dikkate alınmaktadır. Örneğin bu grafikte 6000 kbps değeri incelendiğinde en iyi başarıyı (en yüksek PSNR değerini) HEVC standartının verdiği gözlenmektedir. Ya da 39 dB değerini en düşük bit oranı ile yine HEVC standartının verdiği gözlenmektedir. Fakat bu vb. durumlarda eğer önemliyse uygulamaların

çalışma hızı da dikkate alınmalıdır. Uygulamada istenen kodlama süresine göre bazı durumlarda makul başarı oranı sağlayan yaklaşımlar da kullanılabilir.

Grafik temelli karşılaştırmalarda yöntemin genel performansı hakkında bir fikir edinilebilmesine karşın sayısal bir değer olmadığı için karşılaştırma işleminin sağlıklı olmadığı durumlar gözlenebilmektedir. Bu nedenle son yıllarda ağırlıklı olarak [10]'da önerilen metrik kullanılmaktadır. Bu metrikte temel amaç, yöntemler farklı kuantalama katsayılarıyla (Quantization parameter - QP) çalıştırdıktan sonra, PSNR – bit oranı grafiklerini oluşturmak ve eğri uydurma yöntemleri kullanarak eğrilerin arasında alanı integral olarak ölçmektir. Bu yöntemde elde edilen metrik ΔB olarak isimlendirilmektedir. Bu sayede farklı kuantalama değerleri veya farklı hedef bit oranları ile çalıştırılan kodlayıcıların performansının nesnel bir şekilde değerlendirilmesi hedeflenmektedir.

Kodlama amaçlı önerilen bir yöntemin zaman farkını hesaplamak için ise genellikle Eşitlik (1.4) kullanılmaktadır.

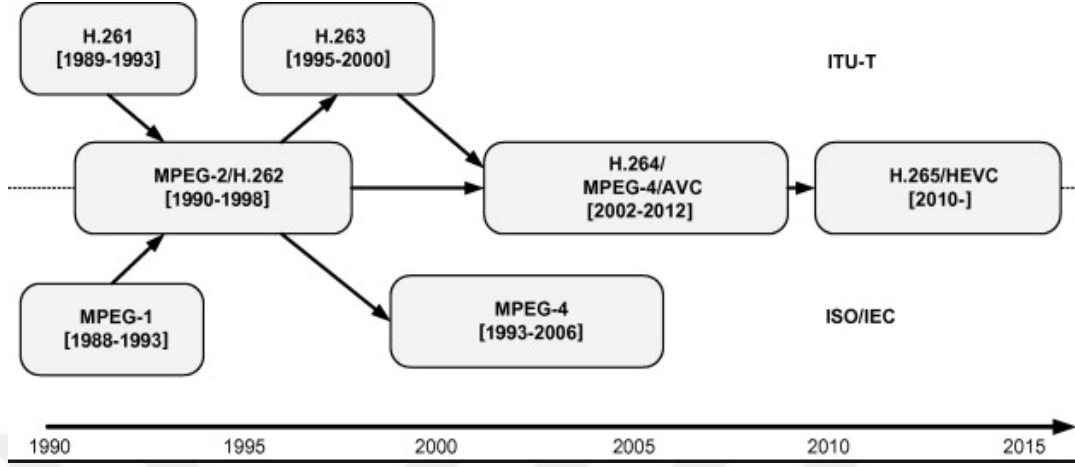
$$\Delta T = \frac{T_{\text{referans}} - T_{\text{önerilen}}}{T_{\text{referans}}} \times 100 \quad (1.4)$$

Bu eşitlikte T_{referans} referans olarak alınan yöntemin kodlama süresini, $T_{\text{önerilen}}$ ise önerilen yöntemin kodlama süresini ifade etmektedir. Yine farklı kuantalama değerleri veya hedef bit oranları için kodlayıcı çalıştırılarak süre değerlerinin ortalaması alınmaktadır. Örneğin HEVC'de genellikle 22, 27, 32 ve 37 kuantalama katsayıları için bu işlemler gerçekleştirilerek tek bir ΔB ve ΔT değeri elde edilir. ΔB ve ΔT değerleri birbirine bölünerek de kodlayıcıların veya yöntemlerin birbirleri arasında performans değerlendirmeleri de gerçekleştirilebilmektedir.

1.4. Video Kodlama Standartları

Video kodlama işlemlerini gerçekleştirmek, bu konudaki standartları belirlemek amacıyla uzun zamandır çeşitli çalışmalar gerçekleştirilmektedir. Bu amaçla VCEG tarafından H.261, H.263 vb. standartları, MPEG tarafından ise MPEG-1 ve MPEG-4 standartları geliştirilmiştir. Bu gruplar arasında işbirlikleri sayesinde de H.262/MPEG-2, H.264/AVC ve HEVC/H.265 standartları geliştirilmiş olup,

üzerlerinde çalışmalar yapılmıştır. Şekil 1.4'te geçmişten günümüze video kodlama standartları kronolojik bir şekilde verilmiştir.



Şekil 1.4. Geçmişten günümüze video kodlama standartları [11]

1.4.1. H.261

VCEG tarafından geliştirilen ve bloklarla çerçeveler-arası hareket kestiriminin ilk defa kullanıldığı video kodlayıcı olan H.261, 1990 yılında duyurulmuştur. H.261 video konferans görüşmeleri için tasarlanmıştır. H.261'de blok tabanlı hibrit bir video kodlama yaklaşımı kullanılmıştır. Video kodlamada temel birimlerden biri olarak değerlendirilen makroblok (MB) kavramı ilk olarak H.261 ile ortaya çıkmıştır. Zamansal artıklığı azaltmak amacıyla hareket kestirimi ile hareket vektörlerinin hesaplanarak kodlanması da H.261'in önemli özelliklerinden birisidir. H.261 standartında I ve P çerçeveleri kodlanmaktadır [12].

1.4.2. MPEG-1

MPEG tarafından geliştirilen MPEG-1 standardı H.261 standartını temel almakla birlikte yeni ve güçlü özellikler içermektedir. Ağırlıklı olarak CD-ROM'lar üzerinde video depolamak için kullanılmıştır. Temel kodlama birimi makrobloktur. Bu standartta I ve P çerçevelerine ek olarak B çerçeveleri de kodlanmaktadır. I, P ve B çerçeveleriyle GOP yapısı ilk olarak bu standartta oluşturulmuştur. Kendisinden önce veya sonra gelen I veya P çerçevelerinden kodlanabilen B çerçeveleri sayesinde daha yüksek sıkıştırma oranları elde edilebilmektedir. Bunlarla birlikte bu standartta ses

sıkıştırma işlemleri de gerçekleştirilmiştir. Çerçeve-içi kestirim ve artıklık kodlama için ise DCT işleminden yararlanılmıştır [12].

1.4.3. H.262/MPEG-2

MPEG ve VCEG gruplarının ortak çalışması sonucunda ortaya çıkan MPEG-2 ise temel olarak MPEG-1'e benzemekle birlikte geçmeli tarama, ölçeklenebilirlik, hata esnekliği ve yüksek bit hızlarında çalışma özelliklerine sahiptir. En yaygın ve önemli kullanım alanlarından biri ise sayısal televizyon yayıncılığı olmuştur. Bununla birlikte DVD üzerinde video depolamak için ağırlık olarak kullanılan bir standart haline de gelmiştir [13].

1.4.4. H.263

H.263 standardı da yine önceki standartlarla benzer özellikler göstermekle birlikte özellikle hareket kestiriminde alt piksel hareket vektörlerinin eklenmesi önemli bir yenilik olarak öne çıkmaktadır. Aynı çözünürlükte bir video dizisi için MPEG-2'ye göre yarı yarıya bit kazancı sağlayabilen H.263 özellikle internet üzerinde birçok uygulamada kullanılmıştır [14].

1.4.5. MPEG-4

MPEG-4 standardı da geleneksel blok tabanlı işlemleri desteklemekle birlikte içerik tabanlı kodlama yapılabilmesine de olanak sağlamaktadır. Sahne içerisinde yer alan içerikler arka planla birlikte video nesnesi olarak değerlendirilmektedir. Bu sayede kodlama işlemlerinde önemli esneklikler sağlanabilmektedir. Bu durumun en önemli örneklerinden birisi video nesnelerinde farklı kodlama işlemleri gerçekleştirilerek sıkıştırma etkinliği artırılabilmesidir. Bu vb. özellikleri sayesinde MPEG-4 çoklu ortam uygulamalarını desteklemekte, doğal ve yapay görüntülerin birbirlerine eklenebilmesi ve kodlanabilmesini sağlamaktadır. İnternet ve özellikle mobil uygulamalarda kullanılan bir video kodlama standardı olmayı hedeflemiştir [15].

1.4.6. H.264/AVC (Advanced video coding)

VCEG ve MPEG grupları tarafından birlikte geliştirilen yeni bir standart olan H.264 ise yine önceki standartlar gibi makroblok yapısını kullanıyor olmasına rağmen

çerçeveler arası kodlamadaki esnekliğiyle ön plana çıkmaktadır. Hareket kestiriminde makrobloklar kare ve dikdörtgenler şeklinde bölünebilir. Her bir makro 16×16 , 16×8 , 8×16 ve 8×8 blok boyutlarına bölünebilirken 8×8 'lik alt bloklar ise yine alt bloklara bölünebilmektedir. Uygun blok boyutlarının seçilmesindeki bu esneklik işlem yükünü arttırmasına rağmen kodlama başarımını arttırmaktadır. Bununla birlikte H.264 çoklu referans resim seçimini ve çeyrek alt piksel doğruluğa kadar hareket kestirimini desteklemektedir. Çoklu referans çerçeve seçimiyle özellikle sahne geçişlerinde performans artışı sağlanmaktadır. Yine bu özellikler de hesapsal karmaşıklığı arttırmasına rağmen kodlama başarımını arttırmaktadır [16].

Makroblokların bir araya gelmesi ile dilimler oluşturulmaktadır. Dilim grupları bağımsız olarak kodlanabilir ve değişik şekillerde gruplanabilir. Dilimlerin içindeki ilk makroblok dilimdeki herhangi bir makroblok olabilir. Bununla birlikte gereksiz dilimler belirlenerek o dilim için kodlama işlemi gerçekleştirilmeyebilir [8]. Hatalı olan dilimler de çıkarılarak çerçeve genelinin hatadan etkilenmemesi de sağlanabilir.

H.264'te önceki standartlardan farklı olarak dönüşüm kodlamasında DCT dönüşümü yerine tam sayı dönüşüm kodlaması yapılmaktadır. Bu sayede kodlama işlemi daha hızlı gerçekleştirilebilmektedir. Ayrıca tam sayılarla işlem yapılıyor olması da donanımsal gerçeklemeler için avantaj sağlamaktadır.

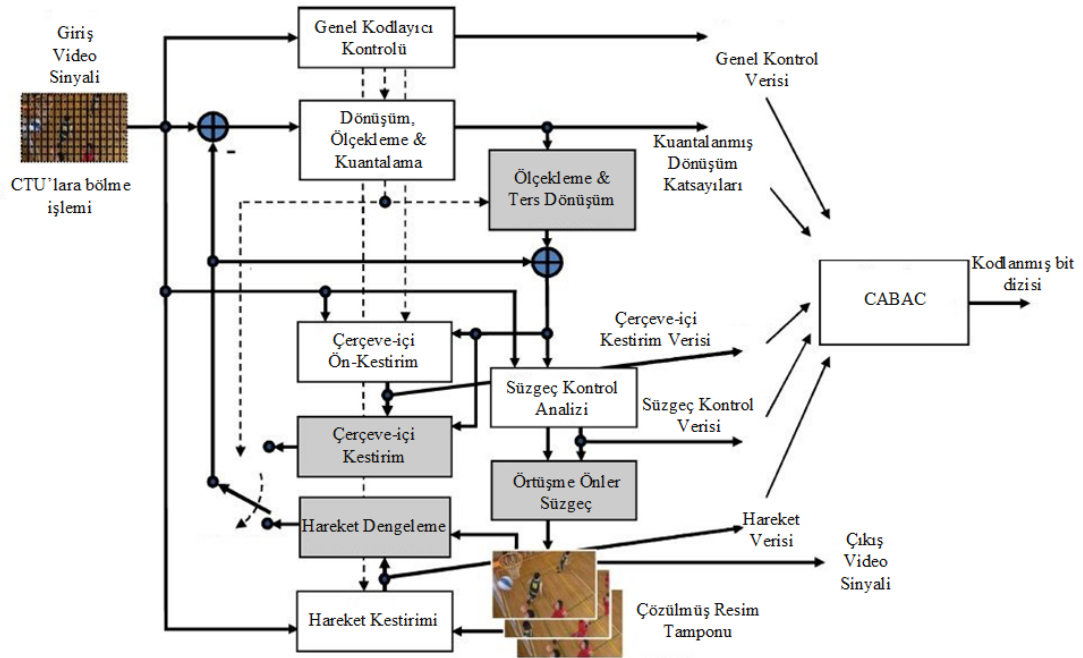
Kodlama işlemleri blok tabanlı yapıldığı için çözme işlemleri tamamlandıktan sonra birleştirilen bloklar arasında sert geçişler olabilmektedir. Bu sert geçişleri gidererek elde edilen imgenin görsel kalitesini arttırmak amacıyla H.264'te bloklama-önler süzgeç kullanılmıştır. Bunlarla birlikte süzgeçlenmiş makroblokları hareket kestiriminde kullanmak da yine daha iyi bir sonuç elde edilmesini sağlamaktadır.

2.HEVC

2.1.Genel Bakış

Gelişen teknolojiyle beraber, daha iyi bir video kodlama başarımına dolayısıyla da daha iyi video sıkıştırma kodlayıcılarına ve algoritmalarına ihtiyaç duyulmuştur. Bu ihtiyaç sonucunda MPEG ve VCEG gruplarının bir araya gelmesiyle kurulan JCT-VC (Joint Collaborative Team on Video Coding) tarafından 2013 yılında HEVC standardı ilan edilmiştir. HEVC'nin genel yapısı da önceki standartlarla benzerlikler göstermekle birlikte birçok yeni özelliği ve karmaşık algoritmayı da içermektedir. Bu yeni özellikler sayesinde HEVC, bir önceki standart olan H264/AVC'ye göre aynı bit oranında iki kat fazla sıkıştırma sağlayabilmektedir.

Şekil 2.1'de HEVC kodlayıcının blok şeması verilmiştir.



Şekil 2.1. Tipik bir HEVC kodlayıcının blok şeması [3]

HEVC'de de H.261'den itibaren süregelen hibrit video kodlama yaklaşımı kullanılmıştır. Yine önceki standartlarda olduğu gibi her çerçeve bloklara bölünerek işlenmektedir. İlk çerçeve I olarak kodlanmakta, sonraki çerçeveler için çerçeve-içi ve çerçeveler-arası hareket kestirim yaklaşımları kullanılarak her blok için uygun olan

yaklaşım belirlenmektedir. Maliyet hesabında da önceki standartlarda olduğu gibi [17]'de verilmiş olan Lagrangian kodlama tekniği kullanılmıştır. Her blok için kestirim işlemleri gerçekleştirildikten sonra da orijinal blok ve kestirilmiş blok arasında farkın alınmasıyla elde edilen artıklık verisine H.264'deki gibi DCT işlemine benzeyen fakat tam sayılarla gerçekleştirilebilen tam sayı dönüşüm işlemi uygulanmıştır. Ölçekleme, kuantalama ve entropi kodlama işlemlerinden sonra hareket kestirimi verileriyle beraber çıkış verisi oluşturulmuştur. Aşağıda sırasıyla HEVC'nin önemli özelliklerinden bahsedilmiştir [3]:

- Kodlama ağaç üniteleri (Coding Tree Units – CTU) ve kodlama ağaç blok yapıları (Coding Tree Block – CTB): H.264 kodlama standartında kodlama işlemleri için kullanılan MB'lar 16×16 boyutunda bir luma bileşeni, 8×8 boyutunda iki adet chroma bileşeni içermektedir. HEVC'de ise MB kavramının yerini CTU yapısı almıştır. CTU'lar ise luma kanalı için bir adet CTB, chroma kanalları için ise iki adet CTB içermektedir. Bir luma CTB'nin boyutu 16×16 , 32×32 veya 64×64 olabilmektedir. Daha büyük blok boyutlarının seçilebilmesi daha iyi sıkıştırma oranlarına olanak sağlamıştır.
- Kodlama üniteleri (Coding Units – CUs) ve kodlama blokları (Coding Blocks – CBs): Benzer şekilde luma ve chroma bileşenlerini içeren CB'lerden oluşan CU'lar ise CTU yapılarını oluşturmaktadır. CTU yapısı bir CU veya kodlayıcı seçimlerine göre daha küçük boyutta CU'ların birleşiminden oluşabilmektedir. CU'lar ise kestirim üniteleri (Prediction Units – PUs) ve dönüşüm ünitelerinden (transform units – TUs) oluşmaktadır.
- Kestirim üniteleri (Prediction Units – PUs) ve kestirim blokları (Prediction Blocks – PBs): Bir bloğun çerçeveler-arası mı yoksa çerçeve-içi yöntemlerle mi kodlanacağına CU seviyesinde karar verilmektedir. CU boyutuna göre farklı PU boyutları için kombinasyonlar denenerek CU'nun hangi boyutta PU'lardan oluştuğu belirlenmektedir. HEVC 4×4 'ten 64×64 'e kadar PB boyutlarını desteklemektedir.
- Dönüşüm üniteleri (Transform Units – TUs) ve dönüşüm blokları (Transform blocks – TBs): Kestirim işlemlerinden sonra orijinal blokla kestirilmiş bloğun farkı alınarak elde edilen artıklık verisi blok dönüşüm işlemleri kullanılarak kodlanmaktadır. TU birimi diğer ünite birimleri gibi luma için bir TB, chroma için

iki TB biriminden oluşmaktadır. Dönüşüm işlemleri için H.264'te olduğu gibi tam sayı tabanlı bir DCT benzeri bir dönüşüm kullanılmaktadır. TB boyutları 4×4 , 8×8 , 16×16 ve 32×32 olabilir. Bunlarla birlikte TU'lar ve PU'lar birbirinden farklı boyutlarda olabilmektedirler.

- Hareket vektörü iletimi: Gelişmiş hareket vektörü tahmini (Advanced Motion Vector Prediction – AMVP) olarak adlandırılan yapılan yapıda zamansal ve uzamsal komşu blokların hareket vektörleri olası adaylar olarak test edilmektedir. Zamansal veya uzamsal komşu PB'lerin hareket vektörleri aday komşu blok için uygun bulunduğu durumda hareket vektörleri birleştirilerek kodlanabilmektedir. Bu duruma birleştirme (merge) modu ismi verilmiştir.
- Hareket dengelemesi: Çeyrek piksel doğrulukta hareket vektörü kestirimi yapılabilmektedir. Bununla birlikte çoklu referans çerçeveler de hareket kestirimi için kullanılabilir.
- Çerçeveler-arası kestirim: Bu aşama diğer standartlar da kullanıyor olmasına karşın HEVC'de öncekilerden daha iyi bir hareket kestirim performansı sunan bir yaklaşım kullanılmaktadır. Çerçeveler-arası kestirim için tam arama yaklaşımı veya HM'de gerçekleştirilmiş olan Test Zone Search (TZS) isimli algoritma kullanılabilir.
- Çerçeve-içi kestirim: Çerçeve-içi kestirim için DC moduna ek olarak 33 adet yönelimsel mod bulunmaktadır.
- Kuantalama kontrolü: H.264'te olduğu gibi dönüşüm bloklarının boyutlarına bağlı olarak farklı kuantalama matrisleri kullanılabilir. Bu sayede istenen bit-oranına veya kaliteye göre kodlama ile ilgili ayarlamalar gerçekleştirilebilir.
- Entropi kodlama: Entropi kodlama için ise içerik uyarlamalı ikili aritmetik kodlama (Context Adaptive Binary Arithmetic Coding – CABAC) kullanılmaktadır. Bu yapı H.264'teki CABAC yapısına benzemesine rağmen hız, sıkıştırma performansı ve bellek gereksinimi ile ilgili geliştirmeler içermektedir. Paralel işlem gerçekleştirmeye de daha uygundur.
- Bloklama-önlü süzgeç: Burada da yapı H.264'e benzemesine karşın paralel işlemler gerçekleştirerek hız kazanmak için daha uygundur.

2.2. HEVC’de Ağaç Yapısının Oluşturulmasına Detaylı Bir Bakış

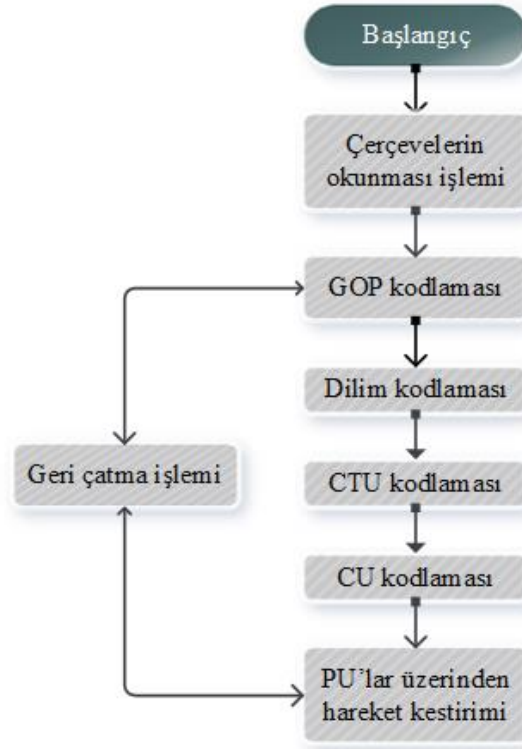
HEVC’de ağaç yapısının oluşturulması için gerçekleştirilen işlemlerin genel bir akış şeması Şekil 2.2’de verilmiştir.

Kodlama işlemleri için ilk olarak, orijinal çerçeveler alınarak GOP’ler oluşturulmaktadır. Bir sonraki aşama ise dilimlerin kodlanmasıdır. Dilim kodlaması içerisinde CTU ve CU’lar oluşturulmakta ve HK (hareket kestirimi) işlemleri CU’ların PU birimleri kullanılarak gerçekleştirilmektedir. Artıklık kodlaması için ise TU’lar kullanılmaktadır. Kodlama işlemleri sırasında maliyet hesapları yapılarak CU’lardan oluşan CTU’lara nihai şekli verilmektedir. Bununla birlikte GOP yapılarının kodlanması sırasında geri çatılmış çerçeveler de referans olarak HK işlemlerine dahil edilmektedir.

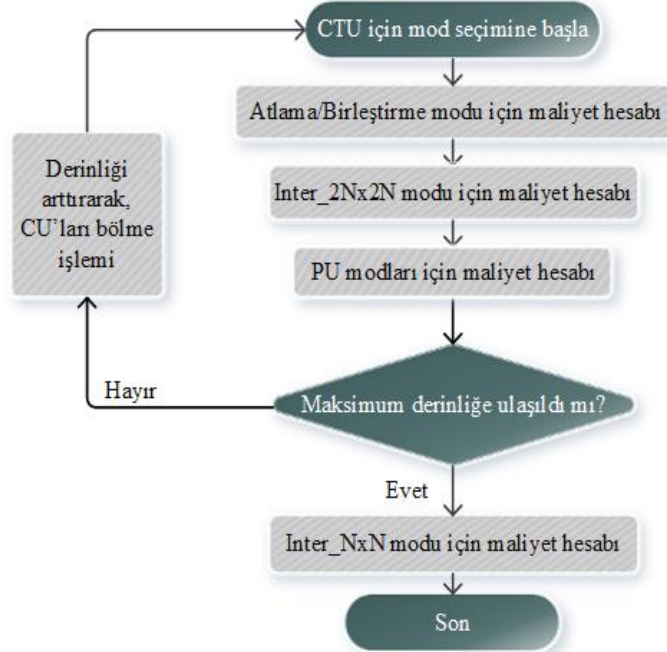
2.2.1. Mod seçimi

HEVC’deki kodlama birimi olan CTU, CU’lardan oluşmaktadır. CTU boyutu 64×64 olarak belirlenmiştir. Yapılan maliyet hesaplarına göre CTU bir veya birçok CU’dan oluşabilir. HEVC’deki mod seçimi işlemleri Şekil 2.3’de verilmiş olan algoritmaya göre gerçekleştirilmektedir.

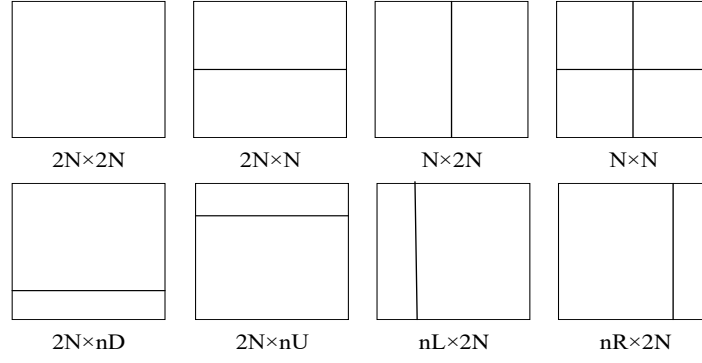
CU boyutları 8×8 piksel ile 64×64 piksel arasında değişebilmektedir. Derinlik 0 olduğu zaman CU boyutu 64×64 iken derinlik 3 olduğu zaman CU boyutu 8×8 ’dir. Mod belirleme işleminde ilk aşamada derinlik 0 olarak alınmaktadır. Her derinlikte ilk aşamada skip (atlama) modu kontrolü yapılır ve en iyi mod skip mod olarak belirlenirse diğer işlemler atlanır. Diğer durumda ise $2N \times 2N$ modu ve diğer modlar için inter prediction ve intra prediction işlemleri yapılarak maliyet (cost) hesabı yapılır. HEVC’deki PU modları Şekil 2.4’de verilmiştir. İlgili derinlikteki kodlama maliyeti en az olan mod bulunduktan sonra CU simetrik bir şekilde 4 parçaya bölünerek 4 tane CU oluşturulur ve aynı işlemler bu CU’lar için gerçekleştirilir. Öz-yinelemeli bir şekilde maksimum derinliğe kadar PU modları için maliyet hesabı işlemleri gerçekleştirilerek kodlanacak bloğun yapısı oluşturulur.



Şekil 2.2. HEVC'deki hareket kestirimi ile ağaç yapısı oluşturulması işlemlerinin akış şeması



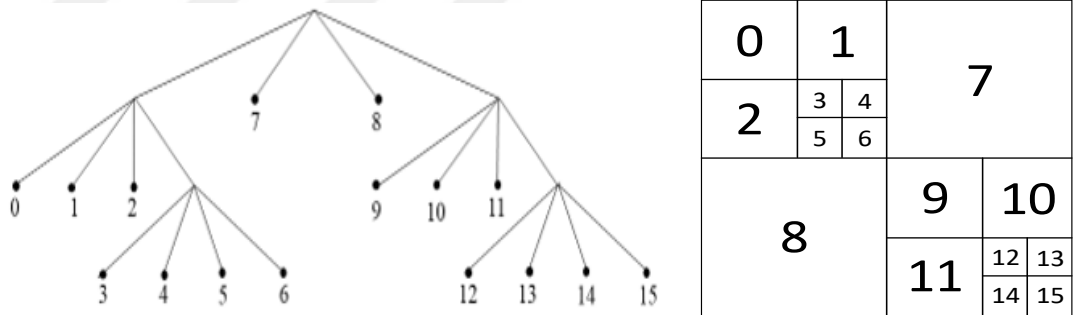
Şekil 2.3. HEVC'de derinlik belirleme ve mod seçimi işlemleri



Şekil 2.4. Örnek PU modları

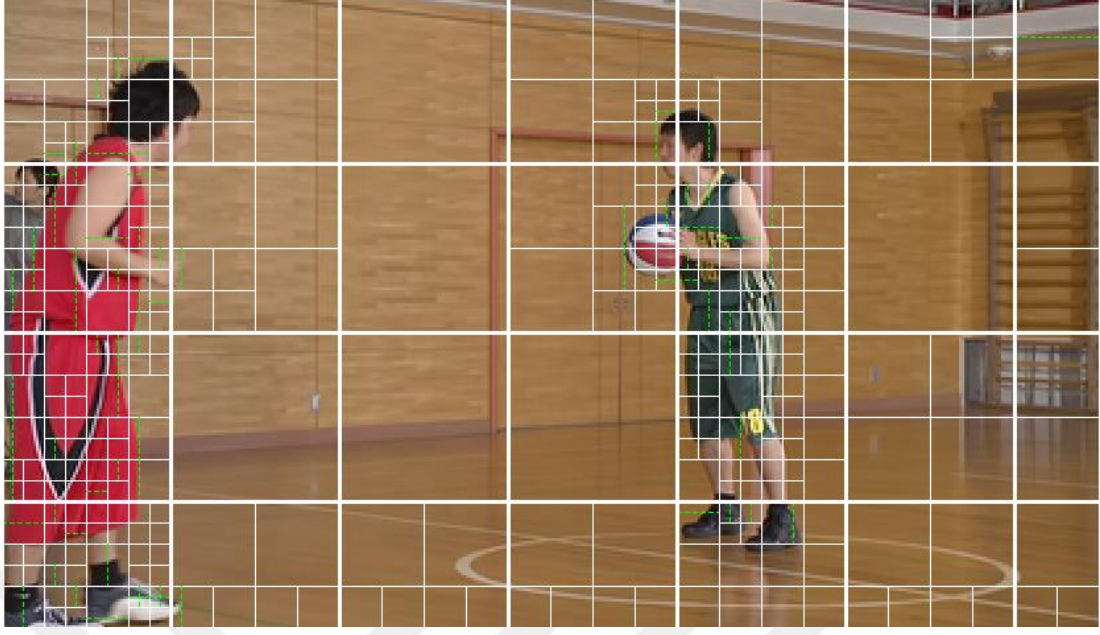
Maliyet hesabı işlemleri sonrasında çerçevenin ana iskeletini oluşturan CTU'lar farklı derinliklere sahip CU'lar içerebilir. Bu nedenle, kodlanacak her bir video çerçevesinde bulunan CTU'lar farklı bir ağaç yapısına sahip olabilir. Şekil 2.5'te CTU'ya ait örnek bir CU dağılımı ve buna ait ağaç yapısı verilmiştir.

