

**HAREKET DENGELİMLİ VE LİFTİNG TABANLI
VİDEO KODLAYICI TASARIMI**

Sedat TELÇEKEN
Doktora Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Haziran-2010

JÜRİ VE ENSTİTÜ ONAYI

Sedat Telçeken'in "Hareket Dengelemeli ve Lifting Tabanlı Video Kodlayıcı Tasarımı" başlıklı **Elektrik-Elektronik Mühendisliği** Anabilim Dalındaki Doktora Tezi 05.07.2010 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı Soyadı	İmza
Üye (Tez Danışmanı) :	Prof. Dr. ÖMER NEZİH GEREK
Üye	: Prof. Dr. ATALAY BARKANA
Üye	: Yard. Doç. Dr. CÜNEYT AKINLAR
Üye	: Doç. Dr. YUSUF OYSAL
Üye	: Yard. Doç. Dr. EROL SEKE

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Doktora Tezi

HAREKET DENGELLEMELİ VE LİFTİNG TABANLI VIDEO KODLAYICI TASARIMI

Sedat TELÇEKEN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ömer Nezih GEREK

2010, 85 sayfa

Bu tezde, hareket dengeleme yöntemini kullanan yükseltme tabanlı hibrit bir kodlayıcı tasarlanarak görüntü kalitesinden fark edilir düzeyde ödün vermeden video sinyallerinin sıkıştırma oranının artırılması üzerine çalışılmıştır. Bu amaçla video çerçeveleri zamansal doğrultuda gruplara ayrılmış ve zamansal yükseltme tabanlı ayrıştırma işlemi blok tabanlı hareket dengeleme ile birlikte kullanılarak çerçeveler öngörülmeye çalışılmıştır. Öngörü için kenar-uyarlamalı yükseltme ve blok karşılaştırma yöntemleri farklı düzeylerde kullanılmıştır. Üzerinde öngörü işlemi yapılmayan çerçeveler ve öngörü çerçevelerinin orijinallerinden farkları nicelenerek kodlanmıştır. Kodlamanın başarımını değerlendirmek için PSNR ölçümleri kullanılmıştır. Deneysel sonuçlar, blok karşılaştırma düzeyi arttıkça daha fazla sıkıştırma elde edilirken PSNR değerinin kabul edilebilir seviyelerde kaldığını göstermiştir. Buna göre 37,04 : 1 oranında sıkıştırılarak kodlanan videolar 29 dB sinyal gürültü seviyesinde geri çatılmıştır.

Anahtar Kelimeler: Hareket Dengeleme, Yükseltme, Kenar-Uyarlamalı Yükseltme, Blok Karşılaştırma, Video Kodlama, Dalgacıklar

ABSTRACT

PhD Dissertation

MOTION COMPENSATED AND LIFTING BASED VIDEO CODER DESIGN

Sedat TELÇEKEN

Anadolu University

Graduate School of Sciences

Electrical and Electronics Engineering Program

Supervisor: Prof. Dr. Ömer Nezh GEREK

2010, 85 pages

In this thesis, it is aimed to compress video signals, without compromising the quality, by using a lifting based hybrid encoder which uses motion compensation method. For this purpose, video frames are separated into temporal groups and some frames in each group are predicted from other frames using temporal lifting with motion compensation. Edge-adaptive lifting and block-matching methods with different levels are used for the prediction. Unpredicted frames and residues between predicted frames and their originals are quantized and encoded. PSNR measurements are used to measure the performance of the suggested encoding method. Experimental results reveal that higher compression ratios are obtained as block-matching level is increased, while PSNR values remain in reasonable ranges. 29 dB of PSNR value is obtained in a 37.04 : 1 compression ratio.

Keywords: Motion Compensation, Lifting, Edge-Adaptive Lifting, Block Matching, Video Coding, Wavelets

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	ix
TERİMLER DİZİNİ	xi
1. GİRİŞ	1
1.1. Video Sıkıştırma Yöntemleri ve Standartları	2
1.2. DWT ile Video Kodlama Alanında Yapılmış Çalışmalar	5
2. İMGE VE VİDEO SIKIŞTIRMANI TEMELLERİ	7
2.1. İnsan Görme Sistemi ve Sıkıştırma	7
2.2. İzgesel Fazlalık	7
2.3. Uzamsal Fazlalık	8
2.3.1. Ayrık Kosinüs Dönüşümü (DCT)	9
2.3.2. Ayrık Dalgacık Dönüşümü (DWT)	13
2.3.3. DCT ile DWT'nin karşılaştırılması	17
2.4. Zamansal Fazlalık	18
2.5. İstatistiksel Fazlalık	18
3. HAREKET DENGELLEMELİ VE LİFTİNG TABANLI VİDEO KODLAYICI TASARIMI	19
3.1. Hareket Dengeleme ve Hareket Kestirimi	19
3.2. Zamansal Lifting Şeması	22
3.3. Fark Verinin Kodlanması	25
3.4. Önerilen Hareket Dengelemeli ve Zamansal Lifting Hibrit Modeli	26
3.4.1. Hareket Dengelemede Kenar-Uyarlamalı Yükseltme Yöntemi	26
3.4.2. Hareket Dengelemede Blok Karşılaştırma Yöntemi	33

3.4.3. Hareket Dengelemede Farklı Düzeylerde Blok Karşılaştırma Yöntemi	37
.....	37
4. DENEYSEL SONUÇLAR	56
4.1. Hareket Dengelemede Kenar Uyarlamalı Yükseltme Yöntemi'nin	
Deneysel Sonuçları.....	58
4.2. Hareket Dengelemede Blok Karşılaştırma Yöntemi'nin Deneysel	
Sonuçları	62
4.3. Hareket Dengelemede Farklı Düzeylerde Blok Karşılaştırma Yöntemi'nin	
Deneysel Sonuçları.....	65
5. SONUÇLAR	75
KAYNAKLAR	78
Ek-1 RGB ↔ YUV Renk Uzayı Dönüşümü	82
Ek-2 Ayrık Dalgacık Dönüşümü	83