Bunlarla birlikte CU'lar da yine maliyet hesabına göre farklı PU modlarından oluşabilmektedir. Şekil 2.6'da Basketball Pass dizisinin 2. çerçevesi için kodlama sırasında seçilmiş olan CU'lar ve bu CU'lar için seçilmiş olan PU modları verilmiştir.



Şekil 2.5. Örnek bir CTU yapısı (A sample CTU structure)

Şekil 2.6'da beyaz renk içinde kalan bölgeler seçilmiş CU'ları, yeşil renk ile çizilmiş çizgiler CU'lar içindeki seçilmiş PU modunu göstermektedir. Eğer CU içinde yeşil çizgi yoksa bu o CU için aynı boyuttaki PU'nun yani $2N \times 2N$ modunun seçilmiş olduğunu göstermektedir. Şekil 2.6'dan görüldüğü üzere homojen bölgeler için genellikle daha büyük boyutlar seçilmekte iken bölgedeki detaylar arttıkça daha küçük boyutlu CU'lar seçilmektedir. Bununla birlikte CU'lar içindeki detaylı ve farklı bölgelerin ise PU modunun seçiminde çok önemli bir rol oynadığı görülmektedir. Burada dikkat edilmesi gereken önemli bir nokta o PU'ların seçilmesi sırasında her bir PU modu için maliyet hesabının gerçekleştiriliyor olmasıdır.



Şekil 2.6. BasketballPass dizisi için belirlenmiş olan ağaç yapıları

2.2.2. Hareket kestirimi

HEVC’de, diğer video kodlayıcılarda olduğu gibi, HK işlemi için tam ve seyrek arama olmak üzere iki yaklaşım kullanılmaktadır. Seçilen yapılandırma parametrelerine (B ve P çerçeveleri) göre HK işlemleri için kullanılacak aday bloklar önceki ve/veya sonraki çerçevelerde aranabilmektedir. HEVC’de HK işlemleri sırasında blok uyumlama kriteri olarak SAD yaklaşımı kullanılmaktadır.

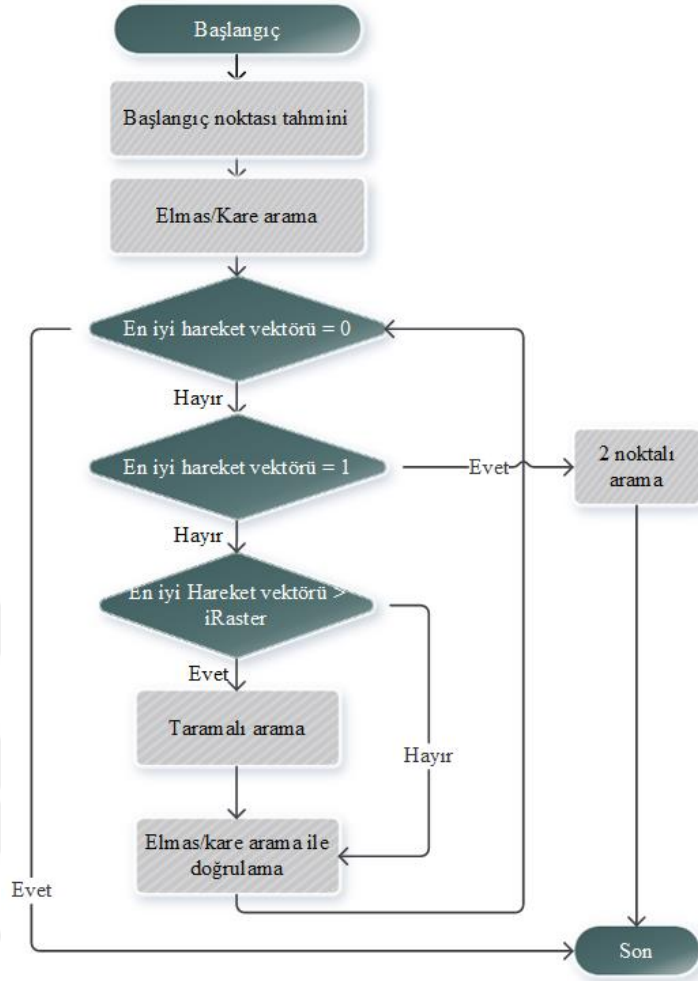
Tam arama yaklaşımı, önceki video kodlama standartlarındaki yapıya benzer şekilde HK yapılacak blok için önceden belirlenmiş arama penceresi içinde her aday blok için blok uyumlama hesabı yapılarak en uygun aday bloğun bulunmasıdır. Artıklık kodlaması ve hareket vektörlerinin kodlanması için gereken bit miktarları göz önünde bulundurularak uygun hareket vektörüne karar verilmektedir.

HEVC’de kullanılan TZS yaklaşımı elmas ve ızgara aramanın birleştirilmiş bir versiyonu olarak değerlendirilebilir. Bu yaklaşımla kodlama zamanını %90’a kadar azaltabilmektedir. TZS algoritmasının akış şeması Şekil 2.7’de verilmiştir.

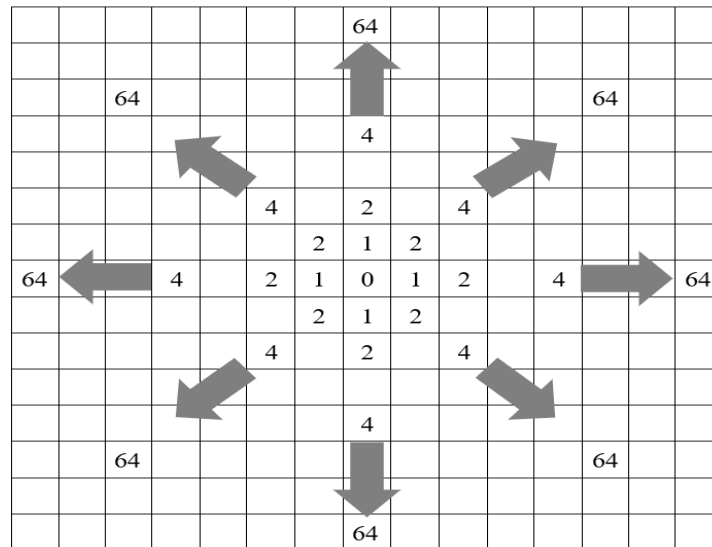
TZS algoritmasında ilk olarak kodlama yapılacak blok için hareket vektörü hesaplanmış komşu blokların yardımıyla HK işlemine başlanacak olan nokta belirlenmektedir. Sonraki aşamada ise elmas veya kare örüntü kullanılarak uyumlama

ölçütünün en uygun sonucu verdiği pozisyon tespit edilmektedir. TZS’de kullanılan elmas örüntüsü Şekil 2.8’de verilmiştir. Standart elmas arama örüntüsünden farklı olarak daha büyük ve aşamalı bir elmas yapısı kullanılmaktadır. Bu yapıdaki arama aralıkları 1 ile 64 arasında 2’nin kuvveti olacak şekilde belirlenmektedir. Şekil 2.8’de verilmiş olan rakamlar arama merkezine olan uzaklığı temsil etmekte ve arama işlemi iç içe geçmiş elmas örüntüler üzerinden gerçekleştirilmektedir. Uzaklığın artması ise bir sonraki tura geçildiği anlamına gelmektedir.

Her turda belirli uzaklıkta kalan noktalarda uyumlama ölçütü hesaplanmaktadır. İki tur sonunda en iyi uyumlama değerini veren nokta değişmiyorsa o konum o an için en iyi hareket vektörü olarak belirlenmekte ve arama örüntüsünün kontrol etmemiş olduğu iki nokta için de uyumlama ölçütü hesaplanarak karar verme işlemi gerçekleştirilmektedir. Fakat bu noktada bir istisna olarak elmas örüntüsünün yapısından dolayı eğer en iyi hareket vektörü 0 seçilmişse iki noktalı arama yapmaya gerek kalmamaktadır. Hesaplanmış olan en iyi hareket vektörü daha önceden belirlenmiş olan tarama eşik değerinden (iRaster) büyükse yerel bir minimum noktasına takılmamak için ızgara şeklinde taramalı bir yapıda belirli noktalar kontrol edilerek en iyi uyumlama değerini veren nokta belirlenmektedir. Izgara şeklindeki arama işleminde arama aralıkları iRaster parametresi büyüklüğündedir. Izgara şeklindeki arama işleminde en iyi uyumlama değerini veren nokta belirlendikten sonra ise ilk aşamadaki elmas veya kare örüntü ile arama işlemlerine benzer olarak özyinelemeli (iterative) bir şekilde en iyi uyumlama değerini veren son hareket vektörü hesaplanmaktadır.



Şekil 2.7. TZS algoritmasının akış şeması



Şekil 2.8. HEVC’de TZS için kullanılan elmas örüntü

2.2.3. Maliyet hesabı

HEVC’de hareket kestirimiyle birlikte mod seçimi işlemleri hesapsal karmaşıklığın önemli bir bölümünü oluşturmaktadır. Hesapsal karmaşıklığın çok yüksek olması da uzun kodlama sürelerine neden olmaktadır. Çerçeveler-arası hareket kestiriminde aday CU’lar için uygun referans bloklar bulunamaması durumunda da çerçeve-içi hareket kestirim yaklaşımlar kullanılmakta, bu durum da sıkıştırma veriminin düşmesine neden olmaktadır. Bu noktalar değerlendirildiğinde sıkıştırma oranı ve işlem yükü dolayısıyla kodlama zamanı arasında bir denge kurulması gerektiği açıktır. Başka bir deyişle kodlayıcı daha verimli bir kodlama için arama işlemine devam etmek veya devam etmeyerek zaman kazanmak arasında bir seçim yapmalıdır. Bu dengeyi sağlamak amacıyla video kodlayıcılarda genellikle Lagrangian kodlama yöntemi kullanılmaktadır.

HEVC’de Lagrangian çarpanı QP parametresi kullanılarak belirlenmektedir. Eşitlik (2.1)’de bu işlem verilmiştir. Eşitlik (2.1)’e göre hesaplanan λ değeri mod seçimi, hareket kestirimi ve artıklık verisinin kodlanması sırasında yapılan kuantalama işlemlerinde kullanılmaktadır.

$$\lambda = \alpha \times Q^2 \quad (2.1)$$

Bu eşitlikte λ Lagrangian çarpanını, Q sembolü QP parametresi tarafından kontrol edilen kuantalama adım aralığını temsil etmektedir.

HEVC’de mod seçimi işlemleri sırasında kodlayıcı Eşitlik (2.2)’ye göre karar verme işlemini gerçekleştirmektedir.

$$c^* = \arg \min (D_k(c) + \lambda \times R_k(c)), c \in C_k \quad (2.2)$$

Bu eşitlikte C_k seçilebilecek olan modların kümesini, $D_k(c)$ orijinal blok ile o bloğun yeniden oluşturulmuş hâli arasındaki SSD (Sum of Squared Difference) değerini, $R_k(c)$ ise c modu kullanıldığında hareket vektörlerinin ve artıklık verilerinin kodlanabilmesi için gerekli olan bit miktarını ifade etmektedir.

HEVC’de hareket kestirimi işlemleri sırasında kodlayıcı Eşitlik (2.3)’e göre karar verme işlemini gerçekleştirmektedir.

$$(p^*, m^*) = \arg \min (D_k(p, m) + \lambda \times R_k(p, m)), p \in P, m \in M \quad (2.3)$$

Bu eşitlikte P referans çerçeve listesine, M ise aday hareket vektör kümesine karşılık gelmektedir. $D_k(p, m)$ aday bloğun kestirilmiş çerçeve ile arasındaki SAD'yi ifade etmekte iken, $R_k(p, m)$ de hareket vektörünün kodlanması için gerekli bit miktarını temsil etmektedir.

Yukarıda verilmiş olan eşitliklerden de görüldüğü üzere kuantalama parametresinin maliyet hesabı üzerinde çok önemli bir etkisi bulunmaktadır. Aynı koşullar altında, QP değerinin düşük seçilmesi görsel kaliteyi ve bit miktarını arttırmakta iken hesapsal karmaşıklığını da arttırmakta, bu değer yüksek seçilmesi ise görsel kaliteyi ve bit miktarını azaltmakta iken hesapsal karmaşıklığını da azaltmaktadır. HM konfigürasyon dosyalarından ayarlanabilen QP değeri sayesinde uygulamaya göre sıkıştırma ve işlem yükü arasında denge sağlanabilmektedir.

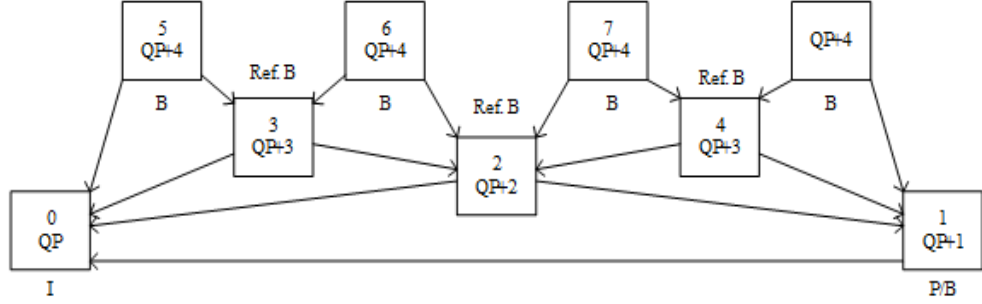
2.3. HEVC İçin Test Konfigürasyonları

HM referans yazılımında kodlayıcı için gerekli ayarlama işlemleri “.cfg” uzantılı dosyalar kullanılarak gerçekleştirilebilmektedir. Bu dosyalar sayesinde CU boyutları, QP parametresi, mod seçiminde derinlik belirlemesi, intra çerçevelerin periyodu, GOP boyutları, hareket kestirimi yöntemleri, hareket kestiriminde arama aralığı gibi birçok özellik ayarlanabilmektedir.

JCT-VC tarafından önerilen test koşulları aşağıda verilmiştir:

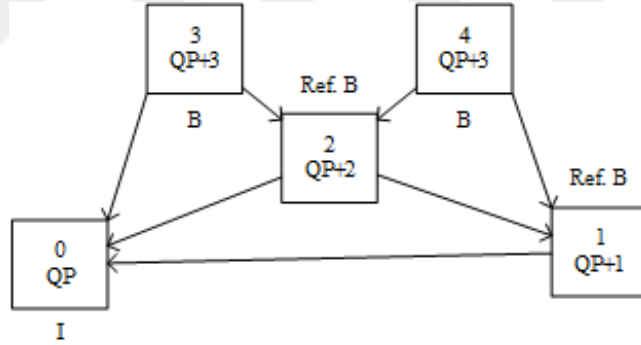
- All Intra (AI): Bu kodlama türünde tüm çerçeveler birbirinden bağımsız olarak çerçeve-içi şekilde kodlanmaktadır. Çerçeveler-arası kodlama gerçekleştirilmediği için bu kodlama türünde sıkıştırma performansı görece olarak düşüktür.
- Random Access (RA): Bu kodlama türünde çerçeve oranına bağlı olarak her saniyedeki çerçeveler kendi içinde çözülebilir. Bu amaçla belirli aralıklarla çerçeveler çerçeve-içi olarak kodlanmaktadır. Görüntüleme sırası ve kodlama-çözme işlemlerinde çerçevelerin farklı bir sırada olması yapısal bir gecikmeye neden olmaktadır. Bu konfigürasyon televizyon ve internet yayıncılığı benzeri uygulamalarda kullanılabilmektedir. Şekil 2.9'da RA konfigürasyonu için örnek bir

GOP yapısı verilerek kullanılan QP değerleriyle birlikte kodlama sırası da blokların üzerinde belirtilmiştir.



Şekil 2.9. RA konfigürasyonu için örnek bir GOP yapısı

- Low Delay (LD): Bu konfigürasyon türü rastgele erişime gerek olmayan, düşük gecikmeli uygulamaları temsil etmektedir. Sadece ilk çerçeve çerçeve-içi olarak kodlanmaktadır. Bu konfigürasyon sadece P çerçeveler veya B çerçeveler ile gerçekleştirilebilmektedir. Şekil 2.10'da LD konfigürasyonunda B çerçevelerin kullanıldığı (LDB) örnek bir GOP yapısı verilerek kullanılan QP değerleriyle birlikte kodlama sırası da blokların üzerinde belirtilmiştir.



Şekil 2.10. LDB konfigürasyonu için örnek bir GOP yapısı

Tablo 2.1'de HEVC'de test için kullanılması önerilen ve bu tez çalışmasında da kullanılmış olan diziler verilmiştir.

2.4. HEVC Zaman Analizi

HEVC'de hesapsal karmaşıklığın önemli bir bölümünü mod seçimi ve hareket kestirimi işlemleri ile maliyet hesaplarının gerçekleştirilerek ağaç yapısına nihai şeklinin verilmesi oluşturmaktadır. Bununla birlikte artıklık verisinin kodlanması sırasında gerçekleştirilen dönüşüm ve kuantalama işlemlerinin de hesapsal karmaşıklığa önemli bir etkisi bulunmaktadır. Tablo 2.2'de BasketballPass dizisi için

RA konfigürasyonunda HEVC'deki birimlerin toplam süreye oranları verilmiştir [7]. Çalışmada hareket kestirimi için TZS algoritmasından yararlanılmıştır.

Tablo 2.1. Performans değerlendirilmesinde kullanılan görüntü dizileri ve özellikleri

Video Dizileri	Çözünürlük	Çerçeve Sayısı	Dizi Özellikleri
Traffic	2560×1600	150	Orta bütünsel hareket, yüksek yerel hareket, zengin içerik
Kimono	1920×1080	240	Orta bütünsel hareket, orta yerel hareket, zengin içerik
ParkScene	1920×1080	240	Orta bütünsel hareket, orta yerel hareket, zengin içerik
FourPeople	1280×720	600	Düşük bütünsel hareket, orta yerel hareket, düşük içerik
Johnny	1280×720	600	Düşük bütünsel hareket, orta yerel hareket, düşük içerik
KristenAndSara	1280×720	600	Düşük bütünsel hareket, orta yerel hareket, düşük içerik

Tablo 2.2. HEVC zaman analizi

Birim	Zaman (%)
Mod Seçimi + Hareket Kestirimi	49,8
Dönüşüm + Kuantalama	10,7
Diğer işlemler	39,5

Bununla birlikte mod seçimi ve hareket kestirimi işlemlerindeki yükün azaltılması dolaylı yoldan diğer birimlerdeki sürelerin azaltılmasını da sağlamaktadır.

3. LİTERATÜRDEKİ ÇALIŞMALAR

3.1. Hareket Kestirimi

Video kodlamanın en kritik ve en fazla işlem yüküne sahip aşaması zamansal artıklığın giderilmesi olarak değerlendirilmektedir. HEVC’de de önceki standartlara benzer bir şekilde hesapsal karmaşıklığın önemli bir bölümünü hareket kestirimi işlemleri oluşturmaktadır.

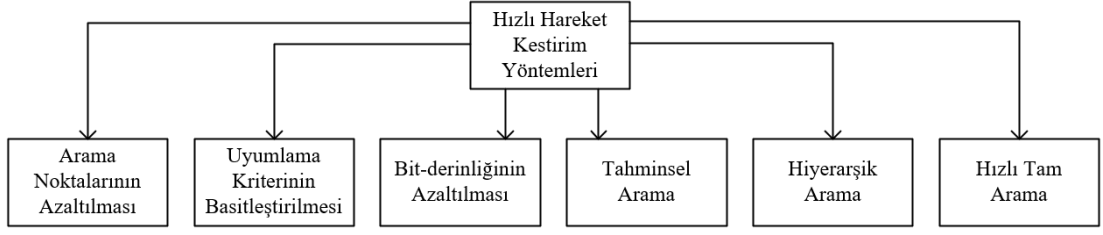
Hareket kestiriminin amacı, aday bloğu referans çerçevelerde belirli bir uyumlama kriteri doğrultusunda arayarak en az artıklık verisine neden olacak uygun bloğu tespit etmektir. Bu sayede sıkıştırma işlemini daha başarılı bir şekilde gerçekleştirilmesi hedeflenmektedir.

Hareket kestiriminde en doğru arama sonucunu tam arama yaklaşımı vermektedir. Tam arama yaklaşımında referans blok içindeki her aday hareket vektörü için uyumlama kriteri hesaplanmaktadır. Hareket kestirimi uygulamalarında uyumlama kriteri olarak genellikle SAD yaklaşımı kullanılmaktadır. Eşitlik (3.1)’de tam arama yaklaşımı için SAD hesabı verilmiştir. SAD hesabında genellikle kodlanacak görüntü dizisi özelliklerine göre 8 veya 10 bitlik piksel değerleri kullanılmaktadır.

$$SAD(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|I^t(i,j)-I^{tref}(i+m,j+n)| \quad (3.1)$$

Bu eşitlikte I^t aday bloğu, I^{tref} ise referans bloğu göstermektedir. (i, j) $N \times N$ boyutundaki bir blokta bulunan piksellerin indislerini, (m, n) ise aday hareket vektörü temsil etmektedir.

Hareket kestiriminde en doğru arama sonucunu tam arama yaklaşımı vermesine rağmen hesapsal karmaşıklığından dolayı hız gerektiren uygulamalarda kullanılması mümkün değildir. Bu problemi gidermek için literatürde önerilmiş bir çok yöntem mevcuttur. Hızlı hareket kestirim yaklaşımları [18]’deki çalışmada 6 sınıfa ayrılmıştır. Şekil 3.1’de bu sınıflar verilmiştir.



Şekil 3.1. Hızlı hareket kestirim yaklaşımları

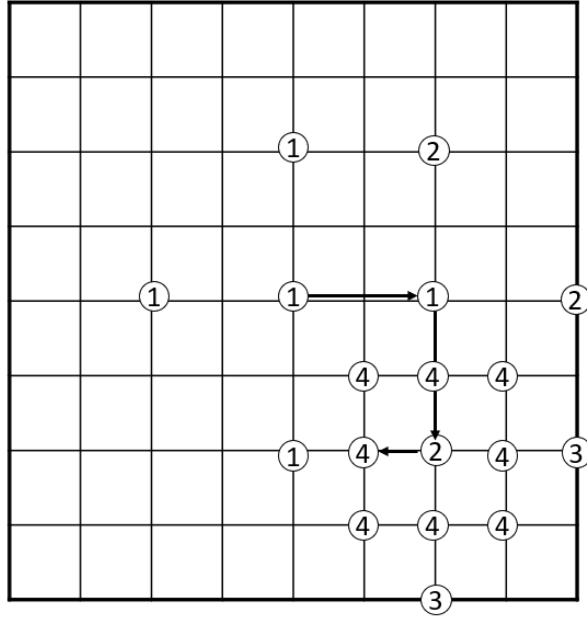
Şekil 3.1’de verilmiş olan hızlı arama yaklaşımları kendi başlarına kullanılabilirlikle birlikte bu yaklaşımların beraber de kullanıldığı birçok çalışma da literatürde mevcuttur.

3.1.1. Arama noktalarının azaltılması

Bu tarz hızlı arama yöntemlerinde referans çerçevedeki tüm aday hareket vektörleri için uyumlama kriteri hesaplanması yerine, belirli bir arama örüntüsü doğrultusunda belirli noktalarda uyumlama kriteri hesaplayarak sonuca ulaşmak hedeflenmektedir.

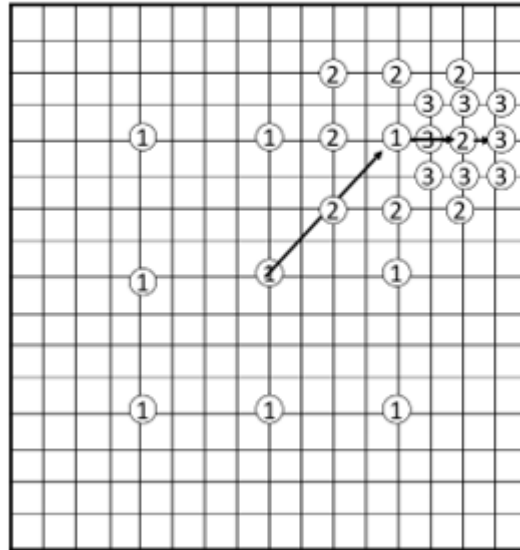
TDLS (Two-Dimensional Logarithmic Search – İki Boyutlu Logaritmik Arama) [19], TSS (Three Step Search – Üç Adımlı Arama) [20], NTSS (New Three Step Search – Yeni Üç Adımlı Arama) [21], FSS (Four Step Search – Dört Adımlı Arama) [22], CS (Cross Search) [23] ve DS (Diamond Search – Elmas Arama) [24], HS (Hexagonal Search – Altıgen Arama) [25] yöntemleri bu yaklaşımların en bilinen örnekleridir.

Şekil 3.2’de TDLS yöntemi ile ilgili örnek bir arama senaryosu verilmiştir. Bu yöntemde ilk olarak şekil üzerinde 1 numaralı noktaların bulunduğu bir örüntü başlangıç noktasına yerleştirilmektedir. Sonrasında merkez noktada en küçük SAD değeri elde edilene kadar en küçük SAD değerini veren noktanın etrafına başlangıçtaki örüntü yerleştirilerek arama işlemine devam edilmektedir. Merkez noktada en küçük SAD değeri elde edildiğinde ise aday noktanın etrafındaki noktalar kontrol edilerek (4 numaralı noktaların bulunduğu aşama) en küçük SAD değerini veren nokta bulunarak arama işlemi sonlandırılmaktadır.



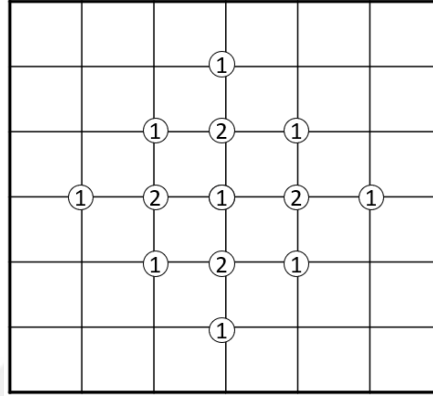
Şekil 3.2. TDSL için örnek bir arama senaryosu

TSS yöntemindeki örüntü ise 9 arama noktası içermektedir. Bu yöntemde her aşamada en küçük SAD değerine sahip nokta bulunduğundan sonra örüntüdeki adım aralıkları küçültülerek arama işlemine devam edilmektedir. Bu sayede $[-16,16]$ arama aralığına sahip bir pencerede en fazla 25 SAD hesaplama işleminde sonuca ulaşılabilmektedir. Şekil 3.3’de TSS yöntemi ile ilgili örnek bir arama senaryosu gösterilmiştir.



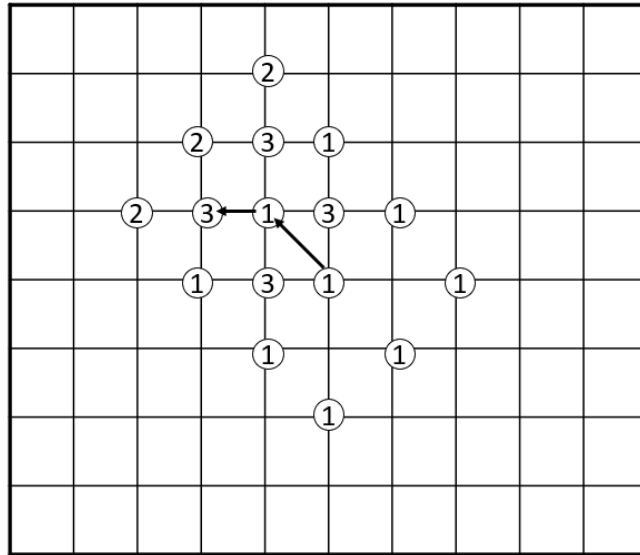
Şekil 3.3. TSS yöntemi için örnek bir arama senaryosu

DS yönteminde Şekil 3.4'te verilen elmas şeklinde bir arama örüntüsü kullanılmaktadır. İlk etapta 1 numaralı noktalar kullanılarak diğer yöntemlerdeki gibi en küçük hesaplanan SAD değeri merkezde olana kadar arama işlemine devam edilmekte, sonrasında 2 numaralı noktalar kullanılarak arama işlemi tamamlanmaktadır.



Şekil 3.4. DS'de kullanılan arama örüntüsü

DS yöntemi ile gerçekleştirilmiş örnek bir hareket kestirimi senaryosu ise Şekil 3.5'te verilmiştir.



Şekil 3.5. DS ile gerçekleştirilmiş örnek bir hareket kestirimi

Bu yöntemlerde hareket kestirimi işlemleri hızlandırılmasına rağmen arama örüntülerinden dolayı yerel minimum noktalarında arama işlemi sonlanabilmektedir.

Örneğin detayları Bölüm 2.2.2’de verilen TZS’de bu problemin önüne geçmek için başlangıç noktasından belirli bir eşikten daha fazla uzaklaşıldıysa tüm arama penceresinde alt örnekleme yapılarak belirli noktalar değerlendirilmekte ve bulunan noktanın etrafında DS işlemine benzer bir işlem gerçekleştirilmektedir. Bu yöntemle yerel minimum noktaları ile ilgili yaşanabilecek problemlerin engellemesi amaçlanmaktadır.

3.1.2. Uyumlama kriterinin basitleştirilmesi

Blok hareket kestirimi işlemlerindeki hesapsal karmaşıklık, aday hareket vektörünün bulunduğu konumda SAD hesabı gerçekleştirilirken kullanılan piksellerin alt örneklenmesi ile de azaltılabilmektedir. [26]’da hiyerarşik bir yaklaşımla bu yöntem uygulanmıştır. [27]’deki çalışmaya benzer bir yaklaşımla uyarlamalı bir şekilde hesaba katılan piksel sayılarının azaltıldığı yaklaşımlar da literatürde mevcuttur. Bunlarla birlikte tümlev imgeler sayesinde 3 toplama ve 1 çıkarma işlemiyle elde edilen blok toplamlarının farkının kullanılması da bu yöntemlere örnek olarak verilebilir.

[28]’de ekran içerik görüntüleri için bir çerçeve-içi ve çerçeveler arası kestirim yöntemi önerilmiştir. Bu yaklaşımda blokların özeti çıkarılmakta ve maliyet hesapları bu özetle gerçekleştirilmektedir. Özet boyutu blok boyutlarından daha küçük bir yapıda olduğu hesapsal karmaşıklığın azaltılması sağlanmaktadır. Bununla birlikte bu yöntem HM üzerinde de çalıştırılmıştır.

3.1.3. Bit-derinliğinin azaltılması

Bu yöntemlerin temel amacı blok uyumlama sırasında yüksek bit çözünürlüklerinde yapılan işlemleri daha düşük bit çözünürlüğüne sahip imgelerde yaparak blok uyumlama işleminin işlem yükünü azaltmaktır. Blok uyumlama işlemi sırasında genellikle donanımsal olarak da etkin bir şekilde gerçekleştirilebilen XOR işleminden faydalanılmaktadır.

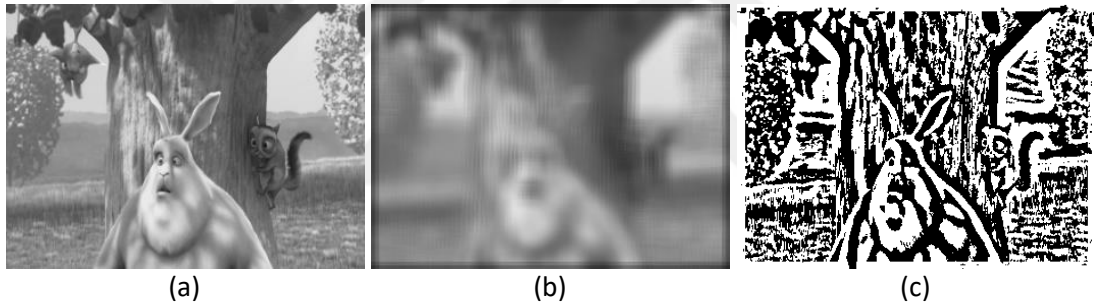
1BT (1-Bit Transform – 1 Bit Dönüşümü) [4] temelli hareket kestiriminde imge ilk olarak Eşitlik (3.2)’de ifadesi verilmiş olan çoklu bant-geçiren süzgeçten geçirilmektedir.

$$K_{1BT}(i,j)=\begin{cases} 1/25, & \text{if } i,j \in [0,4,8,12,16] \\ 0, & \text{diğer durumda} \end{cases} \quad (3.2)$$

Bant-geçiren süzgeçten geçirilmiş imge ile orijinal imge çerçevesi karşılaştırılarak ikili (binary) bir imge edilmektedir. Denklem (3.3)'te bu işlem verilmiştir.

$$B(i,j)=\begin{cases} 1, & \text{if } I(i,j) \geq (I(i,j) * K_{1BT}(i,j)) \\ 0, & \text{diğer durumda} \end{cases} \quad (3.3)$$

Bu denklemde I giriş imgesini, * evrişim (convolution) işlemini, B ise elde edilen ikili imgeyi temsil etmektedir. 1BT dönüşümünde kullanılan bant-geçiren süzgeç yapısı sayesinde imgedeki detaylar klasik eşikleme yöntemlerine daha fazla korunmakta, bu durum da hareket kestirimi başarımına katkı sağlamaktadır. Şekil 3.6'da "Big Buck Bunny" dizisinden alınmış örnek bir çerçeve, bu çerçevenin süzgeçlenmiş ve 1 bit dönüşümü gerçekleştirilmiş hâli verilmiştir.



Şekil 3.6. a) "Big Buck Bunny" dizisinden örnek bir görüntü b) (3.2) ile süzgeçlenmiş görüntü c) (3.3) ile dönüşümü gerçekleştirilmiş görüntü

1BT dönüşümü temelli hareket kestiriminde blok uyumlama işlemi NNMP (Number of Non-Matching Points- Uyumsuz Nokta Sayısı) ölçütü kullanılmaktadır. Eşitlik (3.4)'te NNMP hesabı verilmiştir.

$$NNMP_{1BT}(m,n)=\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B^t(i,j) \oplus B^{t_{ref}}(i+m,j+n) \quad (3.4)$$

Bu eşitlikte B^t aday bloğu, $B^{t_{ref}}$ ise referans bloğu göstermektedir. (i, j) $N \times N$ boyutundaki bir blokta bulunan piksellerin indislerini, (m, n) ise aday hareket vektörü temsil etmektedir. \oplus sembolü ile temsil edilmiş olan XOR işlemiyle NNMP hesabı gerçekleştirilerek, en küçük NNMP değerine sahip (m,n) pozisyonu hareket vektörü olarak belirlenmektedir.

[29]'da önerilmiş olan MF-1BT (Multiplication Free 1-Bit Transform – Çarpmasız 1-Bit Dönüşümü) temelli hareket kestirimi yaklaşımında ise evrişim işleminde kullanılan çekirdek normalizasyon için kullanılacak olan katsayı 2'nin kuvveti olacak şekilde değiştirilmiştir. Bu sayede dönüşüm işleminin bit kaydırma işlemleriyle gerçekleştirilebilmesi hedeflenmiştir. MF-1BT yaklaşımında da uyumlama kriteri olarak NNMP ölçütü kullanılmıştır.

2BT (2-Bit Transform – 2-Bit Dönüşümü) [30] temelli hareket kestirimi yaklaşımında 1 bit düzlemi yerine 2 bit düzlemi kullanılması önerilmiştir. Bit düzlemleri oluşturulurken blokların ortalamaları ve değişinti değerleri dikkate alınmıştır. Bit düzlemlerinin oluşturulması işlemleri Eşitlik (3.5)'de verilmiştir.

$$B_1(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq \mu \\ 0, & \text{diğer} \end{cases}$$

$$B_2(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq \mu + \sigma_a \text{ veya } I(i,j) \leq \mu - \sigma_a \\ 0, & \text{diğer} \end{cases} \quad (3.5)$$

Bu eşitlikte μ blok ortalamasını, σ_a ise standart sapmayı temsil etmektedir. B_1 ve B_2 ise oluşturulmuş bit düzlemlerini temsil etmektedir. Bu yöntemde kullanılan blok uyumlama ölçütü Eşitlik (3.6)'da verilmiştir.

$$NNMP_{2BT}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i,j) \oplus B_1^{tref}(i+m,j+n)\} \parallel \{B_2^t(i,j) \oplus B_2^{tref}(i+m,j+n)\} \quad (3.6)$$

Bu eşitlikte \parallel sembolü OR işlemini göstermektedir. Bu yöntemde 1BT yönteminden daha iyi sonuçlar elde edilmesine rağmen ortalama ve standart sapma hesabı işlemleri nedeniyle hesapsal karmaşıklığı daha fazladır.

C-1BT (Constraint 1-Bit Transform – Kısıtlanmış 1-Bit Dönüşümü) [5] yaklaşımında 1BT yaklaşımında kullanılan bit uzayına ek olarak uyumlamaya katılacak güvenilir pikselleri belirlemek amacıyla kısıt maskesi kullanılmaktadır. 1BT yaklaşımında süzgeçleme işlemleri gerçekleştirilirken karşılaştırma işlemi sırasında ortalamaya yakın olan değerler başarıyı düşürmektedir. Süzgeçleme işleminde kullanılan bir eşik değeri sayesinde güvenilir pikseller belirlenmiştir. Kısıt maskesi oluşturma işlemi Eşitlik (3.7)'de verilmiştir.

$$CM(i,j)=\begin{cases} 1, & \text{if } |I(i,j)-I(i,j)*K_{MF-1BT}|\geq D \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

Bu eşitlikte CM kısıt maskesini, K_{MF-1BT} ise MF-1BT yaklaşımında kullanılan evrişim çekirdeğini temsil etmektedir.

Kısıt maskesiyle birlikte güncellenmiş olan uyumlama kriteri ise Eşitlik (3.8)'de verilmiştir.

$$CNNMP(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{CM^t(i,j)\|CM^{tref}(i+m,j+n)\} \odot \{B^t(i,j)\oplus B^{tref}(i+m,j+n)\} \quad (3.8)$$

Bu eşitlikte \odot sembolü AND işlemini göstermektedir. AND işlemi sayesinde kısıt maskesiyle birlikte piksellerin değerlendirmeye alınıp alınmayacağı belirlenmektedir.

[31]'de önerilmiş olan T-GCBPM (Truncated Gray-Coded Bit Plane Matching- Kesik Gray Kodlamalı Bit Düzlemi Eşleme) yönteminde piksellerin gray kodu karşılıklarının en değerliksiz bitleri kesilerek kullanılması önerilmiştir. 8 bitlik piksel değerleri gözönüne alındığında standart ikili kodlamada ardışık iki değer arasında 7 bite kadar değişim olabilirken, Gray kodlamada her aşamada maksimum 1 bit değişim olmaktadır. Bu durum, hareket kestirimi işlemi az sayıda bitle gerçekleştirileceği zaman başarımdan verilecek ödünün standart ikili sayı sistemine göre daha az olmasını sağlamaktadır. İmgeler ikilileştirildikten sonra HK aşamasında kullanılan uyumlama ölçütü ise Eşitlik (3.9)'da verilmiştir.

$$CM_{TGC}(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=NTB}^{K-1}2^{k-NTB}\times\{g_k^t(i,j)\oplus g_k^{tref}(i+m,j+n)\} \quad (3.9)$$

Bu eşitlikte CM_{TGC} uyumlama kriterini, k ilgili bit-uzatını, NTB (Number of Truncated Bits) kesilen bit sayısını, g_k^t ve g_k^{tref} giriş ve referans çerçeveler için ilgili bit uzayı için için gray kodları temsil etmektedir. Örneğin NTB değerinin 5 olarak seçilmesi en değerlikli 3 bit uzayının kullanılacağı anlamına gelmektedir.

[6]'da SGC (Selective Gray Coding – Seçmeli Gray Kodlama) temelli, 1 bit uzayı kullanan bir hareket kestirimi yaklaşımı önerilmiştir. Bu çalışmada hareket kestirimi yapılacak bloktaki piksellerin her biri için farklı bir düzlem seçilmesi önerilmiştir. Bu sayede tek bir bit düzlemine bağımlı kalmak yerine T-GCBPM yöntemindeki gibi farklı bit uzayları da değerlendirmeye alınabilmektedir. Fakat T-GCBPM yönteminden farklı olarak her piksel için bir bit düzlemi seçilmektedir. Çalışmada 5. ve 7. bit düzlemleri kullanılmıştır. Pikseller için bit düzlemi seçme işlemleri Şekil 3.7'de verilen şablona göre seçilmektedir.

g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5

Şekil 3.7. SGC yönteminde önerilmiş olan şablon

Aday blok referans çerçeve içinde aranırken referans bloklarda da seçme işlemi bu şablona göre gerçekleştirilerek NNMP hesabıyla uyumlama işlemi gerçekleştirilmektedir.

[32]'de önerilmiş olan çalışmada önerilen 2BT-SG isimli yaklaşımda ise Şekil 3.7'deki şablona benzer bir şekilde 5. ve 6. düzlemlerin birlikte, 7. düzlemin ise tek başına kullanıldığı, toplamda 2 bit düzlemi kullanan bir yöntem önerilmiştir. 7. bit düzlemi imgeler hakkında kabaca bilgiler içerirken, 5. ve 6. bit düzlemi sayesinde uyumlama aşamasında imgedeki detayların da dikkate alınması sağlanmaktadır. Bu düzlemlere uygun şekilde düzenlenmiş olan $NNMP_{2BT-SG}$ uyumlama kriterinin hesaplanması da Eşitlik (3.10-3.13)'de verilmiştir.

$$\text{nnmp}_1(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_1^t(i,j) \oplus B_1^{\text{ref}}(i+m,j+n) \quad (3.10)$$

$$\text{nnmp}_2(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_2^t(i,j) \oplus B_2^{\text{ref}}(i+m,j+n) \quad (3.11)$$

$$\text{nnmp}_3(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i,j) \oplus B_1^{\text{ref}}(i+m,j+n)\} \odot \{B_2^t(i,j) \oplus B_2^{\text{ref}}(i+m,j+n)\} \quad (3.12)$$

$$\text{NNMP}_{2\text{BT-SG}}(m,n) = \text{nnmp}_1(m,n) + \text{nnmp}_2(m,n) + \text{nnmp}_3(m,n) \quad (3.13)$$

[33]'te 8-bit piksel derinliğine sahip ekran içerik görüntülerinde bit düzlemi seçerek hareket kestirimi hesabına ilişkin bir yöntem önerilmiştir. Bu yaklaşımda doğal ikili (binary) kodları kullanılmaktadır. Hareket kestirimi yapılacak olan blokta bulunan piksel değerleri Eşitlik (3.144) yardımı ile birbirleri ile karşılaştırılarak o bloktaki pikseller için hangi bit düzleminin seçici olduğu belirlenmektedir. O blok için uyumlama işlemi seçilmiş olan bit düzlemini kullanarak NNMP yardımı ile yapılmaktadır.

$$M_p(i,j) = \left\{ \begin{array}{l} 1 \leftarrow \exists(m,n) \in \{(\pm 1, 0), (0, \pm 1)\}, \\ BC_p(i,j) \oplus BC_p(i+m,j+n) \\ 0 \leftarrow \text{aksihalde} \end{array} \right\} \quad (3.14)$$

Bu eşitlikte p değeri bit düzlemini, BC ise ikili kodlanmış imge çerçevesini göstermektedir. Bir blokta bulunan piksellere en değerlikli bit düzleminden başlanarak sağındaki, solundaki, üstündeki ve altındaki piksel değerlerinin işlem yapılan bit düzlemindeki karşılıklarıyla XOR işlemi uygulanmaktadır. Herhangi bir karşılaştırma işleminin sonucunun 1 çıkması o pikselin işlem yapılmakta olan bit düzleminde yanındaki pikselden farklılaştığı anlamına gelmektedir. Bu işlem seçilen blok için sırasıyla bütün bit düzlemlerinde yapılmaktadır. O blokta bulunan piksel değerleri için ilgili bit düzleminde elde edilen XOR işlemi sonuçları M matrisine yazılmaktadır. M matrisinde bulunan 1 değerleri önceden belirlenmiş bir eşik değeriyle karşılaştırılmakta ve 1 değerlerinin sayısının belirli bir eşik değerinden yüksek olması durumunda o blok için işlem yapılmış olan bit düzlemi seçilmektedir. Sonrasında ise NNMP kriteri yardımı ile o blok için seçilmiş olan bit düzleminde uyumlama işlemi

gerçekleştirilmektedir. Ekran içerik görüntülerinin yapısı sayesinde bit düzlemlerinde gürültü vb. istenmeyen etkiler olmadığı için etkin bir şekilde ayırt edici bit düzlemleri seçilebilmektedir. Şekil 3.8’de Big Buck Bunny dizisinden alınan örnek bir görüntü için bit düzlemi seçme işlemi sonuçları verilmiştir.

[34]’te ağırlıklandırma temelli, iyileştirilmiş 2BT yöntemi temelli bir yaklaşım sunulmuş ve bu yaklaşımda uyumlama ölçütünün hesabında yapılan değişiklik ile hareket kestirimi başarımının artırılması hedeflenmiştir.

[35]’de C-1BT yönteminde önerilen kısıt maskesine ağırlık faktörü eklenerek iki bit kullanan kısıt maskesi önerilmiş ve böylelikle toplamda üç bit uzayı kullanılarak HK başarımının artırılması hedeflenmiştir. Bu yöntemin başarımı C-1BT yöntemine göre daha yüksek olmakla birlikte hesapsal karmaşıklığı da fazladır.

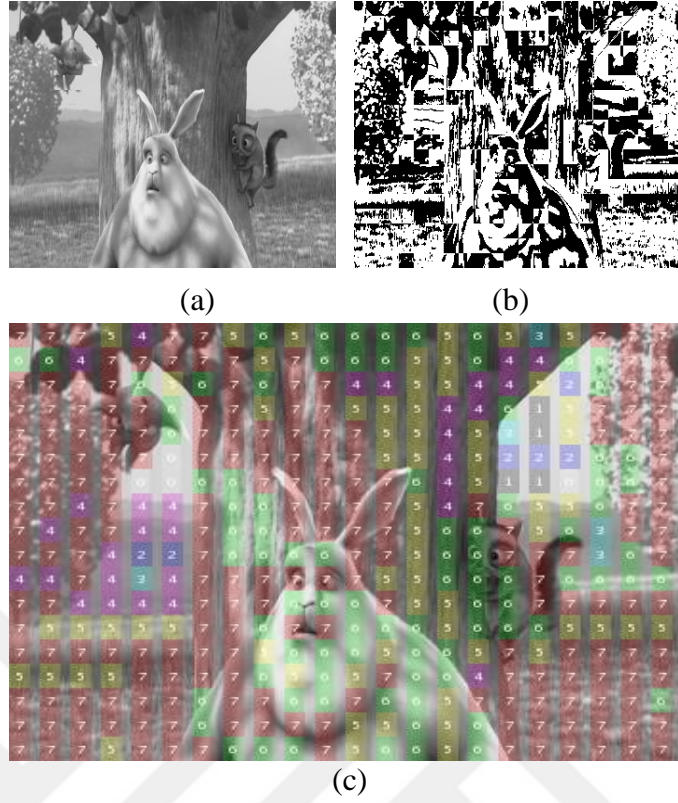
[36]’da hareket kestirimi başarımını arttırmak amacıyla C-1BT için daha geniş bir dinamik aralık sağlayan NNMP ölçütü önerilmiştir.

[37]’de bulanık mantık kullanılarak ikili dönüşüm işlemi gerçekleştirilmiştir. Bu çalışmada bulanık mantık yardımıyla en uygun dönüşümü gerçekleştirmek amaçlanmıştır.

3.1.4. Tahminsel arama

Hareket kestirimi algoritmalarında başlangıç noktasının belirlenmek amacıyla zamansal veya uzamsal komşu bloklardan yararlanılabilmektedir. O an ilgilenilen bloğun yakınındaki bloklara benzer bir hareket gösterme ihtimaline dayanarak arama işleminin hesapsal karmaşıklığı azaltılabilmektedir. [38] ve [39]’deki çalışmalarda bu teknik benimsenmiştir.

Bununla birlikte HM’de kullanılan TZS algoritmasında da arama başlangıç noktasını seçmek için zamansal ve uzamsal komşu blokların hareket vektörlerinden yararlanılarak, en düşük SAD değerini veren konum başlangıç noktası olarak belirlenmektedir.



Şekil 3.8. Big Buck Bunny a) Orjinal çerçeve b) Bloklar için seçilmiş bit düzlemleriyle oluşturulmuş çerçeve c) Bloklar için seçilmiş bit düzlemleri

3.1.5. Hiyerarşik arama

Hiyerarşik arama tekniklerinde genellikle daha düşük çözünürlüklerde kabaca hareket vektörü hesaplandıktan sonra, yüksek çözünürlüklerde doğrulama işlemi gerçekleştirilmektedir. Bu teknikler aynı zamanda piramit yapısı olarak da isimlendirilmektedir. İlk aşamada düşük çözünürlüklerde eşleme için yeterli detay olmamasından dolayı yerel bir minimum noktasına takılınması yöntemin başarımını düşürebilmektedir. [40] ve [41]'de bu tekniği kullanan iki yöntem önerilmiştir.

3.1.6. Hızlı tam arama

Bu yaklaşımlarda tüm aday hareket vektörleri test edilirken hesapsal karmaşıklığı azaltmak amacıyla çeşitli eleme yaklaşımları, erken sonlandırma yaklaşımları kullanılabilir.

[42]'de önerilmiş olan SEA'da (Successive Elimination Algorithm – Ardışık Eleme Algoritması) piksellerin toplamlarının farkının, piksellerinin farkları toplamından her

zaman küçük olduğu varsayımına dayanan bir yöntem önerilmiştir. Bu çalışma piksellerin toplamlarının farkını hesaplamak için tümlev imgelerden yararlanılmıştır. Tümlev imge sayesinde 3 toplama ve 1 çıkarma işlemiyle blok toplamları hesaplanmaktadır. İlk etapta blokların toplamlarının farkı hesaplanmakta eğer hesaplanan bu değer o ana kadar hesaplanmış olan en iyi SAD değerinden daha büyükse o hareket vektörü için SAD hesaplama işlemi atlanmaktadır.

PDE (Partial Distortion Elimination – Kısmi Bozulma Elemesi) [43] algoritmasında hesapsal karmaşıklığı azaltmak için bloğun yarısına ait SAD değeri hesaplandıktan sonra hesaplanan değer 2 ile çarpılarak elde edilen değer minimum SAD değerinden küçükse ilgili hareket vektörü için SAD hesaplama işlemi sonlandırılabilir. Bununla birlikte bloğa ait farklı boyutta parçalar için de yöntem genişletilebilmektedir [44].

3.1.7. Karma yöntemler

Üst başlıklarda verilmiş olan yöntemler tek başlarına kullanılabildiği gibi bu yöntemlerin birleştirilmesiyle yapılmış olan birçok çalışma da literatürde mevcuttur. Örneğin HEVC'deki TZS algoritmasında elmas bir arama örüntüsüyle arama noktalarının sayısı azaltılmakta, alt-örnekleme yapılarak uyumlama kriteri hesabı kolaylaştırılmakta ve bunlarla birlikte zamansal-uzaysal komşu bloklar için hesaplanmış olan hareket vektörleri kullanılarak arama işlemine başlangıç noktası belirlenmektedir.

[45]'te önerilmiş olan yöntemde 8 bitlik piksel değerlerinin en değerlikli 4 biti kullanılarak bir uyumlama kriteri hesaplanması önerilmiştir. MSAD (Modified Sum of Absolute Differences – Modifiye Edilmiş Mutlak Farklar Toplamı) olarak isimlendirilmiş olan bu kriter SEA algoritmasına benzer şekilde her konuma SAD hesabını atlamak için kullanılmıştır. Önerilmiş olan yöntem tam arama ve hızlı hareket yaklaşımlarıyla kullanılabilmektedir. Bununla birlikte yöntem SIMD (Single Instruction Multiple Data – Tekli Komut Çoklu Veri) kullanılarak da çalıştırılmış, elde edilen deneysel sonuçlar yöntemin açık döngü şemada PSNR değerinde önemli bir kayıp olmadan %90'a kadar hız kazandırabildiğini göstermiştir.

[46]'da gerçekleştirilmiş olan çalışmada asimetrik arama kalıpları kullanan bir yöntem önerilmiştir. Başlangıçta beş noktaya sahip küçük elmas örüntü kullanılırken, en düşük SAD değerinin elde edildiği noktanın etrafına arama işlemi yönlendirilerek, yönelime göre her konum için farklı örüntüler kullanılmıştır. Bununla birlikte çalışmada bloğun solunda bulunan komşusu için hesaplanmış olan hareket vektörü de test edilerek arama için başlangıç noktası belirlenmiştir.

[47]'de önerilmiş olan yöntemde elmas örüntünün ve PDE ile birlikte kullanılması önerilmiştir. Bu yöntemde örüntünün merkezinde bulunan noktadaki blok için SAD değeri hesaplandıktan sonra diğer noktalar için SAD hesaplama işlemi yapılırken minimum SAD değerinin aşıldığı satırda hesaplama işlemi sonlandırılmaktadır. Yöntem açık çevrim olarak uygulanmış olup PSNR değerinde önemli bir kayıp olmadan TSS algoritmasına göre %30 hız kazancı elde edilmiştir.

[48]'de C-1BT temelli hareket kestirimi yaklaşımı altıgen bir arama örüntüsüyle gerçekleştirilmiştir. [49]'da ise C-1BT temelli hareket kestirimi gerçekleştirilirken (0,0) hareket vektöründeki NNMP değerinin küçük olması hareketin yakın bir bölgede olduğu anlamına geldiğinden bu değerın büyüklüğüne göre arama penceresinin büyüklüğü belirlenmiştir.

[50]'de HEVC için bir hareket kestirim algoritması önerilmektedir. Bu çalışmada da hareket vektörü tahminiyle başlangıç noktası belirlenerek [46]'daki yönteme benzer şekilde yönelimsel bir arama algoritması kullanılmaktadır. Başlangıçta kullanılan arama örüntüsü 9 noktalı bir dikdörtgen şeklindedir. 9 noktalı örüntüde en düşük SAD değerinin elde edildiği hareket vektörüne göre farklı örüntülerle yönelimsel olarak hareket kestirimi işlemine devam edilmektedir. Bununla birlikte başlangıçta 13 noktaya sahip çapraz bir arama örüntüsü kullanılarak en düşük SAD değerinin elde edildiği nokta etrafında da arama yapılmaktadır. İyi yaklaşımın sonucu birlikte değerlendirilerek en düşük SAD değerinin elde edildiği konum hareket vektörü olarak belirlenmektedir. Yöntem HM'e uygulanmış olup toplam kodlama süresinde TZS'ye göre %23'e kadar iyileştirme sağlayabilmektedir.

[51]'de gerçekleştirilmiş olan çalışmada komşu blokların hareket vektörleri ve kodlama maliyetlerine göre PU'lar az hareketli PU, orta hareketli PU ve karmaşık hareketli PU olmak üzere 3 sınıfa ayrılarak her bir sınıf için farklı bir arama yaklaşımı

kullanılmıştır. Az hareketli PU'lar için küçük elmas arama örüntüsü kullanılırken, karmaşık hareketli PU'lar için [50]'dekine benzer bir şekilde arama penceresinde daha büyük alanı kapsayan çapraz bir örüntü kullanılmıştır. Geliştirilmiş olan algoritma HM'e uygulanmış olup bit oranında küçük bir bozunumla toplam kodlama süresinde TZS algoritmasına göre %20'ye kadar kazanç sağlayabilmektedir.

[52]'de gerçekleştirilmiş olan çalışmada HEVC'deki SAD ve SSD hesaplarının GPU (Graphics Processor Unit – Grafik İşlemci Birimi) kullanılarak gerçekleştirilmesi önerilmiştir. Fermi mimarisi kullanan bir GPU sayesinde arama işlemleri sırasında SAD hesapları paralel bir şekilde gerçekleştirilerek hareket kestirimi hesabında, belirli dizilerde ortalama %0,6 bit kaybıyla %23'e kadar hızlanma sağlanmıştır.