ŞEKİLLER DİZİNİ

Şekil 2.1.	YUV Sistemi. (a) 4:2:2; (b) 4:1:1.	8
Şekil 2.2.	"Lena" imgesi üzerinde 8x8 Blok DCT.....	10
Şekil 2.3.	Zig-Zag tarama ile DCT katsayılarının elde edilmesi	11
Şekil 2.4.	"Lena" imgesinin (a) 4 katsayı (%6,25), 30.62 dB; (b) 2 katsayı (%3,125), 26,97 dB ile geri çatılması.	13
Şekil 2.5.	İmge üzerinde 1-Düzy 2B DWT: (a) Satır doğrultusunda DWT; (b) Sütun doğrultusunda DWT	14
Şekil 2.6.	2B DWT'nin Piramidal hiyerarşisi.....	14
Şekil 2.7.	"Lena" imgesi üzerinde DWT (a) 1-Düzy Ayrışım; (b) 2-Düzy Ayrışım.	15
Şekil 2.8.	Zerotree kodlama yapısı.....	17
Şekil 3.1.	En iyi eşleme hareket tahmini.....	22
Şekil 3.2.	Lifting yöntemiyle 1B DWT.....	23
Şekil 3.3.	Video dizisinde $F1$, $F2$, $F3$ Çerçveleri kullanılarak öngörü ile $R2$ kalan çerçvenin bulunması.....	26
Şekil 3.4.	$2i - 1$. çerçve ve piksellerinin gösterimi	27
Şekil 3.5.	$2i$. çerçve ve piksellerin gösterimi	28
Şekil 3.6.	$2i + 1$. çerçve ve piksellerin gösterimi	28
Şekil 3.7.	Ardışık üç çerçvenin zamansal doğrultuda gösterimi	29
Şekil 3.8.	Merkez pikseli aralarına alan 9 farklı gradyan yaklaşımı.....	30
Şekil 3.9.	Bloklar üzerindeki merkez piksellerin eşleştirme ile tahmin edilecek olan pikselin konumlandırılması.....	32
Şekil 3.10.	Blok karşılaştırma ile tahmin edilecek piksel için blokların konumlandırılması	33
Şekil 3.11.	Kenar uyarlamalı yükseltme ve blok karşılaştırma yöntemleri görselleri	35
Şekil 3.12.	Hareket dengelemede Düzy1, Düzy2 ve Düzy3 blok karşılaştırma yöntemi	38
Şekil 3.13.	Bus videosu Düzy1 Blok Eşleştirme Yöntemi Görselleri.....	40
Şekil 3.14.	Bus videosu Düzy2 Blok Eşleştirme Yöntemi Görselleri.....	43

Şekil 3.15.	<i>Bus</i> videosu Düzey3 Blok Eşleştirme Yöntemi Görselleri	45
Şekil 3.16.	<i>Coastguard</i> videosu Düzey1 Blok Eşleştirme Yöntemi Görselleri ..	47
Şekil 3.17.	<i>Coastguard</i> videosu Düzey2 Blok Eşleştirme Yöntemi Görselleri ..	48
Şekil 3.18.	<i>Coastguard</i> videosu Düzey3 Blok Eşleştirme Yöntemi Görselleri ..	50
Şekil 3.19.	<i>Container</i> videosu Düzey1 Blok Eşleştirme Yöntemi Görselleri	52
Şekil 3.20.	<i>Container</i> videosu Düzey2 Blok Eşleştirme Yöntemi Görselleri	53
Şekil 3.21.	<i>Container</i> videosu Düzey3 Blok Eşleştirme Yöntemi Görselleri	55
Şekil 4.1.	" <i>Lena</i> " imgesinin SPIHT kodlayıcı ile 0.5 bpp oranında sıkıştırılması ile elde edilen sonuçlar	57
Şekil 4.2.	" <i>bus</i> " videosunun 2. çerçevesinin SPIHT kodlayıcı ile 0.5 bpp oranında sıkıştırılması ile elde edilen sonuçlar	58
Şekil 4.3.	Düzey1 ayrışımının ve geri çatılmasının F1 ve F3 çerçeveleriyle gösterimi	59
Şekil 4.4.	" <i>bus</i> " videosu için PSNR – (1/CR) grafiği.	60
Şekil 4.5.	" <i>coastguard</i> " videosu için PSNR – (1/CR) grafiği	61
Şekil 4.6.	" <i>container</i> " videosu için PSNR – (1/CR) grafiği	62
Şekil 4.7.	Düzey2 ayrışımının ve geri çatılmasının F1 ve F5 çerçeveleriyle gösterimi	65
Şekil 4.8.	Düzey3 ayrışımının ve geri çatılmasının F1 ve F9 çerçeveleriyle gösterimi	66

ÇİZELGELER DİZİNİ

Çizelge 3.1. Kenar Uyarlamalı Yükseltme ve Blok karşılaştırma yöntemlerinin OKH değerleri (%).....	34
Çizelge 3.2. "Bus" videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%).....	36
Çizelge 3.3. "Coastguard" videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%).....	36
Çizelge 3.4. "Container" videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%).....	37
Çizelge 3.5. Bus videosunun ilk 9 çerçevesi üzerinde 3x3 arama bloğu ile 4 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%).....	45
Çizelge 3.6. Coastguard videosunun ilk 9 çerçevesi üzerinde 7x7 arama bloğu ile 2 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%).....	50
Çizelge 3.7. Container videosunun ilk 9 çerçevesi üzerinde 7x7 arama bloğu ile 2 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%).....	55
Çizelge 4.1. "bus" videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri.	59
Çizelge 4.2. "coastguard" videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri.	60
Çizelge 4.3. "container" videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri	61
Çizelge 4.4. "bus" videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri	63
Çizelge 4.5. "bus" videosunun Düzey1'de elde edilen PSNR grafiği	63
Çizelge 4.6. "coastguard" videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri ...	63
Çizelge 4.7. "coastguard" videosunun Düzey1'de elde edilen PSNR grafiği.....	64

Çizelge 4.8. " <i>container</i> " videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri ...	64
Çizelge 4.9. " <i>container</i> " videosunun Düzey1'de elde edilen PSNR grafiği	64
Çizelge 4.10. " <i>bus</i> " videosunun farklı bpp oranları ile Düzey2 ile elde edilen PSNR değerleri	67
Çizelge 4.11. " <i>bus</i> " videosunun Düzey2'de elde edilen PSNR grafiği	68
Çizelge 4.12. " <i>coastguard</i> " videosunun farklı bpp oranları ile Düzey2 ile elde edilen PSNR değerleri	68
Çizelge 4.13. " <i>coastguard</i> " videosunun Düzey2'de elde edilen PSNR grafiği	69
Çizelge 4.14. " <i>container</i> " videosunun farklı bpp oranları ile Düzey2 ile elde edilen PSNR değerleri	69
Çizelge 4.15. " <i>container</i> " videosunun Düzey2'de elde edilen PSNR grafiği.....	70
Çizelge 4.16. Tüm videoların Düzey2 PSNR-1 değerlerinin grafiği.....	70
Çizelge 4.17. " <i>bus</i> " videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri	71
Çizelge 4.18. " <i>bus</i> " videosunun Düzey3'te elde edilen PSNR grafiği	71
Çizelge 4.19. " <i>coastguard</i> " videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri	72
Çizelge 4.20. " <i>coastguard</i> " videosunun Düzey3'te elde edilen PSNR grafiği.....	72
Çizelge 4.21. " <i>container</i> " videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri	73
Çizelge 4.22. " <i>container</i> " videosunun Düzey3'te elde edilen PSNR grafiği.....	73
Çizelge 4.23. Tüm videoların Düzey3 PSNR-1 değerlerinin grafiği.....	74

SİMGELER VE KISALTMALAR DİZİNİ

BBMC	: Block-Based Motion Compensation
BPP	: Bit-per-pixel
BMA	: Block-Matching Algorithm
CD-ROM	: Compact Disc Read-Only Memory
CIF	: Common Intermediate Form
CR	: Compression Ratio
DCT	: Discrete Cosine Transform
DivX	: Digital Video Express
DV	: Digital Video
DVB-C	: Digital Video Broadcasting-Cable
DVB-S	: Digital Video Broadcasting-Satellite
DVB-T	: Digital Video Broadcasting-Terrestrial
DVD	: Digital Versatile Disc
DWT	: Discrete Wavelet Transform
EZW	: Embedded Zerotree Wavelet
F	: Frame
FPS	: Frame-per-second
HD	: High Definition
HF	: High Frequency
HVS	: Human Visual System
ISDN	: Integrated Services Digital Network
ISO/IEC	: Intern. Standards for Organization / Intern. Electrotechnical Commission
ITU	: International Telecommunication Union
KLT	: Karhunen–Loève Transform
LB	: Lifting Based
LF	: Low Frequency
LOT	: Lapped Orthogonal Transform
MC	: Motion Compensation
MCTF	: Motion Compensated temporal filtering
ME	: Motion Estimation

MPEG	: Moving Picture Experts Group
MP3	: MPEG-1 Audio Layer III
MV	: Moving Vector
OKH	: MSE, Mean Squared Error
P	: Predicted Frame
PSNR	: Peak Signal to Noise Ratio
QCIF	: Quarter Common Intermediate Form
R	: Residual Frame
ReC	: Reconstructed Frame
RGB	: Red-Green-Blue
SPIHT	: Set Partitioning In Hierarchical Trees
SQCIF	: Sub Quarter Common Intermediate Form
VHS	: Video Home System
VQ	: Vector Quantization
YUV	: Color space Y-Luminance, UV-Chrominance
1B	: 1 Boyut
2B	: 2 Boyut
3B	: 3 Boyut
3D-DWT	: 3 Dimensional-Discrete Wavelet Transform
3G	: 3rd Generation
4CIF	: 4 x Common Intermediate Form
16CIF	: 16 x Common Intermediate Form

TERİMLER DİZİNİ

Alçak-geçiren	: Low-pass
Alt bant	: Subband
Analiz	: Analysis
Arka plan	: Background
Aşağı örnekleme	: Down sampling
Ayrık Dalgacık Dönüşümü	: Discrete Wavelet Transform
Ayrık Kosinüs Dönüşümü	: Discrete Cosinus Transform
Ayrışım	: Decomposition
Aydınlatma	: Illumination
Bağlı	: Relative
Belirtim	: Specification
Birim dik	: Orthonormal
Bit düzlemi	: Bit plane
Blok	: Block
Bozulum oranı	: Distortion rate
Çekirdek	: Kernel
Çerçeve	: Frame
Çerçeveler arası	: Interframe
Çift-dik	: Bi-Orthogonal
Çözüğü	: Warping
Dalgacık	: Wavelet
Dekorelasyon	: Decorrelation
Dik	: Orthogonal
Dönel	: Rotational
Dönüşüm alanı	: Transformed Domain
Fark	: Residual
Fazlalık	: Redundancy
Frekans konumlama	: Frequency Localization
Genlik	: Magnitude
Genişleme ve daralma	: Expansion and contraction

Gerçek-zamanlı	: Real-time
Geri çatma	: Reconstruction
Gölgeleme	: Ghosting
Gömülü Zerotree Dalgacık Algoritması	: Embedded Zerotree Wavelet Algorithm (EZW)
Görüntü hatası	: Artifact
Frekans duyarlılığı	: Frequency sensitivity
Hareket Dengeleme	: Motion Compensation
Hareket Dengelemeli Zamansal Filtreleme	: Motion Compensated Temporal Filtering
Hibrit	: Hybrid
Hiyerarşik Ağaçlarda Küme Bölüntüleme	: Set Partitioning In Hierarchical Trees (SPIHT)
İstatistiksel fazlalık	: Statistical redundancy
İzgesel	: Spectral
İzgesel fazlalık	: Spectral redundancy
Kaba	: Coarse
Katsayı	: Coefficient
Keskin kenar	: Sharp edge
Lokal olarak düzgün	: Locally smooth
Mevcut	: Current
Nicemleme	: Quantization
Ortalama Kare Hata	: Mean Squared Error
Öngörü	: Prediction
Önplan	: Foreground
Özyinelemeli	: Recursive
Parlaklık kontrast duyarlılığı	: Luminance contrast sensitivity
Piksel	: Pel, Pixel
Renk	: Chrominance
Salt uzamsal kodlama	: Pure spatial encoding
Sentez	: Synthesis
Sıkıştırma	: Compression, Compaction
Sıkıştırma şeması	: Compression scheme
Uyarlanır	: Adaptive
Uzamsal	: Spatial

Uzamsal fazlalık	: Spatial redundancy
Tersinir	: Reversible
Yerel Blok Çözücü	: Local Block Warping
Yukarı Örnekleme	: Up-Sampling
Yumuşak	: Smooth
Yüksek geçiren	: High-pass
Yükseltme	: Lifting
Zamansal	: Temporal
Zamansal fazlalık	: Temporal redundancy

1. GİRİŞ

Video sıkıştırma yöntemlerini kullanan ticari ürünlerin ilk olarak ortaya çıkmasından bu yana video sıkıştırma algoritmalarının gelişiminde olağanüstü ilerlemeler yaşanmıştır. Ancak halen günümüz teknolojisinde mobil aygıtlar üzerinde çalışan gerçek zamanlı ve hızlı uygulamalara ihtiyaç duyulmaktadır.