[53]'te önerilmiş olan yöntemde SAD hesabının GPU ile paralel bir şekilde yapılması önerilmiştir. Önerilmiş olan yöntemde HEVC referans yazılımında tam arama yaklaşımına göre %0,9 bit kaybıyla ortalama %48,7 hız kazancı sağlanmıştır.

[54]'te kodlama süresini azaltmak için SAD hesaplama ve dönüşüm işlemlerinin çok çekirdekli işlemci desteği ile SIMD kullanarak paralel bir şekilde gerçekleştirilmesi önerilmiştir. Yapılmış olan optimizasyonlar sayesinde HEVC referans yazılımındaki tam arama yaklaşımına göre bit kaybı olmadan %33,56 hız kazancı sağlanmıştır. Dilimlerin uyarlamalı bir şekilde bölünmesi ve paralel bir şekilde işlenmesi ile de RA konfigürasyonu için toplam kodlama süresinde %5,83 bit kaybıyla %86,99 hızlanma sağlanmıştır.

[55]'te gerçekleştirilen çalışmada hareket kestirimi bir istatistik problemi olarak ele alınmıştır. Bu yöntemde komşu blokların artıklık verilerinden yararlanarak hareket kestirimi için maliyet hesabı fonksiyonunda güncellemeler gerçekleştirilerek arama işleminde erken sonlandırma sağlanmıştır. Geliştirilmiş olan yöntem HM'deki tam arama ve TZS yöntemleriyle birlikte çalıştırabilmektedir. Elde edilmiş olan deneysel sonuçlar belirli diziler için ortalama %0,97 bit oran bozunumuyla TZS algoritmasına göre %13 hızlanılabildiğini göstermiştir.

[56]'da TZS algoritmasında ilk aşamada kullanılan çok boyutlu elmas örüntüye benzer bir şekilde çok boyutlu bir altıgen örüntü kullanılması önerilmiştir. Doğrulama işleminde kullanılan elmas örüntü de standart altıgen örüntüyle değiştirilmiştir.

Bununla birlikte TZS’de yerel minimum değerlerine takılmamak için gerçekleştirilmiş olan taramalı arama işlemi gerçekleştirilmemiştir. Ayrıca çalışmada 0 ve 1 numaralı CTU derinliklerinde asimetrik PU modlarının bit oranını ve PSNR değerini fazla etkilemediği ifade edilerek bu derinliklerde sadece simetrik PU modları kullanılmıştır. Önerilmiş olan yöntem TZS algoritmasına göre %55 daha az noktada SAD hesabı yapılmasını sağlamaktadır.

[57]’de önerilen yöntemde de TZS’deki elmas arama yaklaşımı yerine altıgen bir örüntü kullanılması önerilmiştir. Bununla birlikte önceki çerçevelerde yer alan blokların kodlama maliyetleri değerlendirilerek PU seviyesinde erken sonlandırma işlemi de gerçekleştirilmiştir.

[58]’de HM’de TZS algoritmasının gerçekleştirildiği sırada arama penceresinin sınırlarının daraltılarak hesapsal karmaşıklığın azaltılması önerilmiştir. Arama penceresinin sınırlarını daraltmak amacıyla komşu blokların hesaplanmış olan hareket vektörlerinden yararlanılmaktadır. Önerilmiş olan yöntem arama noktalarının TZS algoritmasına göre %60’a kadar azaltılmasını sağlayabilmektedir.

3.2. Mod Seçimi

HEVC’de CTU yapısının oluşturulması sırasında çeşitli derinliklerde PU modları için çerçeve-içi ve çerçeveler-arası kestirim işlemleri gerçekleştirilerek uygun CTU yapısına karar verilmektedir. Uygun derinliğin ve PU modunun uzamsal veya zamansal yöntemlerle kestirilmesi hesapsal karmaşıklığın azaltılmasını sağlamaktadır. Mod seçimi için HEVC referans yazılımının önerdiği yöntemler olmakla birlikte literatürde de bu konuda birçok çalışma mevcuttur.

HM 3.2 sürümünden itibaren mod seçimi işlemleri için referans yazılıma çeşitli yöntemler uygulanmıştır. ESD (Early Skip Detection – Erken Atlama Tespiti) [59] yönteminde ilk olarak $2N \times 2N$ modunda hareket kestirimi gerçekleştirildikten sonra artıklık verisi 0 olarak bulunmuşsa ve bulunmuş olan hareket vektörü komşu hareket vektörüne eşitse asimetrik ve simetrik PU modları için herhangi bir işlem yapılmamaktadır. CBF (Coded Block Flag – Kodlanmış Blok Bayrağı) [60] yaklaşımında herhangi bir PU modunda artıklık verisi 0 olarak bulduysa o derinlikteki diğer PU modları için işlemler sonlandırılmaktadır. ECU (CU Early

Termination – CU Erken Sonlandırma) [61] yaklaşımında ise tüm PU modları denendikten sonra en az maliyet atlama modunda elde edilmişse derinlik seviyesinde sonlandırma işlemi gerçekleştirmektedir. Bu yöntemler referans yazılımda kapalı olup kullanıcıyı isteğine bağlı olarak açılabilir. Bu yöntemlerin birlikte kullanımıyla ortalama %1,9 bit kaybıyla toplam kodlama süresi %52,7 azaltılabilmektedir.

[62]'de gerçekleştirilmiş olan çalışmada komşu bloklar için hesaplanmış olan CU derinliğinden yararlanılarak kestirim yapılan CU derinliğinin belirlenmesi amaçlanmıştır. Bu sırada komşu blokla aday bloğun korelasyonundan yararlanmıştır. Korelasyon değerinin büyük olması aday blok için de benzer bir derinlik kullanılabilirdi anlamına gelmektedir. Önerilmiş olan yöntem %1,29 bit kaybıyla toplam kodlama süresini %42'ye kadar azaltabilmektedir.

[63]'te CU boyutlarını belirlemek için Bayes teoremi tabanlı bir algoritma önerilmiştir. Bu çalışmada kodlama maliyetlerinin istatistiksel analizi yapılarak CU'lar için erken sonlandırma ve atlama moduna karar verme işlemleri gerçekleştirilmiştir.

[64]'te önerilmiş olan yöntemde homojen bölgelerin daha büyük, karmaşık bölgelerin ise daha küçük boyutta CU'larla kodlandığı gözönüne alınmış, bu doğrultuda piksellerin birbiriyle ilişkileri değerlendirilerek homojen bölgelerde alt derinliklere inilmesini engelleyerek hesapsal karmaşıklığı azaltan bir yöntem önerilmiştir. Geliştirilmiş olan yöntem RA konfigürasyonu için ortalama %1,4 bit kaybıyla toplam kodlama süresini %49,6 azaltabilmektedir.

[65]'te CU derinliklerini belirlemek için makine öğrenmesi tabanlı bir yöntem önerilmiştir. SVM (Support Vector Machine – Destek Vektör Makinesi) tabanlı sınıflandırıcı için kodlama maliyeti ve hareket vektörleri giriş verisi olarak kullanılmıştır. Önerilmiş olan yöntem %1,98 bit kaybıyla hesapsal karmaşıklığı ortalama %51 azaltabilmektedir.

[66]'da gerçekleştirilmiş olan çalışmada CU derinlik seviyesine erken karar vermek amacıyla $2N \times 2N$ boyutundaki bloklar için gerçekleştirilen arama işleminin sonuçları, komşu blokların derinlik bilgileri ve [59]'da önerilmiş olan CBF yaklaşımının sonuçları birlikte değerlendirilmiştir. Geliştirilmiş olan yöntem %1,1 bit kaybıyla hesapsal karmaşıklığı ortalama %51 azaltabilmektedir.

[67]'de daha önce kodlanmış olan CU'ların derinlik seviyelerinin standart sapması kullanılarak kodlanacak olan CTU için bir derinlik aralığı belirlenmektedir. Önerilmiş olan yöntem ortalama %1,04 bit kaybıyla toplam kodlama süresinde %42 azalma sağlamıştır.

[68]'de uygun PU modlarını seçmek için bloklardaki kenar bilgilerinin kullanılması önerilmiştir. Bu amaçla Sobel operatörü kullanılarak CU'larda bulunan kenarlar tespit edilmiş, bu kenarların yönelimlerine göre uygun PU moduna karar verilmiştir. Gerçekleştirilmiş olan çalışma %1,89 bit kaybıyla kodlama süresini %39'a kadar azaltabilmektedir.

[69]'da sunulmuş olan çalışmada, 8×8 ve 64×64 boyutlarındaki CU'larda asimetrik PU modlarının kullanılmamış, diğer boyutlarda ise simetrik PU modları için maliyet hesabı gerçekleştirildikten sonra en iyi sonucu veren PU moduyla ilgili asimetrik modlar için kestirim işlemi gerçekleştirilmiştir. Önerilmiş olan yöntem %1,3'e kadar bit kaybıyla toplam kodlama süresini %50'ye kadar azaltabilmektedir.

[70]'te sunulmuş olan yöntemde uygun CU derinliğinin bulunması ile birlikte uygun PU modunun bulunması üzerinde de çalışılmıştır. Bu amaçla $2N \times 2N$ boyutu için kestirim yapıldıktan sonra artıklık verisinin PU modlarıyla ilişkili bölgelere göre dağılımı değerlendirilmiş, bu dağılıma göre uygun modlara karar verilmiştir. Önerilmiş olan yöntem %4,5 bit kaybıyla toplam kodlama süresini ortalama %80 azaltabilmektedir.

[71]'de sunulmuş olan yöntemde ise $2N \times 2N$ boyutundaki CU'lar için uygun modlara karar verme işleminin $N \times N$ boyutundaki PU'larda kestirim yapıldıktan sonra yapılması önerilmiştir. Bu yaklaşımda $N \times N$ boyutundaki PU'lardan elde edilmiş olan hareket vektörü sonuçlarına göre hangi modlar için maliyet hesabı yapılacağına karar verilmektedir.

4. TEZ ÇALIŞMASI KAPSAMINDA GELİŞTİRİLMİŞ OLAN HAREKET KESTİRİMİ VE MOD SEÇİMİ YÖNTEMLERİ

Bu tez çalışmasında, video kodlama işlemlerindeki hesapsal karmaşıklığı düşürmek amacıyla, uygun bir bit-oran/bozunum ve hız dengesi sağlayan hareket kestirimi ve mod seçimi yaklaşımları önerilmiştir. Bu amaçla 4 ana başlık altında çeşitli çalışmalar gerçekleştirilmiştir.

Tez çalışmasının ilk aşamasında, ekran içerik görüntüleri için düşük-bit derinliği temelli özgün bir hareket kestirim yöntemi geliştirilmiştir.

İkinci aşamasında, 1BT, C-1BT ve SGC temelli hareket kestirim yöntemleri HEVC referans yazılımına uygulanarak bu yöntemlerin başarımı değerlendirilmiştir.

Sonraki aşamada, HEVC referans yazılımındaki hesapsal karmaşıklığı düşürmek amacıyla, referans yazılım üzerinde en iyi başarımı vermiş olan SGC yöntemini temel alan hızlı tam arama ve seyrek arama yaklaşımları geliştirilmiştir.

Dördüncü ve son aşamada ise HEVC'deki CTU yapısıyla birlikte hesapsal yükü artmış olan mod seçimi işlemlerinin hesapsal yükünü düşürmek amacıyla blok benzerliğine/homojenliğine dayalı özgün bir mod seçimi yaklaşımı geliştirilerek referans yazılım üzerinde uygulanmıştır.

4.1. Ekran İçerik Görüntüleri İçin Uyarlamalı Bit Düzlemi Seçme Tabanlı Hareket Kestirimi

4.1.1. Giriş

Animasyon görüntüleri, uzak masaüstü uygulamaları, kablosuz ekranlar, bulut oyun servisleri gibi uygulamalarla birlikte ekran içerik görüntülerini kodlanması video kodlama uygulamalarının önemli bir parçası haline gelmiştir. Ekran içerik görüntüleri geleneksel kamera ile edilmiş görüntülere göre çeşitli farklılıklar içerdiğinden HEVC standart referans yazılımı ile birlikte ekran içerik görüntülerinin kodlanması için ayrı bir sürüm sunulmuştur.

Ekran içerik görüntülerinin genellikle gürültüsüz olması, daha keskin kenar bilgileri içermeleri gibi karakteristik özellikleri kodlama verimini arttırmak için kullanılabilir.

Tez çalışmasının bu bölümünde, ekran içerik görüntüleri için tasarlanmış, düşük-bit temelli bir hareket kestirimi yaklaşımı sunulmuştur. Önerilmiş olan yöntem, en değerlikli bit düzlemindeki detayları inceleyerek kullanılacak gray kodlanmış bit düzlemlerine karar vermektedir. Uyumlama işlemi için 1-bit derinliğinde imgeler kullanılmaktadır. Bununla birlikte uyumlama işlemi sırasında kullanılan bit düzlemleri ağırlıklandırılarak uyumlama işleminin başarımı artırılmıştır.

4.1.2. Önerilen yöntem

[6]'da önerilmiş olan SGC yaklaşımında Şekil 3.7'deki şablon kullanılarak pikseller için bit düzlemi seçilerek ikili bir imge oluşturulmakta ve uyumlama işlemi bu imge kullanılarak NNMP kriteri ile gerçekleştirilmektedir. [33]'te önerilmiş FBMEESC yaklaşımında ise Eşitlik (3.11) kullanılarak bir köşe haritası (M) oluşturulmakta ve M bloğuna göre tek bir ikili bit düzlemi seçilerek uyumlama için kullanılmaktadır. Test dizileriyle yapılmış olan çalışmalar sonucunda FBMEESC yaklaşımındaki bit seçme mekanizmasından dolayı alt bit düzlemlerine inildiğinde, yüksek değerlikli bit düzlemleri hiç değerlendirmeye alınmadığından dolayı hareket kestirimi başarımının düştüğü görülmüştür.

SGC yaklaşımının ise genel olarak FBMEESC yaklaşımından daha iyi sonuçlar verdiği gözlenmekle birlikte bazı dizilerde SGC yaklaşımındaki 5. 6. ve 7. düzlemlerle birlikte bazı diziler için 4. düzlemi de kullanmanın daha iyi hareket kestirimi sonuçları verdiği görülmüştür. Şekil 4.1'de bu şablon verilmiştir.

Şekil 4.2'de ise Big Buck Bunny ve Google Maps dizilerinin en değerlikli 4 bit düzlemi verilmiştir. Bu şekillerden de görüldüğü üzere yakın piksel değerlerinin bulunduğu bölgelerde, yüksek değerlikli bit düzlemlerinde yeterli detay bulunmamaktadır. Bu sebepten dolayı da Şekil 4.1'de verilmiş olan şablonun bazı dizilerde daha iyi bir başarımla sağladığı değerlendirilmektedir.

g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7
g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6
g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5
g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4
g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7
g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6
g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5
g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4
g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7
g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6
g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5
g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4
g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7
g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6
g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5
g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4
g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7
g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6
g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5
g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4	g5	g6	g7	g4

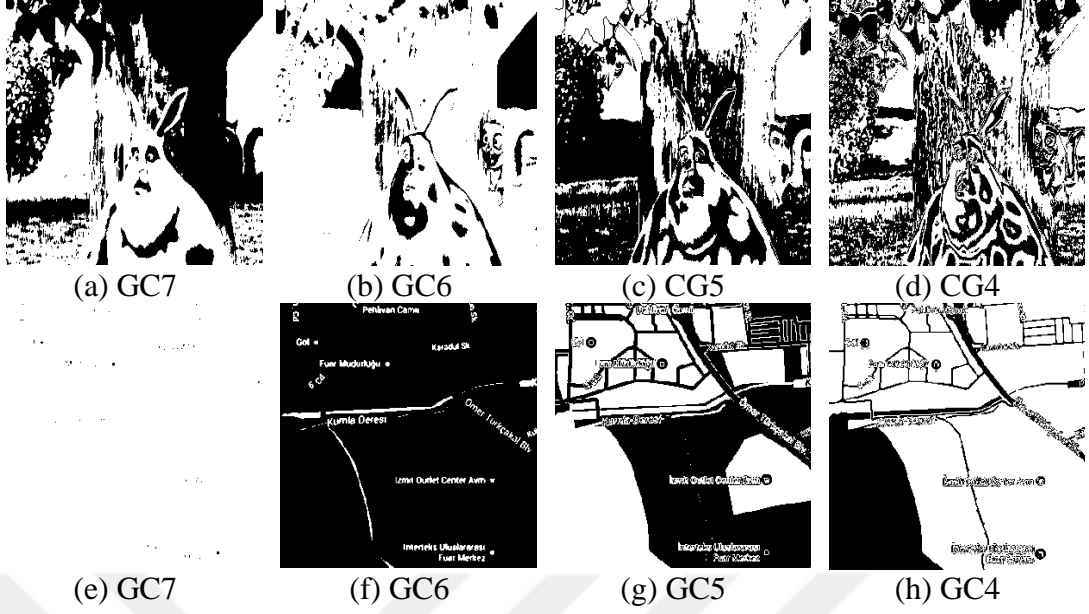
Şekil 4.1. 4. 5. 6. ve 7. bit düzlemlerini kullanan şablon

Elde edilen bu veriler doğrultusunda, uyarlamalı bir şekilde Şekil 3.6 ve Şekil 4.2'deki şablonları kullanmanın hareket kestirimi başarımını arttıracığı açıktır. Bu kararı verebilmek için Eşitlik (3.14) kullanılmıştır. Fakat FBMESC yaklaşımından farklı olarak bu işlem sadece en değerlikli bit düzleminde yapılmıştır. Oluşturulmuş olan M matrisindeki toplam 1 sayısı deneysel çalışmalar sonucu belirlenmiş olan eşik değerinden büyükse (A modu) Şekil 3.6'daki şablon, diğer durumda (B modu) ise Şekil 4.1'deki şablon kullanılmıştır. Şekil 4.3'de Big Buck Bunny dizisinden örnek bir görüntü ve bu görüntü için seçilmiş olan modlar verilmiştir.

Dönüşüm işlemi gerçekleştirildikten sonra ise bit düzlemleri ağırlıklandırılarak uyumlama işlemi gerçekleştirilmiştir. Uyumlama işlemi için kullanılan denklem Eşitlik (4.1)'de verilmiştir.

Bu eşitlikle SG dönüşüm işlemi gerçekleştirilmiş imgeyi, p ise şablona göre işlem yapılan pikselin düzlemini temsil etmektedir. s değeri ise seçme mekanizmasına göre 4 veya 5 olarak belirlenmektedir.

$$WSGC-MC(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}2^{p-s}\{SG_p^t(i,j)\oplus SG_p^{t_{ref}}(i+m,j+n)\} \quad (4.1)$$



Şekil 4.2. Big Buck Bunny ve Google Maps dizileri için Gray-kodlanmış bit düzlemleri GC7, GC6, GC5, GC4 a) - d)Big Bug Bunny and e) - h)Google Maps



Şekil 4.3. a) “Big Buck Bunny” dizisinden alınmış örnek bir görüntü b) Görüntüdeki bloklar için seçilmiş modlar

Şekil 4.3’ten görüldüğü üzere detaylı bölgeler için A durumu seçilmekteyken, daha az detay içeren bölgelerde B durumu seçilmektedir.

4.1.3. Deneysel sonuçlar

Geliştirilmiş olan yöntemin başarımını ölçmek amacıyla orijinal ve kestirilmiş çerçeveler arasındaki PSNR değerleri kullanılmıştır. Yapılan çalışmada Big Buck Bunny (BBB), Elephants Dream (ED) görüntü dizilerinden alınan 250 adet görüntü çerçevesi ve yöntemlerin başarımını değerlendirmek amacıyla oluşturulan, Pro Evolution Soccer (PES), Counter Strike (CS) oyunlarından ve Google Maps (GM)

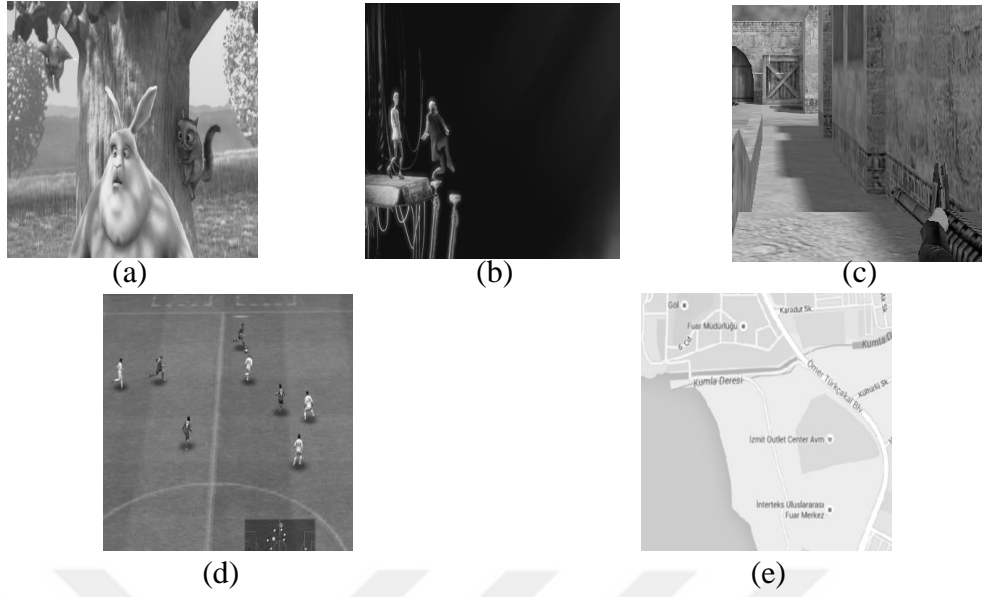
uygulamasından alınan 200'er adet görüntü çerçevesi kullanılmıştır. Bu dizilerden alınan örnek görüntüler Şekil 4.4'de verilmiştir. Hareket vektörleri ise $[-16,16]$ arama bölgesinde 16×16 bloklar için hesaplanmıştır.

Big Buck Bunny isimli animasyon görüntü dizisinden alınan görüntüler genel olarak yerel hareketlere, Elephants Dream isimli animasyon görüntü dizisinden alınan görüntüler ise bütünsel ve yerel hareketlere sahiptir. Animasyon görüntüsü olmalarına rağmen bu görüntülerde gölgelendirme vb. detaylar da mevcuttur. Counter Strike ve Pro Evolution Soccer isimli oyunlardan alınan görüntüler de bütünsel ve yerel hareketlere sahip olmakla birlikte Pro Evolution Soccer isimli oyun nispeten daha tutarlı bir arka plan barındırmaktadır. Google Maps uygulamasından alınan görüntülerde ise bütünsel hareketler ağırlıklı ve genellikle arka planda sabit piksel değerleri mevcuttur.

Yapılan deneysel çalışmalar sonucunda elde edilen PSNR değerleri Tablo 4.1'de verilmiştir. Şekil 4.1'de verilmiş olan şablonun ve ağırlıklandırma işleminin başarımını görmek amacıyla SGC yaklaşımı sadece Şekil 4.1'deki şablon kullanarak çalıştırılmış, her iki şablonu kullanılarak da ağırlıklandırma işlemi de gerçekleştirilmiştir (WSGCME(5-6-7), WSGCME(4-5-6-7)).

En yüksek PSNR değerlerini NTB=4 ve NTB=5 değerlerini kullanan T-GCBPM yönteminin verdiğini görülmüştür. Fakat bu yöntemde NTB=4 ve NTB=5 değerleri için sırasıyla toplamda 4 ve 3 bit düzlemi uyumlama hesabında kullanıldığından dolayı hesapsal karmaşıklığı diğer yöntemlere göre çok daha fazladır.

C-1BT temelli hareket kestiriminin beklendiği şekilde 1BT temelli hareket kestirimine göre daha iyi sonuçlar verdiği görülmüştür. C-1BT temelli hareket kestiriminin Pro Evolution Soccer ve Google Maps dizilerinde NTB=6 için T-GCBPM yöntemine göre iyi sonuç verdiği Tablo 4.1'den gözlenmektedir. Bu diziler diğer dizilere göre daha düz arka planlar içerdiklerinden NTB=6 değerinin uyumlama için yeterince piksel çıkartmadığı görülmüştür. Bu tarz görüntülerde C-1BT yönteminin veya NTB için daha küçük değerler kullanılarak T-GCBPM yönteminin kullanılması gerektiği düşünülmektedir.



Şekil 4.4. Test dizilerinden alınmış örnek görüntüler a) Big Buck Bunny b) Elephants Dream c) Counter Strike d) Pro Evolution Soccer e) Google Maps

Tablo 4.1. Önerilen yöntemin literatürdeki düşük bit gösterimi temelli yöntemler karşılaştırılması (PSNR)

Yöntem	BBB	ED	CS	PES	GM	Ortalama
1BT [4]	34,63	29,00	24,37	29,83	30,29	29,62
C-1BT [5]	34,97	29,36	24,83	30,15	30,42	29,95
T-GCBPM [31] (NTB = 6)	35,59	30,78	25,10	29,81	27,50	29,76
T-GCBPM [31] (NTB = 5)	36,00	30,93	25,85	30,63	32,11	31,10
T-GCBPM [31] (NTB = 4)	36,00	30,90	25,91	31,33	32,60	31,35
FBMESC [33]	34,39	28,76	24,30	29,31	31,12	29,58
SGCME [6]	35,69	30,56	25,55	30,47	31,47	30,75
SGCME (4-5-6-7)	35,40	30,30	25,38	30,91	31,86	30,77
WSGCME (5-6-7)	35,74	30,72	25,65	30,39	31,61	30,82
WSGCME (4-5-6-7)	35,76	30,59	25,66	30,93	32,11	31,01
Önerilen Yöntem	35,80	30,69	25,70	30,92	32,14	31,05

FBMESC yönteminin ise genel olarak 1BT dönüşümü temelli hareket kestiriminden de düşük sonuçlar verdiği gözlenmekle birlikte özellikle Google Maps dizisinde iyi sonuçlar verdiği gözlenmiştir. Bu dizi genellikle sabit piksel değerlerine sahip, bütünsel hareketin ağırlıklı olduğu bir dizidir. Bu tarz dizilerde değerlikli bit düzlemleri yeteri kadar detay vermediğinden dolayı alt bit düzlemlerinin kullanılması daha iyi sonuçlar sağlayabilmektedir. Yoğun hareket ve detaylar içeren dizilerde ise hareket eden bölgeler yakın değerlere sahip olduğunda alt bit düzlemleri seçildiğinden dolayı daha kötü sonuçlar verdiği gözlenmiştir.

SGCME yönteminin ise görece olarak düşük işlem yüküyle birlikte diğer yöntemlere göre daha iyi sonuçlar verdiği görülmüştür. Elde edilen PSNR değerleri NTB=5

değerini kullanan T-GCBPM'ye göre daha düşüktür fakat bu durumda T-GCBPM yönteminin hesapsal karmaşıklığı çok fazladır. Belirli bit düzlemlerinin 1 bit çözünürlüğünde kullanılmasının SGCME yöntemi için avantaj oluşturduğu gözlenmiştir.

Bu tez çalışmasında geliştirilmiş olan yöntemin sadece NTB=4 ve NTB=5 değerlerini kullanan T-GCBPM dışında daha iyi sonuçlar verdiği gözlenmiştir. NNMP hesabında bit düzlemlerini ağırlıklandırma işlemi ve kullanılan şablonlar arasında seçme işlemi elde edilen sonuçların daha iyi olmasını sağlamıştır.

Bununla birlikte Tablo 4.1'den SGCME, SGCME(4-5-6-7), WSGCME(5-6-7) ve WSGCME(4-5-6-7) yöntemleri incelendiğine tez kapsamında önerilen yöntemin hepsinden daha iyi sonuçlar verdiği görülmüştür. Bu sonuçlar seçme ve ağırlıklandırma mekanizmalarının birlikte doğru bir şekilde çalıştığını göstermektedir.

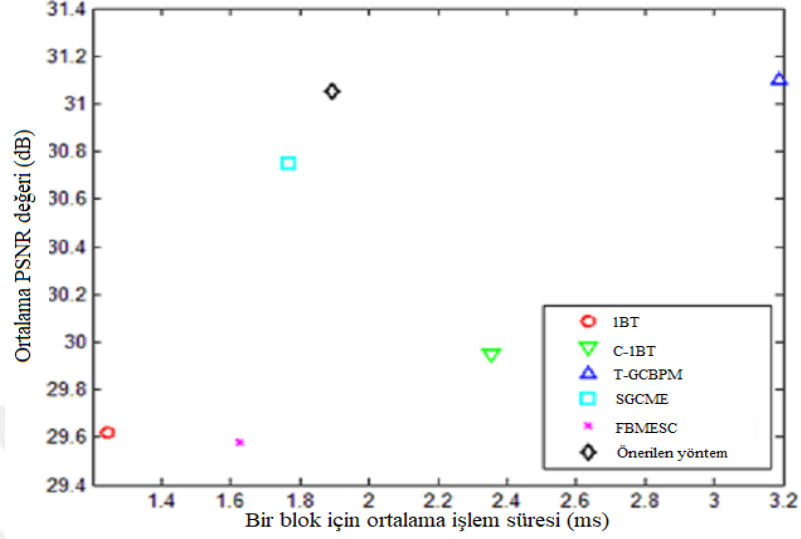
Tablo 4.2'de ise hesapsal karmaşıklığı değerlendirmek amacıyla piksel başına düşen işlem yükleri verilmiştir. Bu tablodan da görüldüğü üzere önerilmiş olan yöntem dönüşüm aşamasında 1BT, C-1BT ve FBMESC yaklaşımlarına göre daha düşük hesapsal karmaşıklığa sahiptir., Uyumlama aşamasındaki işlem yükü ise C-1BT ve SGCME ile benzer olmakla birlikte T-GCBPM'ye göre çok daha düşüktür.