Bu tezde, sayısal video sinyallerinin hareket dengeleme (MC = Motion Compensated) ve yükseltme tabanlı (LB = Lifting Based) hibrit bir kodlayıcı tasarlanarak, görüntü kalitesinden fark edilir düzeyde ödün vermeden sıkıştırma oranının artırılması hedeflenmiştir.

Video verisindeki bilgi fazlalığından ve insan görme sisteminin yapısı sebebiyle video sıkıştırma işlemi genellikle kayıplı bir biçimde gerçekleştirilmektedir [1]. MPEG tipi video kodlayıcılarda, videodaki bazı çerçeveler içsel tip çerçeveler olarak alınır ve diğer çerçeveler intra çerçevelerden öngörülme çalışılır. Bu kodlayıcının çıktısında, hem içsel çerçeveler hem de öngörü çerçevelerinin orijinallerinden farkları nicemlenir ve istenen sıkıştırma oranına göre kodlanır.

Bu tezde [2] ve [3]'te önerilen iki boyutlu (2B) yükseltme yöntemindeki kenar uyarılama stratejisi kullanılarak zamansal ayrıştırma ile uzamsal bilginin birleştirildiği üçüncü bir ayrıştırma yöntemi önerilmiştir. Çalışmada, üç boyutlu (3B) yükseltme tabanlı ayrıştırma başlangıç noktası olarak ele alınmış ve blok tabanlı hareket dengeleme yöntemi, zamansal yükseltme algoritmasının öngörü adımının hemen öncesine dâhil edilerek bir iyileşme aranmıştır. Bu çalışmanın ana motivasyonu, hem hareket dengelemenin hem de zamansal yükseltme öngörü işleminin öngörülen çerçevelerdeki sinyal varyansını en küçükmeye çalışmaktır. İki yöntemin birlikte kullanımının deneysel çalışmalar yoluyla doğrulanması gerekir. Zamansal yükseltme yönteminin doğrudan uygulanması ardışık çerçevelerde var olan hareketi hesaba katmayacağından tek başına arzulan verimliliği sağlamayabilir. Hareket dengeleme gibi uyarılma yöntemleri bu durumda yükseltme şemasıyla birlikte uygulanabilirler.

1.1. Video Sıkıştırma Yöntemleri ve Standartları

Temel olarak video sıkıştırma, video akış girdisinin analiz edilip izleyicinin algılayamayacağı bilgilerin atılıp kalan verinin entropi kodlanmasıyla gerçekleştirilir. Video çerçeveleri arasında ilintisi fazla olan durumlar daha az bitle, ilintisi daha az olanlar ise daha fazla bitle kodlanmaktadır. Kodlamada sırasıyla, sinyal dönüşümü, nicemleme ve değişken uzunluklu kodlama kullanılır.

Sinyal dönüşümü yöntemlerinden en popüler olan Ayrık Kosinüs Dönüşümü (DCT), uzamsal genlik verilerini uzamsal frekans verilerine dönüştüren kayıpsız ve tersi alınabilir matematiksel bir dönüşümdür. Blok dönüşümleri komşu imge pikselleri arasında yüksek ilintiyi kullanarak dönüşüm alanında enerji toplama veya kodlama kazancı elde etmektedir. DCT, birimlik bir dönüşüm olarak Karhunen–Loève dönüşümüne (KLT) yakın bir sıkıştırma performansı sağlamaktadır [4]. 8×8 boyutlu DCT dönüşüm matrisinin (C) katsayıları şu şekilde ifade edilmektedir:

$$C_{i,j} = \begin{cases} \sqrt{1/8} \cos \frac{(2j+1)i\pi}{16}, & i = 0, j = 0, 1, \dots, 7 \\ \sqrt{2/8} \cos \frac{(2j+1)i\pi}{16}, & i = 1, 2, \dots, 7, j = 0, 1, \dots, 7 \end{cases} \quad (1.1)$$

8×8 boyutu için verilmiş DCT katsayıları 16×16 ve 32×32 blok boyutları için de oluşturulabilir. DCT; JPEG, MPEG, H.261 ve H.263 gibi standartlarda 8×8 imge blokları üzerinde başarılı sonuçlar vermektedir.

Bir diğer dönüşüm türü olan Ayrık Dalgacık Dönüşümü (DWT) tabanlı sıkıştırma algoritması, bir yandan DCT gibi bir imgeyi frekans bileşenlerine matematiksel olarak ayrıştırırken, diğer yandan imgenin zamansal içeriğini de korur. Diğer yöntemlerin aksine, verinin küçük parçalarıyla çalışmak yerine süreç tüm imge üzerine filtreleme ile yapılır. Sonuç, her katmanın bir frekans bandındaki zamansal değişimine karşılık geldiği, görüntünün hiyerarşik bir temsildir.

Sıkıştırılmış video biçemlerinin mevcut standardizasyonu yirmi yıl kadar sürmüş bir çalışmanın sonucunda belirlenebilmiştir. Uluslararası organizasyonlar

tarafından sayısal video-ses ve görüntüleri üzerinde yukarıda bazıları sayılan çok sayıda standart tanımlanmıştır [5-8].

Uluslararası haberleşme birimine bağlı (ITU) Uluslararası Standartlar Örgütü (ISO) 1988 yılında Hareketli Resimler Uzmanları Grubu'nu (MPEG) kurmuştur. 1990 yılında sayısal televizyon yayını göz önüne alınmadan, çoklu-ortam kaynaklarının sıkıştırılmış sayısal sunumunu ele almak amacıyla 1.5Mbps sabit bit hızı ve sürekli tarama kullanan MPEG-1 standardı ortaya atılmıştır ve 1992 yılında resmi olarak kabul edilmiştir. MPEG-1 formatı kullanılarak 1.2Mbps ile sıkıştırılan ve açılan Ortak Arabirim Formatındaki (CIF) görüntü kalitesinin, Video Ev Sistemi (VHS) formatında kaydedilen analog videonun kalitesi ile aynı (veya VHS'den daha iyi) olduğu görülmüştür.