Tablo 4.2. Önerilen yöntemin ve literatürdeki yöntemlerin piksel başına düşen işlem yükleri

Yöntem	Dönüşüm					Uyumlama						
	Toplama	Bölme	Kayıtma	Çıkarma	Karşılaştırma	Boolean İşlem.	Bellek Erişimi	Boolean İşlem.	Kayıtma	Toplam	Karşılaştırma	Bellek Erişimi
1BT [4]	25	1	-	-	1	-	28	1	-	1	-	2
C-1BT [5]	16	-	1	1	3	-	22	3	-	1	-	4
T-GCBPM [31]	-	-	3	-	-	3,67	6	3	2	3	-	6
(NTB = 5)												
SGCME [6]	1	-	1	-	2,67	1,67	3	1	-	1	2,67	2
FBMESC [33]	3,33	-	3,33	-	0,01	26,6	23,3	1	-	1	-	2
Önerilen Yöntem	1	-	1	-	3,19	1,72	3,06	1	0,72	1	3,12	2

Hesaplama zamanını değerlendirmek amacıyla ise yüm yöntemler C++ dili kullanılarak aynı optimizasyon seviyesinde gerçekleştirilmiştir. Şekil 4.5'te düşey eksen tüm diziler için elde edilmiş olan ortalama PSNR değerini, yatay eksen ise 16×16 boyutundaki bir blok için gereken dönüşüm ve uyumlama işlemlerinin toplam süresini göstermektedir. Bu grafikte en iyi PSNR değerini ve en az hesapsal yükü sağlayan

yöntemin sol üst köşede olması beklenmektedir. Bu kısımda verilmiş olan deneysel sonuçlar Intel i7 2.2 GHz işlemci, 16 GB RAM özelliklerine sahip bir sistem ile elde edilmiştir.



Şekil 4.5. Hareket kestirimi performansı/zaman grafiği

Bu grafikten görüldüğü üzere 1BT temelli hareket kestirimi yaklaşımı en düşük hesaplama zamanına sahip olmasına rağmen hareket kestirimi başarımı en düşük olan yöntemdir. T-GCBPM yaklaşımı ise en iyi hareket kestirimi performansın rağmen hesapsal yükü en yüksek olan yöntemdir. Önerilen yöntemin başarımı ise T-GCBPM'ye yakın olmakla birlikte, hareket kestirimi başarımı ve hesapsal yük arasında iyi bir dengeye sahiptir.

4.1.4. Sonuçlar

Tez çalışmasının bu aşamasında ekran içerik görüntüleri için düşük-bit gösterimi tabanlı bir hareket kestirim yöntemi geliştirilmiştir. En değerlikli bit düzlemi kullanılarak bir köşe haritası oluşturulmuş, bu köşe haritasına göre hangi bit düzlemlerinin uyumlama aşamasında kullanılacağına karar verilmiştir. Bununla birlikte uyumlama aşamasında bit düzlemlerinin ağırlandırılarak kullanılmasının da daha iyi PSNR sonuçları elde edilmesine katkı sağladığı görülmüştür. Geliştirilmiş olan yöntem, düşük hesapsal yüküyle hareket kestirim performansı arasında iyi bir denge ile genel olarak yüksek bir hareket kestirim başarımı göstermektedir. Fakat sonraki bölümlerde de anlatılacağı üzere NNMP hesabında sadece XOR işlemi yerine birden fazla işlem gerçekleştirilerek uyumlama hesabının karmaşıklaştırılması,

HEVC’de hesapsal yükün artmasına neden olmaktadır. Bu sebepten dolayı geliştirilmiş olan yöntemin donanımsal gerçeklemelere daha uygun olduğu değerlendirilip, HEVC referans yazılımına uygulanmamıştır.

4.2. HEVC Üzerinde Düşük Bit Derinliği Temelli Hareket Kestirim Yöntemlerinin Başarımlarının Değerlendirilmesi

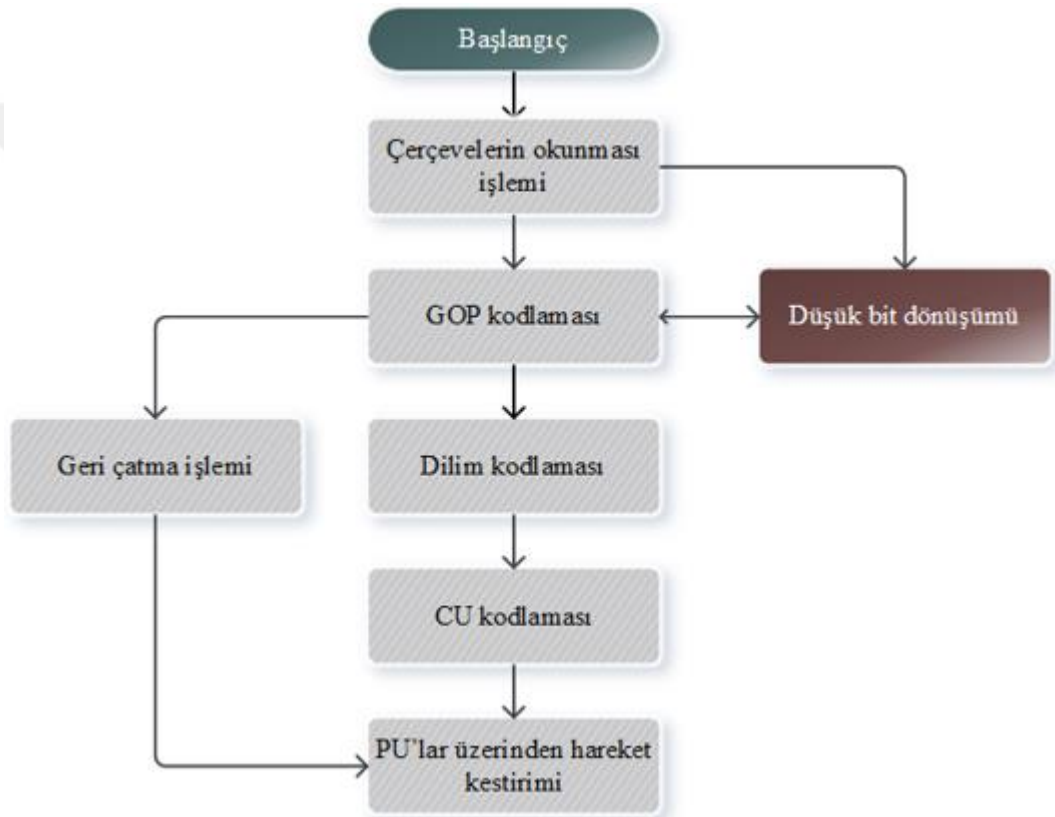
4.2.1. Giriş

HEVC’deki hareket kestirimi işlemleri Bölüm 2.2’de, düşük bit derinliği temelli hareket kestirimi yaklaşımları ise Bölüm 3.1.3’de detaylı olarak incelenmiştir. Tez çalışmasının bu aşamasında ise, düşük bit derinliği gösterimi temelli hareket kestirimi yaklaşımlarının analizi en güncel video kodlama standardı olan HEVC üzerinde literatürde ilk kez gerçekleştirilmiştir. Böylelikle geçmişte çoğunlukla açık çevrim performans değerlendirmesine tabii tutulan bu yaklaşımların bir kodlayıcı içerisindeki gerçek performansı ortaya çıkarılmıştır. Bunun yanı sıra, düşük bit-derinliği gösterimi temelli hareket kestirim yaklaşımlarının daha da hızlandırılması için seyrek arama yaklaşımları ile tümleştirilerek HEVC üzerine uygulanması literatürde ilk kez incelenmiştir.

4.2.2. Düşük bit derinliği temelli hareket kestirim yöntemlerinin HEVC’ye uygulanması

Bu tez kapsamında düşük bit gösterim temelli yöntemlerin HEVC üzerindeki başarımını değerlendirmek için HEVC referans yazılımı olan HM16.7 kullanılmıştır. Referans yazılımda hâlihazırda 8-bit derinliğinde pikseller üzerinden çalışan tam arama yaklaşımı ve seyrek arama yaklaşımı olan TZS gerçeklemeleri mevcuttur. Düşük bit temelli HK yöntemlerinin başarımını değerlendirmek için referans yazılımda bazı değişiklikler yapılmıştır. Öncelikle 1BT, C-1BT ve SGC dönüşümü temelli tam arama yaklaşımly yöntemler referans yazılıma ayrı ayrı entegre edilmiştir. Bunlarla birlikte TZS, üç-adımlı (TSS), altıgensel (HS), elmas biçimli (DS) seyrek arama yöntemleri de 1BT temelli görüntü çerçeveleri üzerinden referans yazılımda gerçekleştirilmiştir. Şekil 2.2’de verilen referans yazılımın akış şeması üzerinden yazılımsal değişiklikler yapılarak Şekil 4.6’da verildiği gibi yeni bir akış şeması elde edilmiştir. Bu akış şemasının referans yazılımın akış şemasına göre temel farkı, şekilde

farklı renk ile gösterilen dönüşüm işleminin eklenmesidir. Bu işlem, orijinal çerçevelerin okunması ve geri çatılmış olan çerçevelerin oluşturulması sırasında yapılmaktadır. Referans yazılımda bulunan ve 8 bit düzleminde gerçekleştirilen çerçeve grubu (Group of Pictures- GOP), çerçeve, dilim, CTU ve CU gibi yapılar seçilen yöntemle göre (1BT, C-1BT, SG) düşük bit düzleminde de oluşturulmuştur. HK işlemleri için oluşturulan bu yeni yapılar kullanılmıştır. Ayrıca C-1BT yöntemi için kısıt maskesi, SG yöntemi için hangi bit düzlemlerinin seçileceğini gösteren maske yapısı da düşük bit derinliğinde benzer şekilde oluşturulmuştur.



Şekil 4.6. HEVC'deki düşük bit gösterim temelli hareket kestiriminin akış şeması

HEVC'de her bir PU bölümü için HK'daki maliyet hesabı Bölüm 2.2.3'de anlatılmış olan işlemlere göre gerçekleştirilmektedir.

Düşük bit gösterimi temelli HK işlemlerinde ise PU'lar için blok uyumlama ölçütü olarak SAD yerine EX-OR işlemi kullanan NNMP ölçütünden yararlanılmıştır. NNMP ölçütü ile hesaplanan bozunum değerleri SAD ölçütü ile hesaplanan bozunum değerlerine göre düşük olmaktadır. Bu durumda maliyet hesabının NNMP kriterine göre ölçeklendirilmesi gerekmektedir. Bu çalışmada NNMP ölçütü ile hesaplanan

bozunum deęerleri, deneysel alıřmalardan elde edilen uygun bir katsayıyla (128) leklendirilmiřtir. Bununla birlikte HM yazılımında hareket kestirimi ařamasında uyumlama yapılırken gerekleřtirilen rnek seyreltme iřlemleri kullanılmamıřtır.

4.2.3. Deneysel sonular

Bu alıřmada dřük bit gsterim temelli yntemlerin bařarımlarının deęerlendirmek iin iki ařamalı bir deneysel yaklařım kullanılmıřtır. İlk ařamada, dřük bit temelli yntemlerin HEVC zerindeki HK bařarımı tam arama yaklařımı kullanılarak incelenmiř, sonraki ařamada ise seyrek aramaya dayalı dřük bit temelli yntemlerin HEVC zerindeki bařarımı irdelenmiřtir.

Dřük bit gsterim temelli yntemlerin bařarım analizinde, Tablo 2.1’de verilen farklı znrlk deęerlerine ve ereve sayısına sahip grnt dizileri kullanılmıřtır. HM 16.7’de tm yntemler iin kullanılan nemli yapılandırma parametreleri Tablo 4.3’de verilmiřtir. Yntemlerin nicel bařarımlarını deęerlendirmek iin 22, 27, 32, 37 olmak zere farklı QP deęeri kullanılmıřtır. Bu kısımda verilen deneysel sonular 2.90 GHz Xeon iřlemci, 48 GB RAM zelliklerine sahip bir sistem ile elde edilmiřtir.

Tablo 4.3. Temel yapılandırma parametreleri

Yapılandırma Tipi	low_delay_main
ereve Tipi	B
Arama Yntemi	Tam Arama / Seyrek Arama
Arama Aralıęı	64
Bit Oran Kontrol	Kapalı
QP	22, 27, 32, 37

4.2.3.1. Dřük bit gsterim temelli yaklařımların HEVC zerindeki tam arama hareket kestirim bařarımı

Bu alt blmde, dřük bit temelli 1BT, C-1BT ve SGC yntemlerinin HEVC zerindeki bařarımları deęerlendirilmiř ve hareket vektrleri tam arama yaklařımıyla elde edilmiřtir.

1BT, C-1BT ve SGC’nin “Kimono”, “ParkScene” ve “PeopleOnStreet” dizileri iin video kodlama bařarımları Tablo 4.4, Tablo 4.5 ve Tablo 4.6’da sırasıyla verilmiřtir. Bařarım ltleri PSNR aısından deęerlendirildięinde, dřük bit gsterim temelli yntemler birbirine yakın performans sergilemekte olup SGC yntemi dięer yntemlere (1BT ve C-1BT) gre nispeten daha iyi bir bařarım vermektedir. Ayrıca

bu yöntemlerin tam arama yaklaşımına göre performans kaybı da kabul edilebilir düzeydedir. Başarım ölçütleri bit-oran açısından değerlendirildiğinde düşük bit gösterim temelli yöntemler tam arama yaklaşımına göre daha fazla miktarda veri kodlama gereksinimi duymaktadır. Diğer yandan, SGC yöntemi diğer yöntemlere (1BT ve C-1BT) göre daha fazla bit kazancı sağlamaktadır.

Tablo 4.4. Farklı QP’lerde düşük bit temelli hareket kestirim yöntemlerinin Kimono dizisi için ortalama PSNR (dB) ve bit-oran değerleri

QP	1BT		C-1BT		SGC		Tam Arama (SAD)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	41,63	5328,36	41,64	5318,40	41,67	5242,73	41,68	5201,74
27	39,61	2436,49	39,61	2436,80	39,65	2415,99	39,66	2400,80
32	37,09	1163,24	37,09	1164,06	37,11	1156,82	37,13	1151,06
37	34,49	568,53	34,50	569,00	34,52	564,89	34,54	562,79

Tablo 4.5. Farklı QP’lerde düşük bit temelli hareket kestirim yöntemlerinin ParkScene dizisi için ortalama PSNR (dB) ve bit-oran değerleri

QP	1BT		C-1BT		SGC		Tam Arama (SAD)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	39,84	8269,98	39,86	8210,96	39,89	8046,6	39,92	7949,17
27	36,95	3247,61	36,96	3232,56	36,97	3192,8	37,00	3163,33
32	34,15	1354,60	34,16	1351,86	34,17	1341,0	34,19	1332,39
37	31,55	582,05	31,56	580,55	31,57	577,2	31,59	574,59

Tablo 4.6. Farklı QP’lerde düşük bit temelli hareket kestirim yöntemlerinin PeopleOnStreet dizisi için ortalama PSNR (dB) ve bit-oran değerleri

QP	1 BT		C1BT		SGC		Tam Arama (SAD)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	40,79	38813,9	40,80	38431,4	40,85	37722,9	40,86	37097,8
27	37,41	17601,9	37,42	17422,4	37,47	17251,7	37,49	16937,8
32	34,27	8968,5	34,29	8905,5	34,34	8880,9	34,38	8770,1
37	31,43	5038,8	31,47	5021,1	31,51	5012,0	31,56	4974,8

Şekil 4.7’de sırasıyla “Kimono”, “ParkScene” ve “PeopleOnStreet” dizileri için 1BT, C-1BT ve SG yöntemlerinin HEVC üzerindeki oran-bozunum grafikleri gösterilmiştir. Her bir dizi için, tam arama yaklaşımı dâhil bütün yöntemler tüm bit değerlerinde birbirlerine yakın görsel kalite gösterdikleri anlaşılmaktadır. Oran-bozunum grafiklerine yakından bakıldığında, 1BT, C-1BT ve SG yöntemlerinin tam arama

(SAD) yaklaşımına göre aynı bit değerlerinde daha düşük görsel kaliteye sahip oldukları görülmektedir.

Diğer yandan, SGC yöntemi, 1BT ve C-1BT'ye göre aynı bit değerlerinde daha yüksek bir görsel kaliteye sahiptir. Benzer şekilde, 1BT ve C-1BT yöntemleri aynı görsel kalite altında, SGC yöntemine göre daha fazla bit harcamaktadır.

Tablo 4.7'de ise SGC, 1BT ve C-1BT yöntemlerinin, [10]'da önerilmiş metrik kullanılarak HM'deki tam arama yaklaşımı ile karşılaştırılması sonucu elde edilmiş olan ΔB değerleri verilmiştir.

Tablo 4.7. SGC, 1BT ve C-1BT yöntemlerinin HM (tam arama) ile karşılaştırılması (LDB)

Yöntem	Kimono	ParkScene	PeopleOnStreet
	ΔB	ΔB	ΔB
SGC	1.12	1.76	1.96
1BT	3.23	5.04	4.77
C-1BT	2.99	4.21	3.75

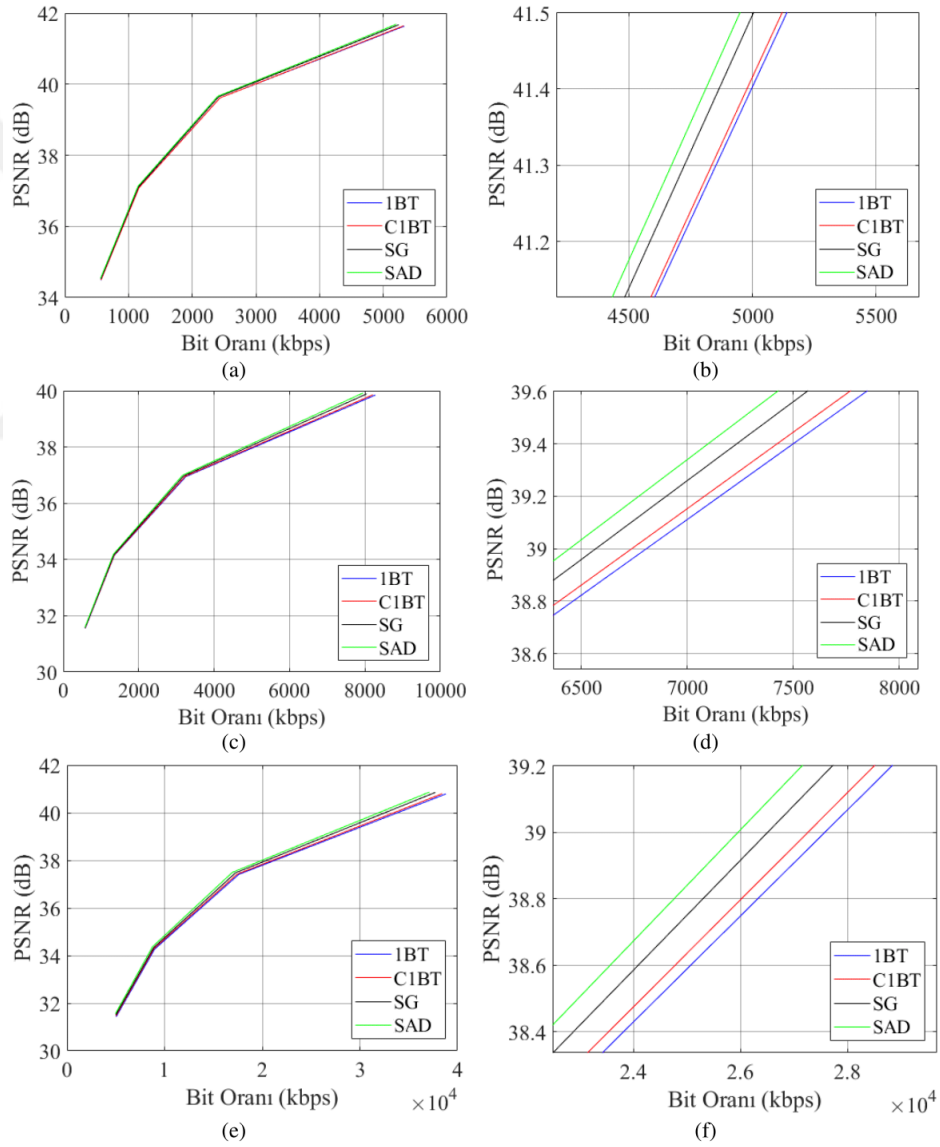
Tam arama temelli 1BT, C-1BT ve SGC yöntemlerinin Q=27 için çalışma süreleri saat bazında Tablo 4.8'de verilmiştir. Tablo 4.8'e göre tüm diziler için, C-1BT yönteminin işlemsel yükü diğer yöntemlere göre daha fazladır. Bunun temel nedeni, C-1BT yönteminde kısıtlama maskesinin kullanılması ve buna bağlı olarak kullanılan XOR tabanlı uyumlama ölçütünün daha fazla karşılaştırma işlemi yapmasıdır. Bununla birlikte, 1BT temelli hareket kestiriminin işlemsel yükü standart SAD yöntemi ile karşılaştırıldığında nispeten değişmemektedir. Bunun temel nedenlerinden biri ise her iki durum içinde işlemcinin ALU'sunda her seferde tek piksele ilişkin uyumlama hesabının yapılmasıdır. SGC yöntemi ise düşük bit derinliğine geçiş işlemi daha etkin yaptığı için diğer yöntemlere göre daha hızlı çalışmaktadır.

Tablo 4.8. Tam arama temelli 1BT, C1BT ve SGC yöntemlerinin HEVC üzerindeki çalışma zamanları (saat)

	1BT	C-1BT	SG	Tam Arama (SAD)
Kimono	172,72	227,13	167,36	171,54
ParkScene	170,47	234,90	171,22	171,21
PeopleOnStreet	153,29	201,15	148,39	152,92

4.2.3.2.1BT temelli hareket kestirim yaklaşımlarının HEVC üzerindeki seyrek arama hareket kestirim başarımı

Bu alt bölümde, düşük bit gösterim temelli HK yöntemlerinden 1BT'nin farklı seyrek arama yaklaşımları ile birlikte HEVC üzerindeki başarımı değerlendirilmiştir. Tam arama yaklaşımında düşük bit gösterim temelli yöntemlerin HEVC üzerindeki başarımları birbirine yakın olduğu için seyrek arama temelli yöntemlerin başarımlarının değerlendirilmesinde düşük bit düzlemine geçmek için en temel yöntemlerden olan 1BT temelli yaklaşım tercih edilmiştir.



Şekil 4.7. Düşük bit temelli hareket kestirim yöntemlerinin oran-bozunum performansı a) Kimono dizisi b) Kimono dizisi yakından görünüm c) ParkScene dizisi d) ParkScene dizisi yakından görünüm e) PeopleOnStreet dizisi f) PeopleOnStreet dizisi yakından görünüm

Tablo 4.9, Tablo 4.10 ve Tablo 4.11’de farklı görüntü dizileri için 22, 27, 32, 37 QP parametreleri altında TZS (8 bit) ve TZS (1BT), HS (1BT), TSS (1BT) ve DS (1BT) yöntemlerinin ortalama PSNR (dB) değerleri verilmiştir. TZS (8 bit), TZS seyrek arama algoritmasının standart 8 bit/piksel çözünürlüğüne sahip görüntü çerçeveleri üzerinden gerçekleştirilmesine karşılık gelmektedir. Diğer seyrek arama algoritmaları ise düşük bit gösterim temelli 1BT yöntemiyle gerçekleştirilmesine karşılık gelmektedir. Tablolardan görüleceği gibi, TZS (1BT) yöntemi, TZS (8 bit) yöntemine göre gerek PSNR gerekse harcanan bit miktarı açısından daha düşük performans sergilemektedir. Performans kaybının miktarı kullanılan görüntü dizisine bağlı olarak değişmektedir. Görüntü dizisinin içeriğine göre hareket vektörleri büyük genlikli olabilmektedir. TZS algoritması, kendisine en çok benzeyen hedef bloğu önce yakın bölgede, gerekirse uzak bölgede aramaktadır. Bu gibi durumlarda 1BT ile TZS’nin birlikte kullanılması PSNR açısından başarıyı düşürmektedir. Bunun temel nedeni, TZS’nin 1BT ile birlikte kullanıldığı durumda yakın bölgede hareket vektörü bulamaması durumunda taramalı arama (raster scan) moduna geçmesi nedeniyle yerel bir minimuma takılması olarak değerlendirilmektedir. Diğer yandan, 1BT temelli seyrek arama yöntemlerinin ortalama PSNR (dB) değerleri genel olarak birbirine yakın olmaktadır.

Tablo 4.9. Farklı QP’lerde bir bit temelli hareket kestirim yöntemlerinin Kimono dizisi için ortalama PSNR (dB) ve bit-oran değerleri

QP	DS (1BT)		TSS (1BT)		HS (1BT)		TZS (1BT)		TZS (8-bit)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	41,63	5372,59	41,63	5398,21	41,63	5398,41	41,60	5845,67	41,68	5201,64
27	39,61	2472,93	39,60	2486,69	39,60	2489,47	39,51	2726,72	39,66	2403,08
32	37,06	1189,62	37,04	1200,27	37,04	1199,97	36,90	1310,86	37,13	1151,58
37	34,43	583,07	34,41	589,42	34,41	589,93	34,26	635,76	34,53	563,18

Tablo 4.10. Farklı QP’lerde bir bit temelli hareket kestirim yöntemlerinin ParkScene dizisi için ortalama PSNR (dB) ve bit-oran değerleri

QP	DS (1BT)		TSS (1BT)		HS (1BT)		TZS (1BT)		TZS (8-bit)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	39,85	8270,49	39,85	8272,03	39,85	8274,01	39,85	8233,49	39,91	7961,37
27	36,94	3266,42	36,94	3269,17	36,94	3269,22	36,95	3252,52	36,99	3166,94
32	34,13	1373,40	34,13	1373,53	34,13	1375,51	34,14	1367,35	34,19	1334,90
37	31,52	595,87	31,52	595,32	31,52	595,55	31,52	593,49	31,58	575,86

Tablo 4.11. Farklı QP’lerde bir bit temelli hareket kestirim yöntemlerinin PeopleOnStreet dizisi için ortalama PSNR (dB) ve bit-oran değerleri

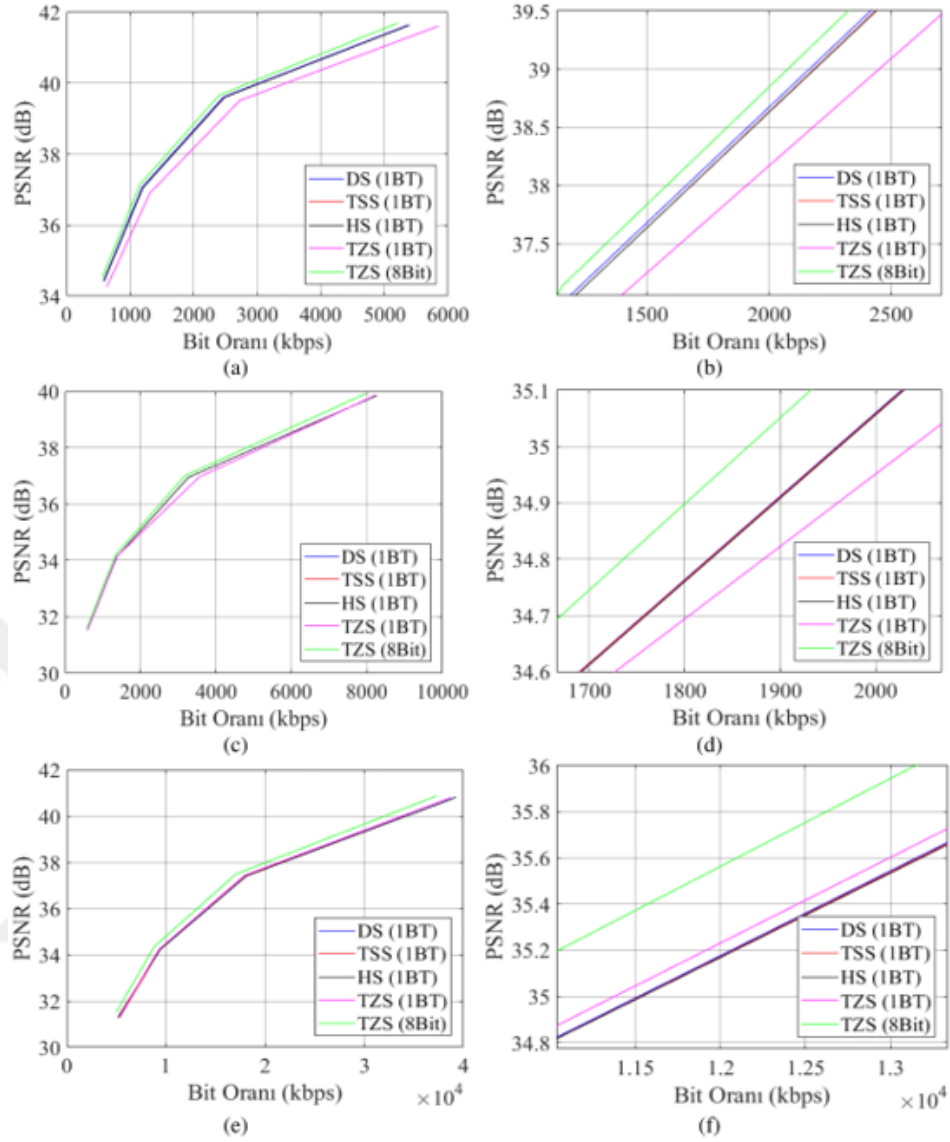
QP	DS (1BT)		TSS (1BT)		HS (1BT)		TZS (1BT)		TZS (8-bit)	
	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı	PSNR	Bit Oranı
22	40,80	39175,36	40,80	39240,21	40,80	39239,05	40,80	38739,7	40,86	37242,9
27	37,42	18101,95	37,42	18121,94	37,41	18130,49	37,42	17883,6	37,48	17036,4
32	34,21	9372,10	34,21	9383,09	34,21	9379,53	34,22	9289,4	34,35	8818,8
37	31,28	5297,06	31,28	5290,13	31,28	5133,18	31,30	5262,3	31,52	4989,1

Şekil 4.8’de sırasıyla “Kimono”, “ParkScene” ve “PeopleOnStreet” dizileri için DS (1BT), TSS (1BT), HS (1BT) TZS (1BT) ve TZS (8 bit) yöntemlerinin HEVC üzerindeki oran-bozunum başarımları gösterilmiştir. Oran-bozunum grafiklerine yakından bakıldığında, 1BT temelli seyrek arama yöntemlerinde, TZS (8 bit)’e göre belirgin bir performans kaybı gözlemlenmektedir. Performans kaybı, görüntü dizisinin içeriğine bağlı olarak yaklaşık 0.2 dB’ye kadar çıkabilmektedir. Diğer yandan, 1BT temelli seyrek arama yöntemleri genel olarak aynı bit değerlerinde birbirlerine yakın performans göstermektedir. Bu bilgiler ışığında düşük bit derinliği temelli yöntemlerin TZS ile uyumlu çalışmadığı ve bu yöntemlere özgü daha etkin seyrek arama yaklaşımları araştırılabileceği değerlendirilmektedir.