MPEG-1, MPEG grubunun ilk standardı olup CD-ROM'lar üzerinde video depolamak için geliştirilmiştir. Bu standardın temel bileşenleri, ses sıkıştırma yöntemleri (MP3), blok temelli hareket dengelemesi (MC), ayırık kosinüs dönüşümü (DCT) gibi video kodlama metotlarıdır.

MPEG-2, MPEG-1'deki video kalitesini arttırmak için 1994 yılında önerilmiştir. DVD'lerde kullanılan sıkıştırma formatıdır. Video kodlamada bu standardın getirmiş olduğu yeniliklerden bazıları; kalite kontrolü için farklı profil yapılarının kullanılması, değişken bit oranının 2 Mbps - 20 Mbps arasında olması ve daha gelişmiş kodlama algoritmaları kullanılmasıdır. MPEG-2 formatı aynı zamanda Sayısal Video Yayınında (DVB-C, DVB-S, DVB-T) kullanılmaktadır [5].

MPEG grubu tarafından geliştirilen diğer bir video kodlama standardı ise MPEG-4 video kodlama tekniğidir. Bu format 'ISO/IEC 14496' ismi altında 1999 yılında uluslar arası standart olarak kabul edilmiştir. MPEG-4 video sıkıştırma formatının getirmiş olduğu en büyük avantaj MPEG-1 ve MPEG-2 ile aynı bit oranında 2 kat daha fazla nesnel video başarımı sağlamasıdır. Örneğin, VHS kalitesinde MPEG-1 ile 1,2 Mbps bit hızında kodlanan bir video dizisi, aynı nesnel video kalitesinde MPEG-4 ile 500-700 Mbps bit hızında kodlanabilmektedir. Son nesil video kodlama standartlarından olan MPEG-4 içerik temelli kodlama tekniği kullanılmaktadır. Bu kodlama tekniğinde video

çerçevelerindeki her nesneye ilişkin doku ve hareket bilgisi ayrı bir katmanda kodlanmaktadır.

ITU tarafından standart olarak kabul edilen H.261 video sıkıştırma formatı Tümüleşik Hizmetler Sayısal Ağı (ISDN) üzerinden video konferansı için tasarlanmıştır. ISDN üzerinden gerçekleştirilen bu sistem için kullanılan bant genişliği 64 Kbps ile 2 Mbps arasındadır. H.261 standardında içsel (çerçeve içi) kip seçiminde ve aynı zamanda diğer çerçeveler için referans olarak kullanılan içsel *I* çerçevesi ile çerçeveler arası hareket tahminiyle öngörülen *P* çerçevesi kodlanmaktadır.

H.261'in sıkıştırma performansını artırmak için H.263 sıkıştırma formatı önerilmiştir. Bu standart H.261 standardı ile karşılaştırıldığında aynı kalitedeki resimler için hemen hemen yarı yarıya bit kazancı sağlamaktadır. H.263 video kodlama tekniğinin H.261'den farklı olarak kullandığı özelliklerden biri alt piksel hareket vektörlerini (MV = Moving Vector) kullanmasıdır. Diğer bir özellik ise *I* ve *P* çerçeveslerine ek olarak çift-yönlü hareket kestiriminde (Bidirectional Motion Estimation) kullanılan *B* çerçevesinin bu standartta kullanılmasıdır.

Daha yeni bir kodlama standardı olan H.264, resmi organizasyon olan ITU ve ISO standartlarının ortaklaşa geliştirdikleri bir kodlama yapısıdır. Bu yeni video kodlama standardı, video kodlama algoritmalarına birçok yeni özellik katarak, önceki video sıkıştırma platformlarından (MPEG-4, H.263 vb.) daha iyi bir sıkıştırma performansı sergilemektedir. Literatürde bu yeni standart için farklı isimler kullanılmaktadır. Bunlar;

- H.26L
- JVT (Joint Video Team) veya JVT CODEC (enCOder DECoder)
- JM2.x, JM3.x, JM4.x
- AVC (Advanced Video Codec)
- MPEG-4 Part 10 (resmi MPEG isimlendirmesi)
- H.264 (resmi ITU isimlendirmesi)

ISO/IEC ve ITU standartlarının yanı sıra DV ya da DivX gibi diğer endüstriyel standartlar da mevcuttur. Bunların çoğu (MPEG-1, MPEG-2, H.261, H.263, DV) DCT'yi bir sıkıştırma yöntemi olarak kullanırlar. Video sıkıştırma

teknikleri gibi geniş uygulama alanlarının dışında JPEG gibi hareketsiz görüntü sıkıştırma standartları da DCT kullanmaktadır. Araştırmacılar, DCT'nin getirdiği bir takım sorunların üstesinden gelmek için görüntü ve video uygulamaları alanında DWT üzerine çalışmalarını yoğunlaştırmaktadırlar [6].

1.2. DWT ile Video Kodlama Alanında Yapılmış Çalışmalar

3B Ayrık Dalgacık Dönüşümleri (3B-DWT) video sıkıştırma alanında birçok araştırmacı tarafından üzerinde çalışılmış bir konudur. Karlsson ve Vetterli video sıkıştırmada ayrılabilir 3B-DWT yöntemini kullanan ilk araştırmacılar olmuşlardır [7]. Taubman ve Zakhor'un ayrılabilir 3B-DWT uygulamasından önce yüksek çözünürlüklü ağlarda uzamsal video çerçevelerinin uyumu konusu üzerine bir yaklaşımı olmuştur [8].

Ohm [9], Woods ve Choi [10] tarafından önerilen diğer bir yaklaşım yerel blok çözücü veya blok yer değiştirme yöntemleri olarak tanımlanabilir. Bu yaklaşım türünde, video çerçeveleri öteleme dönüşümünden oluşan katı harekete maruz kaldığı yerlerde bloklara ayrılır. 3B-DWT dönüşümü ayrılabilir bir biçimde yer değiştirilmiş bloklara tekrar uygulanabilir. Fakat burada bloklar arasındaki “bağlantısız pikseller” den kaynaklanan hareket alanındaki genişleme ve daralmalar gözlemlenir. Bu yaklaşımın bir sınırlaması blok tabanlı hareket modellerine uygulanmasının kısıtlanmış olmasıdır [11]. Zamansal dönüşüm ve hareket modeli arasındaki sıkı bağlantı, zamansal yönde, Haar dışındaki dalgacık çekirdeklerinin kullanılmasını güçleştirmektedir.