1BT temelli seyrek arama yöntemleri, 8 bit temelli TZS yöntemi ile [10]’da önerilmiş metrik ve Eşitlik (1.4) kullanılarak karşılaştırılmış, elde edilmiş olan ΔB ve ΔT değerleri ise Tablo 4.12’de verilmiştir.

Tablo 4.12. SGC, 1BT ve C-1BT yöntemlerinin HM (TZS) ile karşılaştırılması (LDB)

Yöntem	Kimono	ParkScene	PeopleOnStreet
	ΔB	ΔB	ΔB
DS(1BT)	6.02	5.34	10.4
HS(1BT)	7.55	5.38	8.29
TSS(1BT)	7.50	5.33	10.4
TZS(1BT)	21.22	6.01	9.15



Şekil 4.8. 1BT temelli seyrek arama yöntemlerinin oran-bozunum performansı a) Kimono dizisi b) Kimono dizisi yakından görünüm c) ParkScene dizisi d) ParkScene dizisi yakından görünüm e) PeopleOnStreet dizisi f) PeopleOnStreet dizisi yakından görünüm

1BT temelli seyrek arama yaklaşımlarının QP=27 için HEVC üzerindeki çalışma süreleri saat bazında Tablo 4.13'de verilmiştir. Bu aşamada deneysel sonuçlar 2.30 GHz Quad core işlemci ve 16 GB RAM özelliklerine sahip bir sistem ile elde edilmiştir. Tablodan görüleceği gibi, TZS (1BT) ve TZS (8-bit) seyrek arama yöntemlerinin çalışma süreleri birbirine yakın olmakla birlikte, diğer seyrek arama yöntemlerine göre daha düşük değerlerdedir.

Tablo 4.13. 1BT temelli seyrek arama yöntemlerinin HEVC üzerindeki çalışma süreleri (saat) (QP = 27)

	DS (1BT)	TSS (1BT)	HS (1BT)	TZS (1BT)	TZS (8-bit)
Kimono	7,71	5,47	5,89	5,20	5,35
ParkScene	8,16	7,76	7,32	5,25	5,06
PeopleOnStreet	7,18	6,35	6,63	4,96	5,04

4.2.4. Sonuçlar

Tez çalışmasının bu kısmında düşük bit-derinliği gösterimi temelli hareket kestirim yaklaşımlarının en güncel video kodlama standardı olan HEVC üzerine etkin şekilde uygulanması üzerine çalışılmıştır. Yapılan gerçekleştirme ve deneyler düşük bit-derinliği gösterimi temelli yaklaşımların hareket kestirim performansından bir miktar ödün vermeye neden olduğunu göstermiştir. Bununla birlikte 1920×1080 boyutunda bir çerçeve için gerçekleştirilen ikili dönüşüm işlemlerinin 1 sn'den daha az zaman aldığı ve bunun toplam kodlama süresi dikkate alındığında gözardı edilebilir olduğu görülmüştür. SGC yaklaşımının ise diğer düşük bit temelli yöntemlere göre daha iyi sonuçlar verdiği görülmüştür. Diğer yandan, düşük bit-derinliğine sahip bazı hareket kestirim yaklaşımlarının kodlama süresi açısından tam arama temelli yöntemden hızlı, bazılarının ise yavaş kalabildiği ortaya çıkarılmıştır. Bu farklılığın temel nedeni ise uyumlama ölçütünün hesapsal karmaşıklığıdır. Donanımsal olarak etkin şekilde gerçekleştirilebilen uyumlama ölçütlerinin yazılımsal olarak etkin çalıştırılmaması söz konusu olabilmektedir. Bu aşamada düşük bit-derinliği temelli yöntemlerin yazılımsal gerçeklemede tek komut çoklu data (single instruction multiple data – SIMD) yapısının kullanılabileceği değerlendirilmektedir. Öte yandan, düşük bit derinliği gösterimi temelli hareket kestirim yaklaşımlarının seyrek arama temelli yaklaşımlar ile tümleştirilmesi durumunda da TZS gibi yaklaşımların etkin sonuç vermediği tespit edilmiştir. Bu noktada düşük bit derinliği gösterimi temelli yaklaşımlara uygun seyrek arama yöntemlerinin geliştirilebileceği değerlendirilmiştir.

4.3. HEVC için Seçmeli Gray Kodlama Temelli Hızlı Hareket Kestirimi

4.3.1. Giriş

Tez çalışmasının bu bölümünde, Bölüm 4.2’de elde edilen sonuçlar doğrultusunda, hareket kestirimi için düşük bit derinliği temelli yöntemler arasında en iyi sonuçları veren SGC yaklaşımını kullanan hızlı bir hareket kestirim yöntemi önerilmiştir.

Geliştirilmiş olan yöntem iki aşamadan oluşmaktadır. İlk aşamada, HEVC referans yazılımı üzerine Bölüm 4.2’de uygulanmış SGC yöntemi için hızlı bir tam arama yaklaşımı, ikinci aşamada ise seyrek bir arama yaklaşımı geliştirilmiştir. Geliştirilmiş olan bu yaklaşımlarda hareket kestirimi işlemlerindeki zamansal karmaşıklığı azaltmak amacıyla veriler uygun bir hâle getirilerek blok uyumlama işlemleri için SIMD (Single Instruction Multiple Data – Tekli Komut Çoklu Veri) saklayıcıları ve komutları kullanılmıştır. Bununla birlikte, hesapsal karmaşıklığı azaltmak amacıyla blokların yarısına ait NNMP değeri hesaplandıktan sonra hesaplanan değer 2 ile çarpılarak elde edilen değer minimum NNMP değerinden küçükse ilgili hareket vektörü için NNMP hesaplama işlemi sonlandırılmıştır.

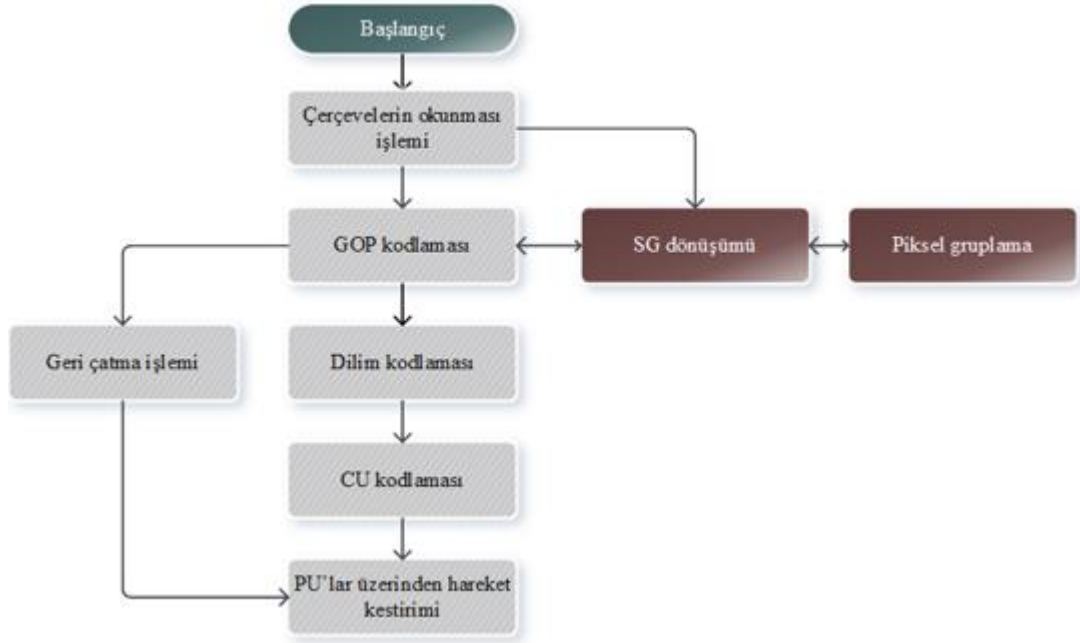
Günümüzde bir çok işlemci SIMD motorları ve saklayıcıları içermektedir. SIMD sayesinde büyük verilerle tekrarlı bir şekilde aynı işlemlerin gerçekleştirileceği durumlarda veriler paralel bir şekilde işlenebilmektedir. Örneğin standart bir işlemcide 8 bitlik 32 adet değişkene bir değer eklemek için toplamak için ilgili komutun 32 kez çalıştırılması gerekmektedir. Fakat 256 bitlik SIMD saklayıcıları kullanıldığı zaman bu işlem tek bir komut ile gerçekleştirilebilmektedir. Bu çalışmada 256 bitlik SIMD saklayıcıları, NNMP uyumlama kriterinin hesabında gerçekleştirilen XOR işlemleri için kullanılmıştır.

4.3.2. Blok hareket kestirimi için verilerin uyumlanması

SIMD saklayıcıları sayesinde tek komut çalıştırılarak işlemler gerçekleştirilebiliyor olsa da SIMD saklayıcılarına veri yüklemek zaman açısından maliyetli bir işlemdir. Yapılan deneysel çalışmalar SIMD saklayıcılarına bellekte sıralı bir şekilde bulunan verilerin yüklenmesinin bellekte dağınık bir şekilde bulunan verilerin yüklenmesine göre daha hızlı gerçekleştiğini göstermektedir. Bu nedenle SIMD saklayıcılarını ve

işlemlerini verimli bir şekilde kullanabilmek için veri yapısının bu işlemlere uygun bir hâle getirilmelidir.

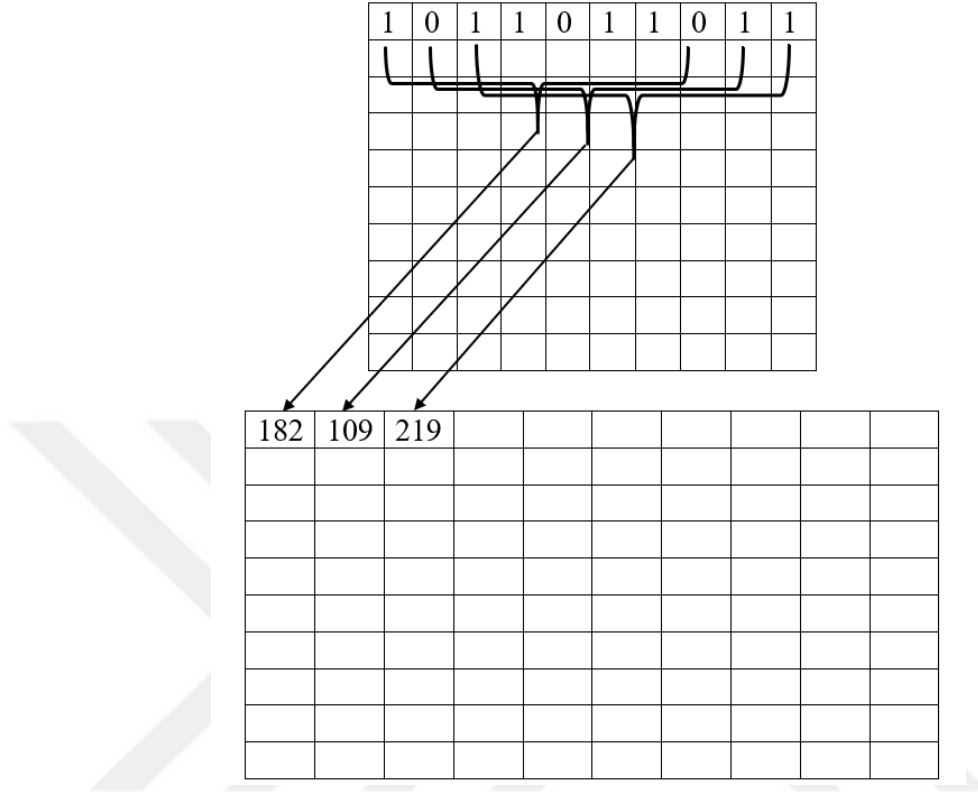
Bununla birlikte SG dönüşümü gerçekleştirilmiş piksel değerleri 1 ve 0 değerlerini içeriyor olsalar da bilgisayar ortamında pratikte bu değerler byte türü değişkenler içinde tutulmaktadır. Eğer veriler bu şekilde byte türü değişkenler ile 256 bitlik saklayıcılara yüklenirse tek seferde 32 piksel için XOR işlemi gerçekleştirilebilecektir. Fakat 8 adet SG dönüşümü gerçekleştirilmiş piksel gruplanarak 1 byte ile temsil edilirse 256 piksel için XOR işlemi tek komut ile gerçekleştirilebilir. Bu doğrultuda geliştirilen yöntemin ilk aşamasında bit kaydırma operatörleri kullanılarak pikseller 8'erli olarak gruplanmış ve 1 byte'lık bir alan içine yerleştirilmiştir. Referans yazılımın HEVC'deki düşük bit gösterim temelli hareket kestiriminin akış şeması (Şekil 4.6) üzerinde piksel gruplama işlemi yapılarak Şekil 4.9'daki yeni akış şeması elde edilmiştir.



Şekil 4.9. HEVC referans yazılımı üzerinde piksel gruplama işlemi

Arama yaklaşımlarında seçilmiş olan yöntem doğrultusunda referans penceredeki aday hareket vektörlerinde uyumlama kriteri hesabı gerçekleştirilmekte ve uygun hareket vektörüne karar verilmektedir. Piksellerin gruplanması aşamasında, referans penceredeki arama işlemlerinin yapısını bozmamak amacıyla her byte kendi konumunu takip eden 8 adet SG dönüşümü gerçekleştirilmiş pikseli tutacak şekilde

ayarlanmıştır. Bu sayede referans penceredeki her aday vektöre erişim kolaylaşmıştır. Gerçekleştirilmiş olan bu işlem ile ilgili bir örnek Şekil 4.10'da verilmiştir.



Şekil 4.10. Örnek bir piksel gruplama işlemi

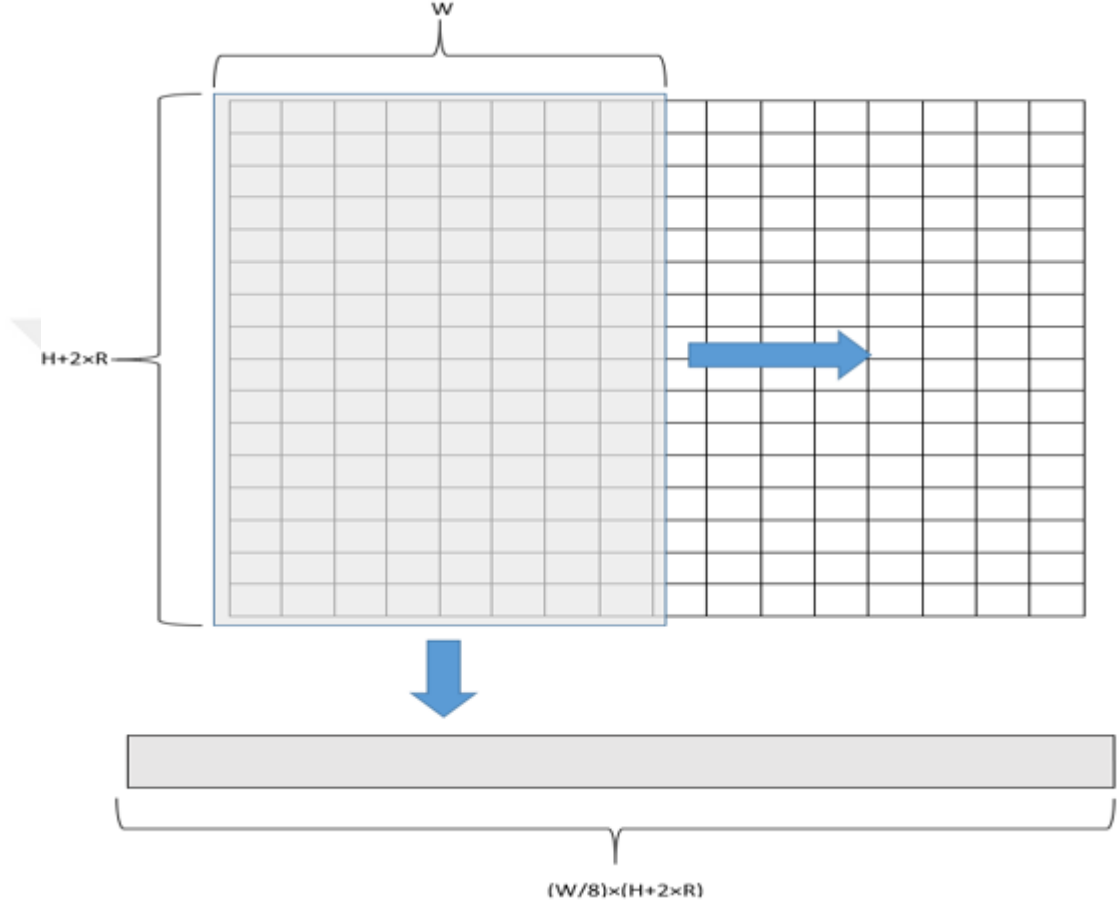
Bu dönüşümden sonra bir satırdaki pikseldeki elemanlara erişmek için 8 piksel atlanması gerekmektedir. Örneğin 16×16 boyutunda bir PU ele alındığında, hareket kestirimi için ilgili satıra ait 0 ve 8. indislerde bulunan byte değerlerinin alınması yeterli olmaktadır. Bu işlemler sadece çerçevelerin okunmasında ve geri çatılmış çerçevelerin dönüşümü sırasında gerçekleştirildiği için HEVC'nin toplam hesapsal karmaşıklığına etkisi çok az olarak değerlendirilmektedir.

4.3.3. Hızlı tam arama yaklaşımı

Tam arama yaklaşımında referans penceredeki tüm aday hareket vektörleri için uyumlama kriteri hesaplanarak hareket vektörüne karar verilmektedir.

SIMD saklayıcılarına verileri sıralı bir şekilde yükleyebilmek amacıyla bu aşamada bir dönüşüm işlemi daha gerçekleştirilmiştir. Bu işlemin ilk adımında $h \times w$ büyüklüğündeki blok, bir önceki bölümde gerçekleştirilmiş olan byte dönüşü

sayesinde $H \times W/8$ büyüklüğünde bir byte dizisine yazılmıştır. Bu işlemin bir kere gerçekleştirilmesi yeterlidir. Arama penceresindeki her sütun başlangıcından itibaren alınan $W \times (H+2 \times R)$ büyüklüğündeki bloklar ise $(W/8) \times H+2 \times R$ büyüklüğündeki byte dizilerine yazılmıştır. Şekil 4.11’de bu işlem verilmiştir.



Şekil 4.11. Referans çerçevedeki verilerin dönüştürülmesi

Her sütunda referans çerçeve için dönüşüm işlemi gerçekleştirildikten sonra, orijinal çerçeveye ait veriler de kullanılarak NNMP hesabı yapılmaktadır. Bu işlem tek boyutlu bir dizinin içinde arama işlemi olarak da değerlendirilebilir. Blok genişliği $W/8$ olduğu için arama işlemi sırasında adım aralığı $W/8$ olmaktadır. Arama işlemi sırasında da veriler sıralı bir şekilde 256 bitlik SIMD saklayıcılarına yüklenmiş ve bu sayede 256 piksel için XOR işlemi tek seferde gerçekleştirilmiştir. Bu işlemler HEVC’de tüm derinliklerde bulunan PU modlarının boyutlarına göre optimize edilmiştir. XOR işlemleri sonucunda elde edilen SIMD saklayıcısındaki 1 değerleri sayılarak toplanmış ve böylelikle NNMP değeri elde edilmiştir.

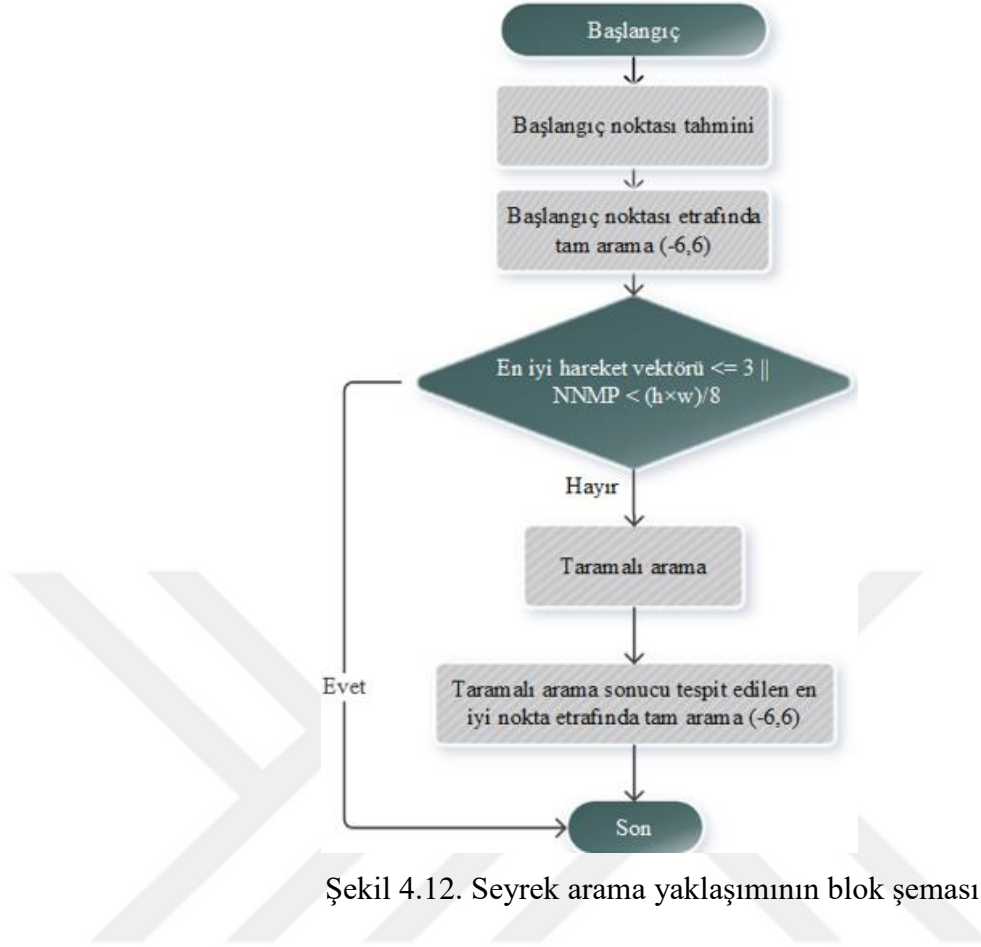
Blok boyutundaki veri miktarı, PU boyutundan dolayı 256 bitten daha küçük olduğu zaman ise veriler ardışık bir şekilde SIMD saklayıcılarına yüklenerek birden fazla hareket vektörü için NNMP değeri hesaplanması sağlanmıştır.

Her sütun için yukarıda verilmiş olan işlemler gerçekleştirilerek tüm aday hareket vektörleri için NNMP değerleri hesaplanmakta ve Bölüm 4.2.2’de verildiği şekilde maliyet hesabı gerçekleştirilerek uygun hareket vektörüne karar verilmektedir.

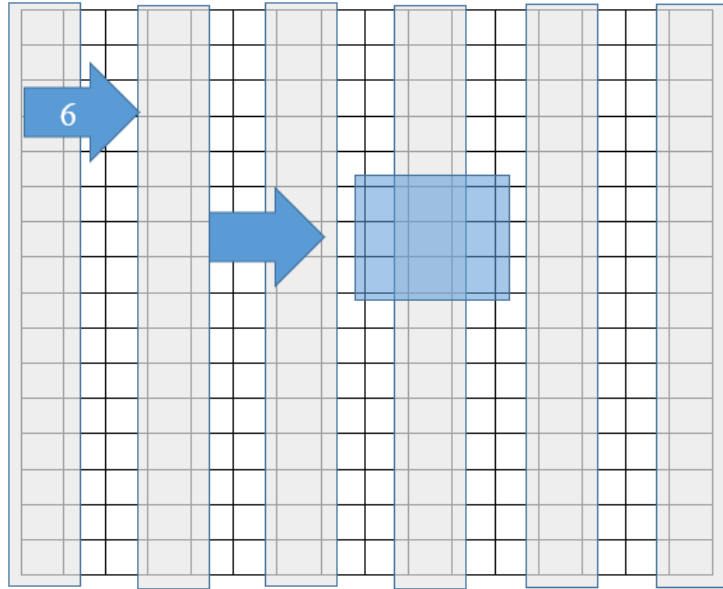
4.3.4. Seyrek Arama Yaklaşımı

Tez çalışmasının bu aşamasında HEVC için SGC temelli bir seyrek arama yaklaşımı geliştirilmiştir. Bu yöntemde de Bölüm 4.3.2’de anlatıldığı şekilde SIMD işlemi için uygun şekilde sıralanmış piksel değerleri kullanılmıştır. Bölüm 4.3.3’e benzer bir şekilde SIMD kullanımı için gerekli dönüşüm işlemleri gerçekleştirilerek NNMP hesabı yapılmış fakat bu işlemler referans arama penceresinin tamamı yerine referans arama penceresindeki belirli noktaların etrafından alınan daha küçük arama pencereleri üzerinde yapılmıştır. Şekil 4.12’de geliştirilmiş olan seyrek arama yaklaşımının blok şeması verilmiştir.