Yukarıda bahsedilen kısıtlamalara karşılık olarak Secker ve Taubman [11] tarafından çerçeveler arası bloklar üzerinde yükseltme üzerine hareket dengelemeli zamansal DWT tabanlı bir çalışma yapılmıştır. Geliştirilen bu yöntemin hareket dengeleme gerektiren hareket modelli dönüşümlerde başarımı yüksek olup, zamansal ölçeklenebilirlik için yüksek kalitede düşük zamansal çözünürlüklü çerçeveler üretmektedir.

Son zamanlarda yapılan çalışmalarda, özellikle video sinyallerinin alt-bant zamansal ayrıştırılmasında Pesquet-Popescu ve ark. [16-18] tarafından hareket dengeleme ile sıkıştırma verimini arttıran bir yükseltme şeması geliştirilmiştir. Bu çalışmalarda, uyarlanabilir alt-bant ve doğrusal alt-bant ayrıştırılmaları kullanılarak

klasik hareket dengelemeli Haar zamansal filtrelemeye karşı başarılı gelişmeler elde edilmiştir.

Bu tezde önerilen kodlayıcıda, yukarıda söz edilen çalışmalarda olduğu gibi hibrit bir kodlayıcı tasarlanması hedeflenmiştir. İkinci bölümde hareket dengelemeli kodlama yöntemlerinden bahsedilecek ve üçüncü bölümde önerilen yöntem ayrıntılı olarak açıklanacak ve yöntemlerin güçlü ve zayıf yönleri deneysel sonuçlar ile birlikte tablolar halinde dördüncü bölümde gösterilecektir. Son bölümde ise tez çalışmasının sonuçları verilecektir.

2. İMGE VE VİDEO SIKIŞTIRMANI TEMELLERİ

2.1. İnsan Görme Sistemi ve Sıkıştırma

Bir imge/video sıkıştırıcının amacı; imge/video sinyalinin İnsan Görme Sistemi (HVS) tarafından algılanamayan tüm imge/video içerisinde tekrarlanan veri fazlalığının ortadan kaldırılmasıdır [6].

Sıkıştırma kullanılan yöntemle bağlı olarak tersi alınabilir. Geri çatılan imge/video'nun kalitesi HVS'nin algısı ile ilgilidir. Bu nedenle insanın görme sisteminin temel yapısını anlamak önemlidir. Parlaklık kontrast duyarlılığı ve frekans duyarlılığı HVS'nin kullandığı en önemli temel özellikler arasında yer alır. Uzamsal frekans arttıkça insan görme sistemi zayıflamaktadır [12]. Parlaklık üzerinde ya da renk ayarları üzerinde yapılan küçük bir değişiklik çoğu zaman görüntünün daha rahat algılanmasını sağlayabilir.

HVS tarafından algılanamayan imge/video verilerinde bulunan yüksek miktardaki fazlalıkların atılabilmesiyle imge/video verisinin sıkıştırması belirgin bir bozulma olmadan yapılabilmektedir. Kodlamada kullanılan fazlalıklar; renk bileşenleri arasındaki ilintiden kaynaklanan izgesel fazlalık, komşu pikseller arasındaki ilintiden kaynaklanan uzamsal fazlalık, çerçeve grupları arasındaki ilintiden kaynaklanan zamansal fazlalık ve bit düzlemindeki semboller arasındaki ilintiden kaynaklanan istatistiksel fazlalık olarak sıralanabilir.

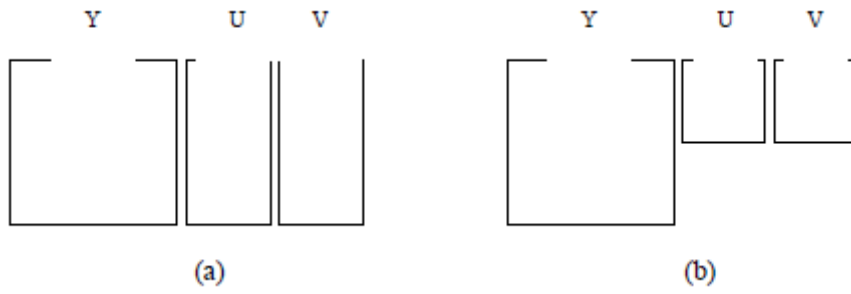
Aşağıdaki bölümlerde bu fazlalık türleri ve fazlalıkların kaldırılması işlemleri ayrıntılı bir biçimde açıklanmıştır.

2.2. İzgesel Fazlalık

Profesyonel bir video kamera çıktısı çoğunlukla RGB renk uzayında renklerden oluşan tam renkli bir video sinyalidir. İnsan gözü imge/videodaki parlaklık değerlerine, renk değerlerine göre daha duyarlı olduğundan, RGB renk uzayı izgesel fazlalık olarak değerlendirilebilir [6]. Bu yüzden sıkıştırmada ilk adım olarak renk uzayını YUV uzayına dönüşümü sağlanır. Y, parlaklık bileşenine karşılık gelirken, U ve V ise renk bileşenlerine karşılık gelmektedir.

YUV renk uzayı izgesel fazlalıkların giderilmesi için kullanışlı bir dönüşüm olarak kabul edilir [13]. Ek-1’de RGB'den YUV'ye dönüşümler hakkında ayrıntılı bilgi verilmiştir.

Sayısallaştırılmış video sinyalinde, kırmızı (R), yeşil (G) ve mavi (M) olmak üzere her piksel için 3 bileşen bulunmaktadır. Bu bileşenlerin her birisi genellikle 8 bit ile temsil edilirler. YUV renk uzayındaki U ve V bileşenleri fazlalık olarak kabul edilir ve bu sinyaller 2 ile aşağı örneklenir. 4:2:2 YUV dönüşüm sisteminde aşağı örnekleme yalnızca yatay ekseninde uygulanır. Bu uygulama sonucunda 3:2 oranında sıkıştırma sağlanır. 4:1:1 YUV dönüşüm sisteminde ise U ve V sinyalleri hem yatay hem de dikey doğrultuda aşağı örneklenir ve 2:1 oranında sıkıştırma oranı elde edilir [6]. YUV renk uzayı gösterimi Şekil 2.1’de gösterilmiştir.



Şekil 2.1. YUV Sistemi. (a) 4:2:2; (b) 4:1:1.