Bu yöntemde ilk olarak komşu CU’lar için elde edilmiş hareket vektörleri değerlendirilerek başlangıç noktasına karar verilmektedir. İkinci adımda başlangıç noktası etrafında (-6,6)’lık bir arama aralığı içerisinde Bölüm 4.3.3’e benzer bir şekilde tam arama yapılmaktadır. En iyi hareket vektörü 3’ten küçükse veya elde edilmiş en iyi NNMP değeri $(h \times w) / 8$ değerinden küçükse arama işlemi sonlandırılmaktadır. Aksi duruma ızgara şeklinde bir örüntü kullanılarak arama penceresindeki noktalarda NNMP hesabı gerçekleştirilmektedir. Taramalı aramadaki adım aralığı SG şablonun yapısı da değerlendirilerek tekrarlı işlemleri önlemek amacıyla 6 olarak belirlenmiştir. Bu işlemde her sütun için 6 adım ilerlenirken, ilgili sütundaki tüm aday vektörleri için NNMP hesabı gerçekleştirilmektedir. Şekil 4.13’de taramalı arama işlemi için örnek bir şablon ve örnek bir arama işlemi için son aşamada değerlendirilmiş olan blok verilmiştir. Son olarak da en iyi NNMP değerini veren noktanın etrafında (-6,6)’lık bir bölgede tam arama yapılarak hareket kestirimi işlemi sonlandırılmaktadır.



Şekil 4.12. Seyrek arama yaklaşımının blok şeması



Şekil 4.13. Taramalı arama işlemi için örnek bir şablon

4.3.5. Deneysel sonuçlar

Tezin bu bölümden geliştirilmiş olan yöntemler HEVC'nin referans yazılımı olan HM 16.7 üzerinde uygulanmıştır. Yöntemlerin başarımını değerlendirmek amacıyla

“Kimono”, “ParkScene”, “FourPeople”, “Johnny” ve “KristenAndSara” dizileri için sonuçlar LDB konfigürasyonunda elde edilmiştir. Tablo 2.1’de bu görüntü dizileri özellikleri verilmiştir. Yöntemin başarımını değerlendirmek amacıyla ise [10]’da önerilmiş olan metrik kullanılmıştır. Bu metrikle ilgili detaylar Bölüm 1.3’de verilmiştir. Önerilmiş olan yöntem 22, 27, 32 ve 37 kuantalama katsayıları için bu işlemler gerçekleştirilerek [10]’daki metrik sayesinde ΔB değeri elde edilmiş, ΔT değeri ise Eşitlik (1.4) kullanılarak hesaplanmıştır. Arama aralığı ise (-64,64) olarak belirlenmiştir. Bu kısımda verilmiş olan deneysel sonuçlar Intel i7 7800X işlemci, 16 GB RAM özelliklerine sahip bir sistem ile elde edilmiştir. SIMD ile gerçekleştirilen işlemler için ise AVX2 kütüphanesi ve 256 bitlik saklayıcılar kullanılmıştır.

Çalışmada geliştirilmiş olan hızlı tam arama yaklaşımının sonuçları, referans yazılımından elde edilmiş olan tam arama yaklaşımının sonuçlarıyla karşılaştırılmıştır. Elde edilen sonuçlar Tablo 4.14’te verilmiştir.

Tablo 4.14. Geliştirilmiş olan hızlı tam arama yaklaşımı ile HM’deki tam arama yaklaşımının sonuçlarının karşılaştırılması

Görüntü Dizisi	ΔB	ΔT
Kimono	1,89	68,5
ParkScene	2,42	68,8

Elde edilmiş olan sonuçlar geliştirilmiş olan hızlı tam arama yaklaşımının hesapsal karmaşıklığı %70’e kadar azaltabildiğini göstermiştir. SGC yaklaşımının beklendiği gibi daha fazla bit harcadığı görülmekle birlikte, ΔB değerinin kabul edilebilir bir düzeyde olduğu görülmüştür.

Tezin bu kısmında geliştirilmiş olan seyrek arama yaklaşımının HM’deki seyrek arama yaklaşımı olan TZS ile karşılaştırılmasının sonuçları ise Tablo 4.15’te verilmiştir.

Tablo 4.15. Geliştirilmiş olan seyrek arama yaklaşımı ile HM’deki TZS yaklaşımının sonuçlarının karşılaştırılması

Görüntü Dizisi	ΔB	ΔT
Kimono	2,18	15,4
ParkScene	2,34	12,5
FourPeople	1,83	12,8
Johnny	1,78	11,6

Seyrek arama yöntemi için elde edilmiş olan sonuçlar geliştirilmiş olan yöntemin HEVC referans yazılımındaki TZS algoritmasına göre %15'e kadar hızlanma sağlayabildiğini göstermektedir. Bu aşamada da SGC yaklaşımının beklendiği gibi daha fazla bit harcadığı görülmekle birlikte, ΔB değerinin kabul edilebilir bir düzeyde olduğu görülmüştür.

Seyrek arama yönteminin performansını sağlıklı bir şekilde ölçmek amacıyla geliştirilmiş olan hızlı tam arama ve seyrek arama yaklaşımları kendi içlerinde de karşılaştırılmıştır. Elde edilmiş olan sonuçlar Tablo 4.16'da verilmiştir.

Tablo 4.16. Geliştirilmiş olan hızlı tam arama yaklaşımı ile seyrek arama yaklaşımının sonuçlarının karşılaştırılması

Görüntü Dizisi	ΔB	ΔT
Kimono	0,59	85,5
ParkScene	-0,43	85,9

Bu tablodan elde edilmiş olan değerlerden, düşük bit gösterimli yaklaşımlar kendi aralarında karşılaştırıldığında, seyrek arama yaklaşımının düşük bir bit oran/bozunumuyla hesapsal karmaşıklığı azalttığı gözlenmektedir. "ParkScene" dizisinde eksi değerler elde edilmesi de hızlı arama yaklaşımı ile daha doğru sonuçlar elde edildiğini göstermektedir. Bu durumun sebebinin seyrek arama yaklaşımı tahmin edilmiş noktalardan arama işlemine başlarken, düşük bit gösterimli hızlı tam arama yaklaşımının yerel minimumlara takılmasından kaynaklı olduğu değerlendirilmektedir.

4.3.6. Sonuçlar

Bu çalışmada SGC yaklaşımını temel alan iki yöntem geliştirilmiştir. Hızlı tam arama yönteminde pikseller 1 bitlik yapılar şeklinde ele alınarak gruplanmış, veri sıralı bir hâle getirilerek SIMD yaklaşımı ile NNMP hesabı gerçekleştirilmiştir. Seyrek arama yönteminde ise düşük bit derinliği temelli yöntemler için uygun bir hareket kestirimi algoritması geliştirilmiştir. Elde edilen deneysel sonuçlar, bit dönüşümü kaynaklı bit/oran bozunumuna rağmen geliştirilmiş olan yöntemlerin HEVC'deki hesapsal karmaşıklığı azaltarak iyi sonuçlar verdiğini göstermektedir.

4.4. HEVC için Blok Benzerliğine Dayalı Mod Seçimi

4.4.1. Giriş

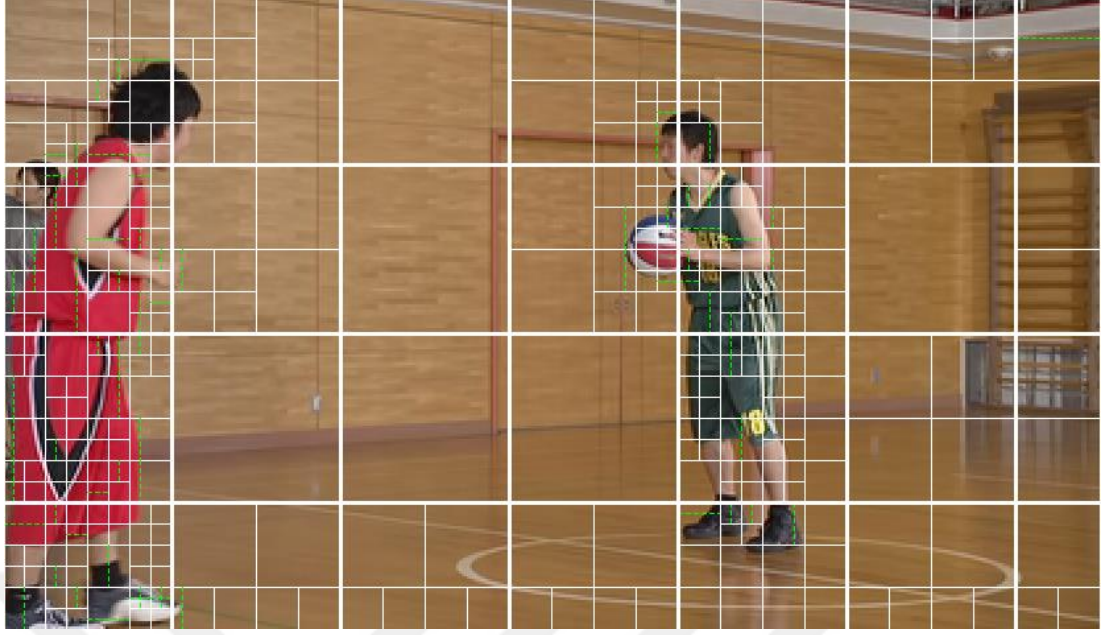
HEVC’de ağaç yapısının oluşturulması sırasında tüm derinliklerde simetrik ve asimetrik PU modları için hareket kestirimi ve maliyet hesabı yapılarak ağaç yapısına nihai şekli verilmektedir. Bu işlemler HEVC’deki hesapsal karmaşıklığın önemli bir bölümünü oluşturmaktadır. HEVC’de CTU yapısının oluşturulması Bölüm 2.2’de detaylı bir şekilde ele alınmıştır. Literatürdeki mod seçimi ile ilgili çalışmalar ise Bölüm 3.2’de verilmiştir.

Tez çalışmasının bu aşamasında mod seçimi işlemleriyle HEVC’deki hesapsal karmaşıklığı azaltmak amaçlanmıştır. Bu amaçla önerilen yöntem temel olarak iki aşamadan oluşmaktadır. Yöntemin ilk aşamasında uygun derinliğin bulunması için, ikinci aşamasında ise uygun PU modunun seçilmesi için işlemler gerçekleştirilmiş, geliştirilmiş olan yöntem HEVC referans yazılımına uygulanmıştır.

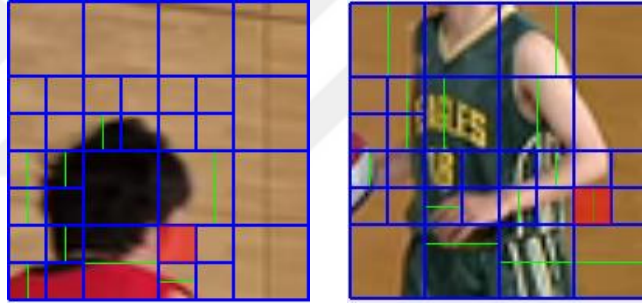
4.4.2. Önerilen Yöntem

Çalışmanın ilk aşamasında kodlama işlemi sırasında kodlayıcı tarafından seçilmiş olan CU’ların derinlikleri ve PU modları analiz edilmiştir. Şekil 4.14’ten görüldüğü üzere detay içeren bölgelerde CTU’lar küçük ve fazla sayıda CU içermekteyken, homojen yapıya sahip bölgelerde CTU’lar daha büyük ve az sayıda CU’dan oluşmaktadır. Bununla birlikte CU’ların içindeki piksellerin ışıklılık dağılımının da genellikle PU modlarının seçimini etkilediği görülmüştür. Homojen bir yapıya sahip CU’lar için genellikle $2N \times 2N$ modu seçilmekteyken, karmaşık bir yapıya sahip olan CU’larda piksellerin dağılımına göre simetrik veya asimetrik PU modlarının seçildiği gözlenmiştir.

Şekil 4.14’te, beyaz renk içinde kalan bölgeler CU’ları temsil etmekteyken, yeşil çizgilerin olduğu bölgeler CU’lar için seçilmiş olan PU’ları göstermektedir. Yeşil çizgi olmaması CU boyutunun $2N \times 2N$ olarak seçildiğini göstermektedir. Bununla birlikte homojen bölgelerde bulunan CTU’lar şekilde görüldüğü gibi tek bir CU’dan da oluşabilmektedir.



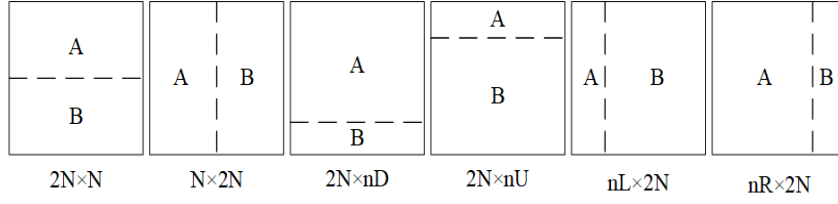
Şekil 4.14. BasketballPass dizisinin bir çerçevesi için kodlayıcı tarafından oluşturulmuş ağaç yapısı



Şekil 4.15. BasketballPass dizisinden alınmış örnek ağaç yapıları

Bu şekilde, mavi çizgiler arasında kalan bölgeler küçük boyutlu CU'ları temsil etmekteyken, yeşil çizgiler ilgili CU'lar için simetrik veya asimetrik PU modlarından birinin seçildiği göstermektedir.

Şekil 4.14 ve 4.15'den görüldüğü üzere CTU bölgelerindeki homojenlik seçilecek olan CU'lara ve bu CU'larla ilgili PU modlarının seçimine etki etmektedir. Kodlayıcının bir CU için kestirim ve maliyet hesabını sırasıyla $2N \times 2N$, $2N \times N$, $N \times 2N$, $2N \times nD$, $2N \times nU$, $nL \times 2N$ ve $nR \times 2N$ modları için gerçekleştirmektedir. $N \times N$ modu ise bir sonraki derinlikte $2N \times 2N$ moduna karşılık geldiğinden sadece 3. derinlikte dikkate alınmaktadır. Tez çalışması kapsamında bölgelerdeki homojenliği tespit etmek amacıyla $2N \times 2N$ ve $N \times N$ modu dışında kalan simetrik ve asimetrik PU modlarının boyutlarından yararlanılmıştır. Şekil 4.16'da bu modlar verilmiştir.



Şekil 4.16. İki parça içeren PU modları

Şekil 4.16'te görüldüğü üzere bu PU modları A ve B olmak üzere iki parçaya alınmıştır. Bu parçalarda bulunan piksellerin ortalamasının birbirine oranı homojenlik tespiti için bir metrik olarak ele önerilmiştir. Bu metrik Eşitlik (4.2)'deki gibi hesaplanmaktadır.

$$S_i = \frac{\max(\sum A_i/nA_i, \sum B_i/nB_i)}{\min(\sum A_i/nA_i, \sum B_i/nB_i)} \quad (4.2)$$

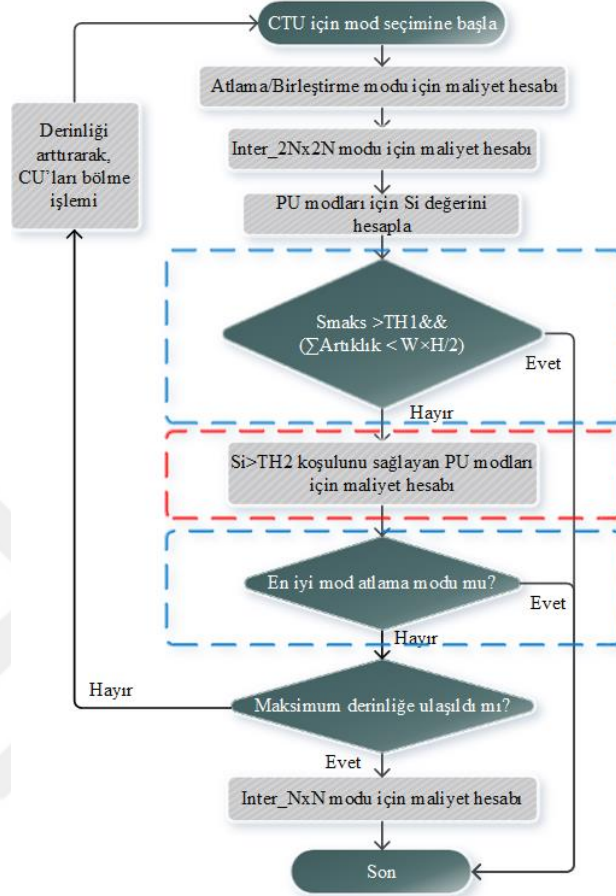
Bu eşitlikte S homojenlik metriğini temsil etmekte olup, tüm PU modları için bu değer hesaplanmaktadır. i ise PU modlarının indisini temsil etmektedir. nA ve nB A ve B bloklarındaki toplam piksel sayısını temsil etmektedir. Bu metrik hesaplanırken her zaman ortalaması büyük olan parça denklemin payında olmaktadır. Dolayısıyla bu metriğin değeri her zaman 1'den büyük olacaktır. Bu metriğin bütün PU modları için değerinin 1'e yakın olması ilgili CU'nun homojen olduğu anlamına gelmektedir.

Bu çalışmada önerilmiş olan yöntem iki aşamadan oluşmaktadır. Birinci aşama CU seviyesinde uygun derinliğe karar vererek alt derinliklerin atlanması (Yöntem 1), ikinci aşama ise simetrik ve asimetrik PU modlarının analiz edilerek sadece belirlenmiş PU modları için kestirim ve maliyet hesabı işlemlerinin gerçekleştirilmesidir (Yöntem 2). Geliştirilmiş olan algoritmanın blok şeması Şekil 4.17'de verilmiştir.

Şekil 4.17'de mavi dikdörtgenin içinde kalan bloklar CU seviyesinde derinlik belirlenmesi ile ilgili işlemleri, kırmızı dikdörtgenin içinde kalan blok ise uygun PU modunun seçilmesini göstermektedir. Bu eşitlikle W ve H değerleri sırasıyla blok genişliğini ve yüksekliğini temsil etmektedir.

Yöntemin ilk aşamasında sırasıyla atlama modu için ve $2N \times 2N$ modu için çerçeveler arası hareket kestirimi gerçekleştirildikten sonra artıklık verisiyle birlikte maliyet

hesabı gerçekleştirilmektedir. Sonraki aşamada ise Eşitlik (4.3) kullanılarak PU modları için S metriği hesaplanmaktadır.



Şekil 4.17. Geliştirilmiş olan algoritmanın blok şeması

4.4.2.1. Uygun CU derinliğinin belirlenmesi (CUD)

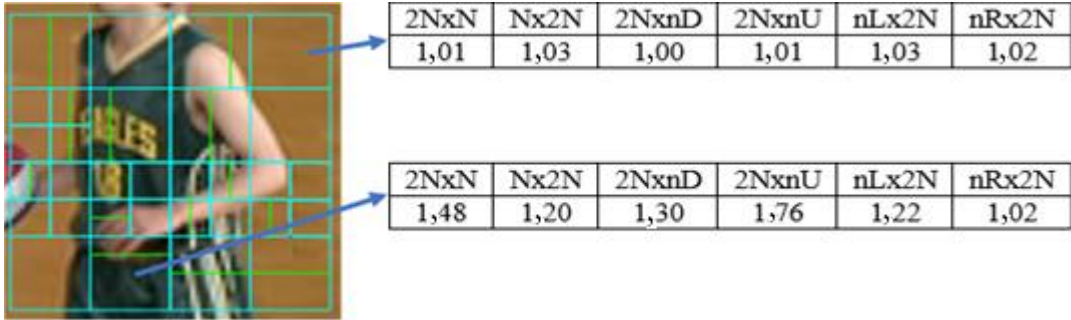
CU seviyesinde uygun derinliğe karar vermek amacıyla ilk olarak maksimum S değeri bulunmaktadır. S değerinin küçük olması bloğun homojen bir yapıda olduğu anlamına geldiğinden dolayı bu değer bir eşik değeriyle (TH1) karşılaştırılmıştır. TH1 değeri deneysel çalışmalar sonucunda 1,05 olarak belirlenmiştir. Bu eşik değerinin tespitine ilişkin detaylı analiz 4.4.3.'te verilmiştir. 2N×2N modu için gerçekleştirilmiş hareket kestirimi sonrasında elde edilen artıklık verisinin toplamı da hareket kestirimi işleminin başarımı hakkında fikir vermektedir. CU homojen bir dağılıma sahipse ve bu homojen bölge için elde edilmiş olan artıklık değeri $W \times H/2$ değerinden küçükse bu derinliğin yeterli olduğuna karar verilerek işlem sonlandırılmaktadır.

Yöntemin diğer aşamasında da HEVC'deki ECU yapısı kullanılarak PU modları için maliyet hesabı gerçekleştirildikten sonra eğer en az maliyet atlama modu ile sağlamıyorsa işlem sonlandırılmaktadır.

4.4.2.2.Uygun PU modunun belirlenmesi (PUM)

Uygun PU moduna karar vermek için de S metriği kullanılmıştır. S değerinin büyük olması bloğun karmaşık bir yapıda olduğu anlamına gelmektedir. Her PU modu için hesaplanan S değeri bir eşik değeriyle karşılaştırılarak (TH2) eşik değerinden büyük S değerine sahip PU modları için hareket kestirimi işlemi gerçekleştirilmiştir. TH2 değeri deneysel çalışmalar sonucunda 1,25 olarak belirlenmiştir. Bu eşik değerinin tespitine ilişkin detaylı analiz 4.4.3.'te verilmiştir.

Şekil 4.18'te bu yöntemle ilgili bir örnek verilmiştir. Bu şekilde turkuaz renkle işaretlenmiş bölgeler CU'ları, yeşil çizgiler ise seçilen PU modunu göstermektedir. Yeşil çizginin olmaması o CU için $2N \times 2N$ modunun seçildiği anlamına gelmektedir. Üst bölgede bulunan CU için S değerleri 1,25'ten küçük çıkmaktayken kodlayıcı $2N \times 2N$ modunu seçmiştir. Alt bölgede bulunan CU'da ise $2N \times N$, $2N \times nD$ ve $2N \times nU$ modları 1,25'ten büyük olup kodlayıcı bu CU için en yüksek S değerine sahip olan $2N \times nU$ modunu en düşük maliyete sahip olan mod olarak belirlemiştir.



Şekil 4.18. BasketballPass dizisinden alınmış örnek ağaç yapıları için S değerleri

4.4.3. Deneysel Sonuçlar

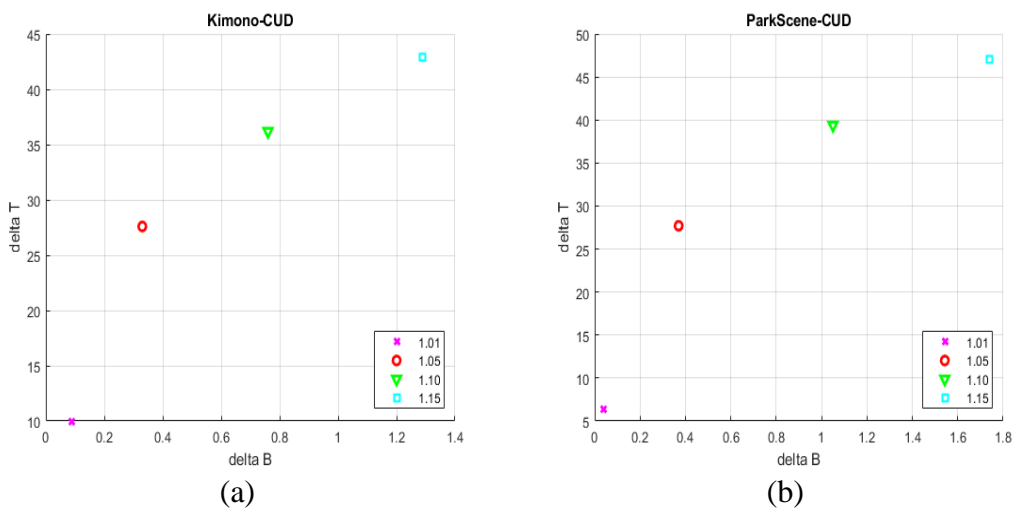
Geliştirilmiş olan yöntem HEVC'nin referans yazılımı olan HM 16.7 sürümü üzerinde gerçekleştirilmiştir. Yöntemin değerlendirmek amacıyla JCT-VC tarafından önerilmiş olan "Traffic", "Kimono", "ParkScene", "FourPeople", "Johnny" ve "KrsitenAndSara" dizileri kullanılmıştır. Tablo 2.1'de bu görüntü dizileri

özellikleriyle birlikte verilmiştir. Yöntemin başarımını değerlendirmek amacıyla ise [10]'da önerilmiş olan metrik kullanılmıştır. Bu metrikle ilgili detaylar Bölüm 1.3'de verilmiştir. Önerilmiş olan yöntem 22, 27, 32 ve 37 kuantalama katsayıları için LDB ve RA konfigürasyonlarında çalıştırılmıştır. [10]'da önerilmiş olan metrik kullanılarak karşılaştırma için ΔB değeri elde edilmiş, ΔT değeri ise Eşitlik (1.4) kullanılarak hesaplanmıştır. Tüm yöntemler arasında sağlıklı bir değerlendirme yapabilmek amacıyla da ΔB değerinin ΔT değerine oranı kullanılmıştır. Bu metrik ΔX olarak isimlendirilmiştir.

Bu bölümde verilmiş olan deneysel sonuçlar Intel i7 7800X işlemci, 16 GB RAM özelliklerine sahip bir sistem ile elde edilmiştir.

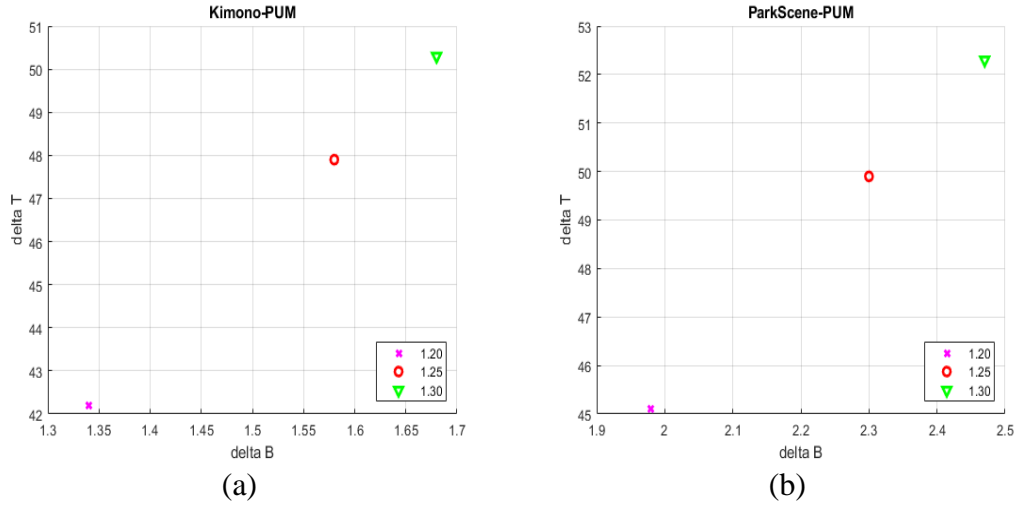
Şekil 4.19'da CUD yönteminde eşik değerini (TH1) belirlemek amacıyla farklı eşik değerleri için elde edilen ΔB - ΔT grafiği verilmiştir. Bu deney sırasında eşik değerini daha sağlıklı bir biçimde belirlemek amacıyla yöntemin ikinci kısmında gerçekleştirilen atlama modu kontrolünün etkisi değerlendirmeye alınmamıştır (Şekil 4.17).

Şekil 4.19'dan görüldüğü üzere eşik değerinin (TH1) artırılması ΔB değerinin artmasına neden olurken ΔT değerinin ise artmasını sağlamaktadır. Bu doğrultuda makul bir denge sağlamak amacıyla TH1 değeri 1,05 olarak belirlenmiştir.



Şekil 4.19. CUD için eşik değerinin belirlenmesi a) Kimono b) ParkScene

Şekil 4.20'de ise PUM yöntemindeki eşik değerini (TH2) belirlemek için farklı eşik değerleri kullanılarak elde edilen ΔB - ΔT grafiği verilmiştir.



Şekil 4.20. PUM için eşik değerinin belirlenmesi a) Kimono b) ParkScene

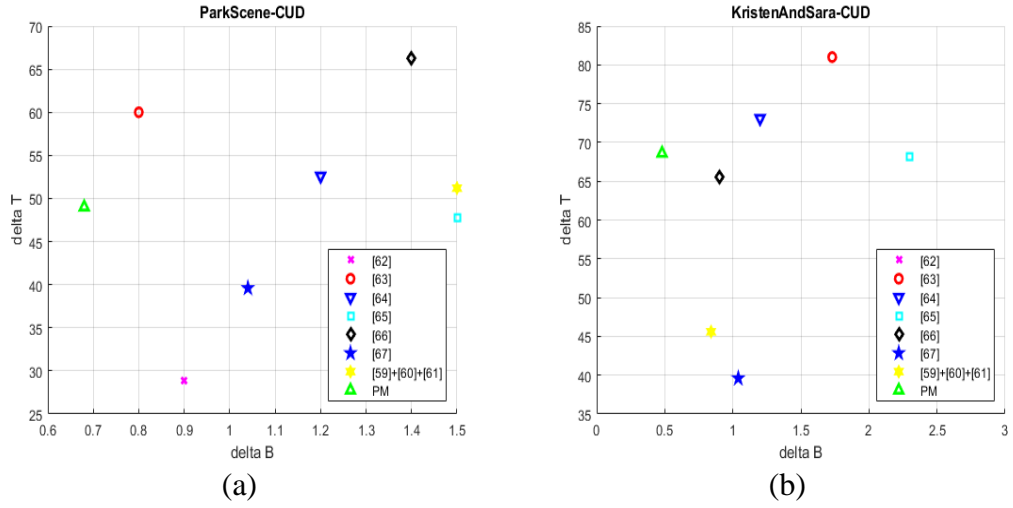
Şekil 4.20’de de Şekil 4.19’a benzer bir şekilde görüldüğü üzere eşik değerinin (TH2) arttırılması ΔB değerinin ile birlikte neden olurken ΔT değerinin ise artmasını sağlamaktadır. Bu doğrultuda makul bir denge sağlamak amacıyla TH1 değeri 1,25 olarak belirlenmiştir.

Literatürdeki çalışmalar kodlama verimliliği ve kazanılan hız arasında genellikle bir oran olduğunu göstermektedir. Bu sebepten dolayı adil bir karşılaştırma yapmak amacıyla uygun CU derinliğini belirlemek için geliştirilmiş olan yöntem literatürdeki benzer yöntemlerle (Tablo 4.17), uygun PU modunu belirlemek için geliştirilmiş olan yöntem de yine benzer yöntemlerle karşılaştırılmıştır (Tablo 4.18). İki yaklaşımın birleştirilmesiyle oluşturulmuş olan yöntem de yine benzer şekilde CU ve PU seviyesinde işlem yapan yöntemlerle karşılaştırılmıştır (Tablo 4.19-4.20).

Tablo 4.17. CUD yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB)

Yöntem	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[62]	2.00	41,3	4,85	0,30	27,8	1,08	0,90	28,9	3,11	2,40	47,8	5,02	2,00	56,0	3,53	2,80	48,5	5,77
[63]	-	-	-	1,18	58,0	2,03	0,80	60,0	1,33	1,30	82,0	1,58	1,45	86,0	1,68	1,73	81,0	2,14
[64]	0.80	61,6	1,30	1,30	58,2	2,23	1,20	52,6	4,18	1,70	74,1	2,29	1,30	75,7	1,71	1,20	73,1	1,64
[65]	2.00	56,1	3,57	2,00	56,1	3,57	1,50	47,8	3,13	2,80	66,4	4,21	2,50	70,9	3,52	2,30	68,2	3,37
[66]	1.30	59,9	2,18	1,00	44,3	2,26	1,30	56,5	2,30	1,40	66,3	2,11	1,10	70,3	1,56	0,90	65,6	1,37
[67]	1.10	40,7	2,70	0,60	42,2	1,42	1,04	39,6	2,62	0,83	45,5	1,82	0,58	46,0	1,26	0,84	45,6	1,84
[59]+[60] +[61]	1.50	56,5	2,65	1,20	43,4	2,76	1,50	51,2	2,92	1,30	70,3	1,85	1,30	77,2	1,68	1,10	70,8	1,55
ÖY	0.41	51,6	0,79	0,71	44,9	1,58	0,68	49,0	1,38	0,48	66,6	0,72	1,09	75,2	1,45	0,48	68,6	0,70

Şekil 4.21’de ise CUD yöntemi için, ParkScene ve KristenAndSara dizileri kullanılarak elde edilmiş ΔB - ΔT değerleri grafik üzerinde verilmiştir.



Şekil 4.21. CUD yöntemi için elde edilmiş olan ΔB - ΔT grafiği a) ParkScene b) KristenAndSara

Tablo 4.18. PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB)

Yöntem	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[68]	2,10	39,2	5,36	-	-	-	-	-	-	2,03	34,6	5,86	1,87	33,9	5,51	-	-	-
ÖY	2,30	41,0	5,60	1,58	47,9	3,30	2,30	49,9	4,60	2,34	46,0	4,87	2,77	54,4	5,09	-	-	-

Tablo 4.19. CUD + PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (LDB)

Yöntem	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[70]	-	-	-	3,30	75,6	4,36	4,40	77,2	5,70	4,80	85,1	5,64	5,90	88,2	6,69	5,00	85,8	5,83
[71]	3,87	60,0	6,45	2,81	59,0	4,76	2,48	55,0	4,51	-	-	-	-	-	-	-	-	-
ÖY	3,50	65,6	5,33	3,16	67,7	4,66	3,79	68,6	5,52	3,30	79,3	4,16	4,21	84,7	4,97	2,35	81,5	2,88

Tablo 4.20. CUD + PUM yönteminin literatürdeki benzer yöntemlerle karşılaştırılması (RA)

Yöntem	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[70]	4,10	81,3	5,04	3,00	76,9	3,90	3,70	79,5	4,65	-	-	-	-	-	-	-	-	-
ÖY	2,84	70,5	4,02	2,90	67,6	4,29	2,98	72,2	4,13	1,41	81,0	1,74	1,68	83,9	2,00	1,90	82,5	2,30

Bu tablolarda ΔX metriğine göre ilk sırada olan yöntemler siyah renkle, ikinci sırada olan yöntemler ise kırmızı renkle işaretlenmiştir.

Deneysel sonuçlar önerilmiş olan yöntemlerin iyi bir bit oran/bozunum ve hız dengesiyle birçok dizide en iyi sonuçları verdiğini göstermektedir. Bu da önerilmiş olan homojenlik metriğinin ve yaklaşımın iyi bir sonuç verdiğini ortaya koymaktadır.

4.4.4. Sonular

Bu alıřmada HEVC iin blok benzerliđine dayalı hızlı bir mod seimi algoritması nerilmiřtir. nerilmiř olan algoritma iki ařamadan oluřmaktadır. İlk ařamada homojenlik ve artıklık kontrol edilerek uygun derinliđe karar verilmekte, ikinci ařamada ise homojenlik metriđine gre maliyet hesabı yapılacak PU modlarına karar vermektedir. Deneysel sonular da nerilmiř olan yntemin literatrdeki yaklařımlara gre daha iyi bir bit oran/bozunum ve hız dengesi sunduđunu gstermektedir.



5.SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, belirli bir kodlama verimliği altında hesapsal karmaşıklığı düşüren yöntemler ve belirli bir hesapsal karmaşıklık altında kodlama verimliliğini arttıran yeni yaklaşımlar önerilmiştir. Deneysel sonuçlar, geliştirilmiş olan yöntemlerde uygun bir bit-oran/bozunum ve hız dengesi sağlandığını göstermektedir.

Bölüm 4.1’de ekran içerik görüntüleri için düşük-bit gösterimi temelli bir hareket kestirim yaklaşımı önerilmiştir. Önerilmiş olan yöntemde ekran içerik görüntülerinin karakteristik özellikleri gözönüne alınarak en değerlikli bit düzlemi değerlendirilmiş, bu değerlendirme sonucuna göre hangi bit düzlemlerinin uyumlama aşamasında kullanılacağına karar verilmiştir. Deneysel sonuçlarda da geliştirilmiş olan yöntemin düşük hesapsal yüküyle diğer yöntemlere göre genel olarak yüksek bir hareket kestirim başarımı gösterdiği görülmüştür.

Bölüm 4.2’de, SGC, 1BT ve C-1BT temelli hareket kestirim yaklaşımlarının HEVC referans yazılımına uygulanması üzerinde çalışılmıştır. SGC yaklaşımının 1BT ve C1-BT yöntemlerine göre yüksek başarımlar verdiği görülmekle birlikte, düşük bit gösterim temelli yaklaşımların hareket kestirim performansından genel olarak bir miktar ödün verdiği görülmüştür. Bununla birlikte uyumlama için 1’den fazla bit düzlemi kullanan C1-BT yaklaşımının diğer yöntemlere göre daha yavaş kaldığı görülmüştür. NNMP kriterini hesaplamak amacıyla kullanılan ve donanımsal olarak etkin bir şekilde gerçekleştirilebilen XOR işleminin, bilgisayar ortamında en düşük değişken biriminin byte olmasından dolayı etkin bir şekilde kullanılmadığı tespit edilmiştir.

Bölüm 4.3’te ise HEVC referans yazılımında düşük-bit gösterimli yöntemler arasında en iyi başarımları veren SGC yaklaşımı için hızlı tam arama ve seyrek arama yaklaşımları geliştirilmiştir. Bununla birlikte XOR işlemini bilgisayar ortamında verimli bir şekilde gerçekleştirebilmek için bir byte değişkeninde 8 adet dönüştürülmüş piksel değeri tutulacak şekilde uyumlama işlemi gerçekleştirilmiş, bu uyumlanmış piksel değerleri ise SIMD saklayıcıları ve komutları yardımıyla XOR işleminde kullanılarak NNMP hesabı gerçekleştirilmiştir. Bununla birlikte düşük-bit gösterimli yöntemlerin

piksellerin derinliklerini sınırlaması dolayısıyla, bu yöntemlerde bit-oran/bozunum parametresini en iyilemede bir sınırın olduğu görülmüştür. Fakat bu yaklaşımlarda, düşük-bit gösterimi sayesinde hesapsal karmaşıklık azaltılabildiği için bir denge sağlandığı değerlendirilmektedir.

HEVC’de CTU adı verilen kodlama ünitelerinin, çerçeve boyutlarına bağlı olarak eski video kodlama standartlarındaki kodlamaya ünitelerinin boyutlarına göre artma yönünde bir eğilim gösterdiği, bu eğilim ile birlikte uygun derinlik ve mod seçme işlemlerinin HEVC referans yazılımındaki hesapsal karmaşıklığının azaltılmasına hareket kestirimine oranla daha fazla katkıda bulunduğu görülmüştür. Bu amaçla Bölüm 4.4’te HEVC için blok benzerliğine hızlı bir mod seçimi yaklaşımı önerilmiştir. PU bloklarının homojenliği ve artıklık verisi kontrol edilerek CU’lar için uygun derinliğine karar verilmiş, PU bloklarının homojenliğine göre de maliyet hesabı gerçekleştirilecek PU modları azaltılmıştır. Deneysel sonuçlarda da önerilmiş olan yöntemin literatürdeki yaklaşımlara göre daha iyi bir bit oran/bozunum ve hız dengesi sunduğu görülmüştür. Bununla birlikte, FVC standardı ile kodlama ünitelerinin boyutlarının 128×128 olmasının planlandığı da gözönüne alınırsa bu yaklaşımların daha da önem kazanacağı değerlendirilmektedir.

6.KAYNAKLAR

- [1] ITU-T, Video Codec for Audiovisual services at $p \times 64$ kbits, *Telecommunication Standardization Sector of ITU*, Geneva, 1990.
- [2] ISO/IEC, Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, *International Organization for Standardization*, Geneva, 1993.
- [3] Sullivan G. J., Ohm J. R., Han W.-J., Wiegand T., Overview of the High Efficiency Video Coding (HEVC) standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1648–1667.
- [4] Natarajan B., Bhaskaran V., Konstantinides K., Low-Complexity Block-Based Motion Estimation via One-Bit Transforms, *IEEE Transactions on Circuits and Systems for Video Technology*, 1997, **7**(2), 702-706.
- [5] Urhan O., Ertürk S., Constrained one-bit transform for low-complexity block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, **17**(4), 478-482.
- [6] Yavuz S., Çelebi A., Aslam M., Urhan O., Selective Gray-Coded Bit-Plane Based Low Complexity Motion Estimation and its Hardware Architecture, *IEEE Transactions on Consumer Electronics*, 2016, **62**(1), 76-84.
- [7] Bossen F., Bross B., Suhring K., Flynn D., HEVC Complexity and Implementation Analysis, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1685-1696.
- [8] Mesut A., Veri Sıkıştırma Yeni Yöntemler, Doktora Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne, 2006, 183894.
- [9] Ohm J. R., Sullivan G. J., Schwarz H., Tan T. K., Wiegand T., Comparison of The Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC), *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1669-1684.
- [10] Bjøntegaard G., Calculation of average PSNR differences between RD-curves, *ITU-T*, VCEG-M33,1-4, 2001.
- [11] Zeng J., Au O., Dai W., Kong Y., Jia L. Zhu W., A Tutorial on Image/Video Coding Standarts, *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Kaohsiung, Tayvan, 29-30 Ekim 2013.

- [12] Roden T. V., H.261 and MPEG1-a comparison, *IEEE 15. Annual International Phoenix Conference on Computers and Communications*, Scottsdale, Amerika Birleşik Devletleri, 27-29 Mart 1996.
- [13] Liu S., Performance comparison of MPEG1 and MPEG2 video compression standards, *COMPCON '96. Technologies for the Information Superhighway Digest of Papers*, Santa Clara, Amerika Birleşik Devletleri, 25-28 Şubat 1996.
- [14] Ashraf G., Chong M.N., Performance analysis of H.261 and H.263 video coding algorithms, *ISCE '97. Proceedings of 1997 IEEE International Symposium on Consumer Electronics*, Singapur, Singapur, 2-4 Aralık 1997.
- [15] Koenen, R., "Overview of the MPEG-4 Standard", *ISO/IEC, JTC 1/SC 29/WG 11/N4668*. 2002.
- [16] Akbulut O., Bir-bit Dönüşümü Temelli Blok Hareket Kestirimlerinin H.264/AVC'ye uygulanması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2007, 232752.
- [17] Sullivan G. J., Rate-Distortion Optimization for Video Compression, *IEEE Signal Processing Magazine*, 1988, **15**(6), 74-90.
- [18] Madhuvappan C. A., Video Compression Motion Estimation Algorithms – A Survey, *International Journal of Scientific & Engineering Research*, 2014, **5**(2), 1048-1054.
- [19] Jain J., Jain A., Displacement Measurement and Its Application in Interframe Image Coding, *IEEE Transactions on Communications*, 1981, **29**(12), 1799-1808.
- [20] Koga T., Linuma K., Hirano A., Lijima Y., Ishiguro T., Motion compensated interframe coding for video conferencing, *National Telecommunication Conference Proceedings*, New Orleans, Amerika Birleşik Devletleri, 29 Kasım – 3 Aralık 1981.
- [21] Li R., Zeng B., Liou, M. L., A New Three-Step Search Algorithm for Block Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 1994, **4**(4), 438-442.
- [22] Po L. M., Ma, W. C., A Novel Four-Step Search Algorithm for Fast Block Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, **6**(3), 313-317.
- [23] Ghanbari M. The cross-search algorithm for motion estimation (image coding), *IEEE Transactions on Communications*, 1990, **38**(7), 950-953.
- [24] Zhu S., A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation, *IEEE Transactions on Image Processing*, 2000, **9**(2), 287-290.

- [25] Zhu C., Lin X., & Chau L. P., Hexagon-Based Search Pattern for Fast Block Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, **12**(5), 349-355.
- [26] Bierling M., Displacement Estimation by Hierarchical Block Matching, in *Proc. of SPIE Visual Commun. Image Processing (VCIP'88)*, 1988.
- [27] Liu B., Zaccarin A., New Fast Algorithms for the Estimation of Block Motion Vectors, *IEEE Transactions on Circuits and Systems for Video Technology*, 1993, **3**(2), 148–157.
- [28] Xiao W., Shi G., Li B., Xu J., Wu F., Fast Hash-Based Inter-Block Matching for Screen Content Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, DOI: 10.1109/TCSVT.2016.2643701.
- [29] Ertürk S., Multiplication-free one-bit transform for low-complexity block-based motion estimation, 2007, *IEEE Signal Processing Letters*, **14**(2), 109-112.
- [30] Ertürk A., Ertürk S., Two bit transform for binary block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, **15**(7), 938-946.
- [31] Çelebi A., Akbulut O., Urhan O., Ertürk S., Truncated gray-coded bit-plane matching based motion estimation and its hardware architecture, *IEEE Transactions on Consumer Electronics*, 2009, **55**(3), 1530-1536.
- [32] Çelebi A.T., Yavuz S., Çelebi A., Urhan O., Selective Gray-Coded Bit-Plane-Based Two-Bit Transform and Its Efficient Hardware Architecture for Low-Complexity Motion Estimation, *IEEE Transactions on Consumer Electronics*, 2018, **64**(3), 259-266.
- [33] Sun T, Wan P, Au OC, Dai W, Jia L, Yuan Y, Zheng A, Ma R., Fast binary motion estimation for screen content video coding, *2014 Asia- Pacific Signal and Information Processing Association Annual Summit and Conference*, Wat, Kamboçya, 9-12 Aralık 2014.
- [34] Choi, C., Jeong, J., Enhanced two-bit transform based motion estimation via extension of matching criterion, 2010, *IEEE Transactions on Consumer Electronics*, **56**(3), 1883-1889.
- [35] Güllü M. K., Weighted constrained one-bit transform based fast block motion estimation, 2011, *IEEE Transactions on Consumer Electronics*, **57**(2), 751-755.
- [36] Lee, S., Jeon, G., Jeong, G., Fast motion estimation based on enhanced constrained one-bit transform, *Electronics Letters*, **50**(10), 746–748, 2014.
- [37] Song C.-M., Guo Y., Wang X.-H, Liu D., Fuzzy quantization based bit transform for low bit-resolution motion estimation, *Signal Processing: Image Communication*, **28**(10), 1435-1447, 2013.

- [38] Chalidabhongse J., Kuo C. C. J., Fast motion vector estimation using multiresolution-spatio-temporal correlations, *IEEE Transactions on Circuits and Systems for Video Technology*, 1997, **7**(3), 477-488.
- [39] Nie Y., Ma K. K., Adaptive rood pattern search for fast block-matching motion estimation, *IEEE Transactions on. Image Processing*, 2002, **11**(12), 1442-1448.
- [40] Tzovaras D., Strintzis M.G., Sahinolu H., Evaluation of Multiresolution Block Matching Techniques for Motion and Disparity Estimation, *Signal Processing: Image Communication*, 1994, **6**(1), 56-67.
- [41] Lee J.H., Lim K.W., Song B.C., Ra J.B., A Fast Multiresolution Block Matching Algorithm and its VLSI Architecture for Low Bit-rate Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 2001, **11**(12),1289-1301.
- [42] Li W., Salari E., Successive Elimination Algorithm for Motion Estimation, *IEEE Transactions on Image Processing*, 1995, **4**(1), 105-107.
- [43] Cheung C.K., Po L.M., Normalized Partial Distortion Search Algorithm for Block Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2000, **10**(3), 417-422.
- [44] Chen Y.S., Huang Y.P., Fuh C.S., Fast Block Matching Algorithm based on the Winner-update Strategy, *IEEE Transactions on. Image Processing*, 2001, **10**(8), 1212-1222.
- [45] Duanmu C. J., Ahmad M. O., & Swamy M. N. S., Fast Block Motion Estimation With 8-Bit Partial Sums Using SIMD Architectures, *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, **17**(8), 1041-1053.
- [46] Kuo C. M., Kuan, Y. H., Hsieh C. H., Lee Y. H., A Novel Prediction-Based Directional Asymmetric Search Algorithm for Fast Block-Matching Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2009, **19**(6), 893-899.
- [47] Zhang R., Pan Z., Ku, W., A Novel Parallel Implementation of Partial Distortion Search Algorithm Based on Template Search, *Multimedia Tools and Applications*, 2018, **77**(16), 20615-20628.
- [48] Urhan O., Constrained one-bit transform based motion estimation using predictive hexagonal pattern, *Journal of Electronic Imaging*, 2007, **16**(3), 033019.
- [49] Urhan O., Constrained one-bit transform based fast block motion estimation using adaptive search range, *IEEE Transactions on Consumer Electronics*, 2010, **56**(3), 1868-1871.

- [50] Yang S. H., Jiang J. Z., Yang H. J., Fast motion estimation for HEVC with directional search, *Electronics Letters*, 2014, **50**(9), 673-675.
- [51] Fan R., Zhang Y., Li B., Motion Classification-Based Fast Motion Estimation for High-Efficiency Video Coding, *IEEE Transactions on Multimedia*, 2017, **19**(5), 893-907.
- [52] Khemiri R., Kibeya H., Sayadi F. E., Bahri N., Atri M., Masmoudi N., Optimisation Of HEVC Motion Estimation Exploiting SAD and SSD GPU-Based Implementation, *IET Image Processing*, 2018, **12**(2), 243-253.
- [53] Lee D., Sim D., Cho K., Oh S. J., Fast Motion Estimation for HEVC on Graphics Processing Unit (GPU), *Journal of Real-Time Image Processing*, 2016, **12**(2), 549-562.
- [54] Ahn Y. J., Hwang T. J., Sim D. G., Han W. J., Implementation of Fast HEVC Encoder Based on SIMD and Data-level Parallelism, *EURASIP Journal on Image and Video Processing*, DOI: 10.1186/1687-5281-2014-16.
- [55] Hu N., Yang E., Fast Motion Estimation Based on Confidence Interval, *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, **24**(8), 1310-1322.
- [56] Singh K., Ahamed S. R., Computationally Efficient Motion Estimation Algorithm for HEVC, *Journal of Signal Processing Systems*, 2018, **90**(12), 1713-1727.
- [57] Purnachand N., Alves L. N.i Navarro A., 2012 *IEEE Second International Conference on Consumer Electronics*, Berlin, Almanya, 3-5 Eylül 2012.
- [58] Varma K. C., Kumar M. V. P., Mahapatra, S., Search Range Reduction for Uni-Prediction and Bi-Prediction in HEVC, *Journal of Real-Time Image Processing*, DOI:10.1007/s11554-016-0636-5.
- [59] Gweon R H., and Lee Y. L., Early Termination of CU Encoding to Reduce HEVC Complexity, *Joint Collaborative Team on Video Coding*, F045, 1-4, 2011.
- [60] Choi K, Park S. H., Jang E. S., Coding Tree Pruning Based CU Early Termination, *Joint Collaborative Team on Video Coding*, F092, 1-4, 2011.
- [61] Yang J., Kim J., Won K., Lee H., Jeon B., Early SKIP Detection for HEVC, *Joint Collaborative Team on Video Coding*, G543,1-4, 2011.
- [62] Shen L., Liu Z., Zhang X., Zhao W., Zhang Z., An Effective CU Size Decision Method for HEVC Encoders, *IEEE Transactions on Multimedia*, 2013, **15**(2), 465-470.
- [63] Lee J., Kim S., Lim K., Lee S., A Fast CU Size Decision Algorithm for HEVC, *IEEE Transactions on Circuits and Systems for Video Technology*, 2015, **25**(3), 411-421.

- [64] Ahn S., Lee B., Kim M., A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015, **44**(3), 422 - 435.
- [65] Xu L., Pan Z., Yuan H., Wang X., Kwong S., Zhang Y., Machine Learning-Based Coding Unit Depth Decisions for Flexible Complexity Allocation in High Efficiency Video Coding, *Delay Differential Equations: Recent Advances and New Directions*, DOI: 10.1109/TIP.2015.2417498.
- [66] Li Y., Yang G., Zhu Y., Ding X., Sun X., Adaptive Inter CU Depth Decision for HEVC Using Optimal Selection Model and Encoding Parameters, *IEEE Transactions on Broadcasting*, 2017, **63**(3), 535-546.
- [67] Bae J. H., Sunwoo M. H., Adaptive Early Termination Algorithm Using Coding Unit Depth History in HEVC, *Journal of Signal Processing Systems*, DOI: 10.1007/s11265-018-1399-y.
- [68] Huang X., Zhang Q., Zhao X., Zhang W., Zhang Y., Gan Y., Fast Inter-Prediction Mode Decision Algorithm for HEVC, *Signal, Image and Video Processing*, 2017, **11**(1), 33-40.
- [69] Vanne J., Viitanen M., Hamalainen T. D., Efficient Mode Decision Schemes for HEVC Inter Prediction, *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, **24**(9), 1579-1593.
- [70] Zhang, B. Li, and Li H., An Efficient Fast Mode Decision Method for Inter Prediction in HEVC, *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, **26**(8), 1502-1515.
- [71] Mallikarachchi T., Talagala D. S., Arachchi H. K., Fernando A., Content-Adaptive Feature-Based CU Size Prediction for Fast Low-Delay Video Encoding in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, **28**(3), 693-705.

KİŞİSEL YAYINLAR VE ESERLER

- [1] **Duvar R.**, Akbulut O., Urhan O., Fast Mode Decision Exploiting Block Similarity in HEVC, *Signal Processing: Image Communication*, (Değerlendirilmek üzere gönderilmiştir).
- [2] **Duvar R.**, Urhan O., Adaptive Bit-Plane Selection Based Low Complexity Motion Estimation for Screen Content Coding, *Turkish Journal of Electrical Engineering & Computer Sciences*, 2018, **26**(5), 2206-2217.
- [3] **Duvar R.**, Küçükmanisa A., Akbulut O., Çelebi A. T., Urhan O., Assessment of the Performance of Low Bit-Depth Representation based Motion Estimation Methods on HEVC, *Journal of the Faculty of Engineering and Architecture of Gazi University*, DOI: 10.17341/gazimmfd.416537.
- [4] **Duvar R.**, Küçükmanisa A., Akbulut O., Çelebi A. T., Urhan O., HEVC’de Seyrek Arama Kullanarak Bir-Bit Dönüşümlü Hareket Kestiriminin Performans Analizi, *26. Sinyal İşleme ve İletişim Uygulamaları Konferansı*, İzmir, Türkiye, 2-5 Mayıs 2018.
- [5] Küçükmanisa A., **Duvar R.**, Urhan O., Geliştirilmiş 1-Bit Dönüşümü Temelli Ön İşlem Kullanarak Gerçek Zamanlı Şerit İşaret Tespiti, *25. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Antalya, Türkiye, 15-18 Mayıs 2017.
- [6] Akbulut O., **Duvar R.**, Küçükmanisa A., Urhan O., Hareket Kestiriminde çoklu Bant Geçiren Süzgeç Tabanlı Bir Bit Dönüşümleri, *25. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Antalya, Türkiye, 15-18 Mayıs 2017.
- [7] **Duvar R.**, Küçükmanisa A., Akbulut O., Çelebi A. T., Urhan O., 1-Bit Dönüşümü Temelli Hareket Kestiriminin HEVC Kodlayıcıya Uygulanması, *25. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Antalya, Türkiye, 15-18 Mayıs 2017.
- [8] **Duvar R.**, Urhan O., Düşük Bit-Derinliği Temelli Hareket Kestirim Yaklaşımlarının Ekran İçerik Görüntülerindeki Başarımlarının Değerlendirilmesi, *24. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Zonguldak, Türkiye, 16-19 Mayıs 2016.
- [9] Çelebi A.T., **Duvar R.**, Urhan O., Mobil Bir Cihaz Üzerinde Tek Giriş Görüntüsü ile Yüksek Dinamik Aralıklı Görüntüleme, *23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Malatya, Türkiye, 16-19 Mayıs 2015.
- [10] Çelebi A.T., **Duvar R.**, Urhan O., Fuzzy Fusion Based High Dynamic Range Imaging using Adaptive Histogram Separation, *IEEE Transactions on Consumer Electronics*, 2015, **61**(1), 119-127.

- [11] **Duvar R.**, Urhan O., 1-Boyutlu Projeksiyon Temelli Görüntü Stabilizasyonunun Mobil Bir Cihaz Üzerinde Gerçek Zamanlı Gerçeklenmesi, 22. *Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU)*, Trabzon, Türkiye, 23-25 Nisan 2014.
- [12] Tekyıldız A., Güvenel Ç., **Duvar R.**, Çelebi A., Urhan O., Güllü M.K., Ertürk S., Panoramik Kamera Sistemi: PAN-KAM, *Gömülü Sistemler ve Uygulamaları Sempozyumu (GömSis)*, İstanbul, Türkiye, 29-30 Kasım 2012.
- [13] **Duvar R.**, Gökbayrak A.B., Aytekin H.S., Güllü M.K., Urhan O., Ertürk S., Lastik Profillerinin Bilgisayarda Görü Yaklaşımı ile Gerçek Zamanlı Kontrolü, 20. *Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU)*, Fethiye, Türkiye, 18-20 Nisan 2012.



ÖZGEÇMİŞ

1988 yılında Tekirdağ'ın Çorlu ilçesinde doğdu. İlköğretim öğrenimini Burdur Gazi İlköğretim Okulu'nda, lise öğrenimini ise Ankara Hasan Ali Yücel Anadolu Öğretmen Lise'sinde tamamladı. 2006 yılında kazandığı Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü'nden 2010 yılında Elektronik ve Haberleşme Mühendisi olarak mezun oldu. 2010 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Ana Bilim Dalı'nda yüksek lisans öğrenimine başladı. Lastik Profilleri için Gerçek Zamanlı Kalite Kontrolü adlı Yüksek Lisans tezi ile 2012 yılında Yüksek Mühendis ünvanını aldı. 2012 yılından beri Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Anabilim Dalı'nda doktora eğitimine devam etmektedir.

Şubat 2012'den itibaren Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümünde Araştırma Görevlisi olarak görev yapmaktadır.