2.3. Uzamsal Fazlalık

İzgesel fazlalıktan farklı olarak imge/video sinyallerindeki uzamsal bilgi insan gözü tarafından algılanamamaktadır. Özellikle HVS’nin, uzamsal alanda düşük frekanslara daha duyarlı olduğu gözlenmiştir [14]. Bu nedenle sinyaldeki yüksek frekanslar görsel kalitede bozulma olmadan daha iyi sıkıştırılabilir. Böylece daha iyi bir enerji sıkıştırmanın mümkün olduğu diğer bir alana sinyalin dönüşümü gerçekleşmiş olur [12]. Dönüşüm sinyalleri dönüşüm katsayıları olarak ifade edilen sayılarla gösterimini dönüşüm alanında sağlar. Bu katsayılar genellikle, sinyalde kullanılan izgesel bileşenlerin analizinde sinyalin frekans

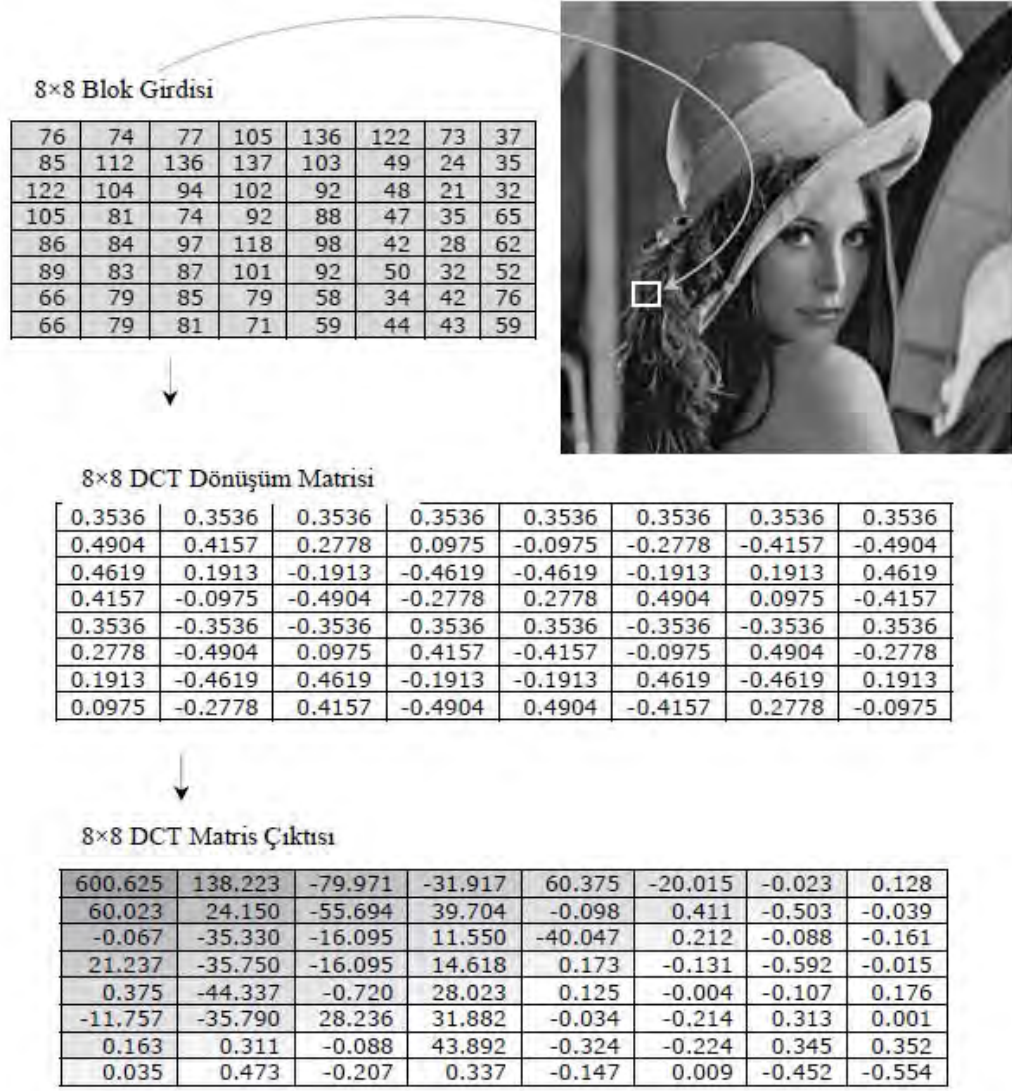
bileşeni hakkında bir fikir vermektedir. Bu nedenle, yüksek frekansların düşürülmesi aynı zamanda izgesel bileşenlere karşılık gelen dönüşüm katsayılarının düşürülmesi demek olacaktır [4].

Uzamsal fazlalıkların imge/video kodlamada atılabilmesinin başarımı tamamen farklı izgesel bileşenlerin katsayılarının ayırt edilmesine bağlıdır. Bu yolla dönüşüm tekniklerinin uygulanması bizlere frekans konumlanma bilgisini sağlamaktadır. Ayrıca katsayıların genliği o konumdaki frekans bileşeninin bağlı ağırlığı hakkında da bir fikir verecektir. Bu bilgi sayesinde, katsayıların uygun nicemlemesiyle verimli bir biçimde uzamsal fazlalığın atılması mümkün olabilir.

Ayrık Kosinüs Dönüşümü (DCT) ve Ayrık Dalgacık Dönüşümü (DWT) uzamsal fazlalığı atabilmek için kullanılan iki ana tekniktir [15]. DCT, bugüne kadar çoğu sıkıştırma şemasında yaygın olarak kullanılmıştır. DWT ise uygulamaları geliştirmekte olan daha yeni bir yöntemdir. Her iki dönüşümün de birbirlerine göre avantajları bulunmasına rağmen, her iki teknik de sinyalin enerjisini farklı frekanslarda konumlandırmayı hedeflemektedir. Günümüzde araştırmacılar imge/video kodlayıcılardaki DCT'nin yerine DWT'yi, izgesel alt bantlarda zaman konumlandırılması gibi dalgacık dönüşümünün eşsiz avantajlarından yararlanmak için kullanmaktadır [16].

2.3.1. Ayrık kosinüs dönüşümü (DCT)

JPEG gibi DCT tabanlı kodlama algoritmalarında imgeler 8×8 'lik bloklara ayrılarak işlenmektedir. DCT'nin girdisi, elemanları 0 ile 255 arasında değişen 8×8 'lik bir tamsayı matrisidir. Bloklara ayırma işlemi uygulamada kolaylık sağlayarak, tüm imgenin dönüşüm işlemlerinin sayısını azaltmaktadır. Dönüşüm blokla aynı ölçüdeki DCT matrisi ile yapılmaktadır. Şekil 2.2'de Lena imgesi üzerinde 8×8 blokluk tipik ayrık kosinüs dönüşümü ve işlemde kullanılan dönüşüm matrisi gösterilmektedir.

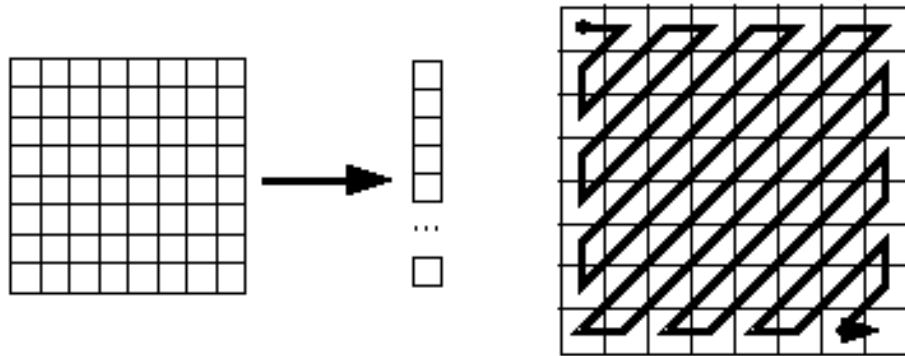


Şekil 2.2. “Lena” imgesi üzerinde 8x8 Blok DCT

Lena imgesi gibi çoğu imgede sinyal enerjileri düşük frekans üzerine yığılırlar. Alçak frekansa karşılık gelen dönüşüm katsayıları DCT matrisinin sol üst köşesinde görülür. Katsayıların genliği ne kadar yüksek olursa, karşılık gelen izgesel bileşenlerin enerjileri de o kadar yüksek olur. Bu nedenle sinyalin enerjisinin büyük çoğunluğu DCT matrisinin sol üst köşesindeki katsayılarda toplanmaktadır. Matrisin sağ alt köşesindeki değerleri yüksek frekans değerlerini gösterir ve diğer değerlere göre çok küçük olduğundan ihmal edilebilir. Bu ihmal, görüntü üzerinde kabul edilebilir küçük bir bozulmaya neden olmaktadır.

Sonuçta her 8×8 'lik blok için elde edilen DCT katsayıları JPEG algoritması ile kodlanırlar. Nicemleme matrisi nicemleme parametreleri ile taban matrisinden elde edilir. Bu parametreler imge/video kodlamayı ayarlamak için kullanılırlar. Büyük nicemleme parametreleri, kodlanmış imge ya da video çerçevesinde düşük kalitede daha küçük boyutta bitleri üretirler. Nicemlenmiş DCT katsayıları sonuç olarak daha kompakt bir temsil için değişken uzunlukta kodlanırlar. Buna karşın kod çözücü de bu işlemlerin tersini yapar.

Nicemlemede en önemli nokta DC katsayısı olarak bilinen matrisin satır ve sütunundaki en küçük indisli katsayısının (Şekil 2.2'deki 600.625 değeri) en az kayıpla kodlanmasıdır. AC katsayıları diye bilinen diğer katsayılar ise Şekil 2.3'te gösterildiği gibi kendi frekanslarıyla sırayla taranır. Burada amaç, 8×8 'lik DCT bloğunu 1×64 boyutunda bir vektöre dönüştürmek ve alçak frekansları bu vektörün en üstünde gruplamaktır. HVS yüksek frekanslara duyarlı olmadığı için bu frekanslar daha yüksek düzeyde bir nicemleme kullanılarak sıkıştırılabilir.



Şekil 2.3. Zig-Zag tarama ile DCT katsayılarının elde edilmesi

Uygulamada, DCT tabanlı kodlayıcılarda nicemleme seviyesi hedef bit oranı esas alınarak ayarlanır [6-8]. Hedef bit oranı küçüldükçe, yüksek frekanslar daha da düşürülür. Her bir blok bağımsız olarak nicemlenip kodlanırken blok sınırları düşük bit oranlarında tespit edilebilir. Bu durum DCT'nin önemli dezavantajlarından biridir. Blokların görüntü hatalarını gidermek için araştırmacılar Bindirilmiş Dik Dönüşümler (LOT) gibi farklı dönüşümler önermişlerdir [17].

DCT tabanlı görüntü sıkıştırma algoritmalarını daha iyi kavramak için 256×256 boyutundaki Lena imgesini ele alalım. Orijinal imge geri çatlarken daha az sayıda dönüşüm katsayısı kullanılabilir. İki farklı örnek şöyle olabilir:

Örnek-1: Her DCT bloğu için geri çatlama 2 DCT katsayısı kullanılması.

Örnek-2: Her DCT bloğu için geri çatlama 4 DCT katsayısı kullanılması.

Geri çatlama kullanılan katsayılar önem sırasına göre belirlenir. Katsayılar vektöründeki katsayıların çoğu ihmal edilirken, sırasıyla bunların geriye kalan % 6,25 ve %3,125'i kullanılmıştır. Denklem (2.1) ile geri çatlama imgelerinin kalite ölçümü PSNR değerleri ile hesaplanmıştır. Şekil 2.4'te ise geri çatlama imgeleri ve bunların PSNR değerleri verilmiştir.

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{\frac{\sum_{r,c} (I_{r,c} - \tilde{I}_{r,c})^2}{N \times M}}} \right) \quad (2.1)$$

$I_{r,c}$ ve $\tilde{I}_{r,c}$ sırasıyla r -sıra, c -sütun konumundaki orijinal ve geri çatlama imgeleri göstermektedir. İmgenin piksel olarak N sıra (yükseklik) ve M sütundan (genişlik) oluştuğunu varsayalım.



Şekil 2.4. "Lena" imgesinin (a) 4 katsayı (%6,25), 30,62 dB; (b) 2 katsayı (%3,125), 26,97 dB ile geri çatılması.

Şekil 2.4'te bloklar üzerinde görüntü hataları olduğu görülmektedir. Geri çatılmış (a) resmindeki hatalar (b)'ye göre insan gözü tarafından daha az fark edilmektedir.

2.3.2. Ayrık dalgacık dönüşümü (DWT)

DWT, imgelere 1B DWT olarak önce satır doğrultusunda sonra sütun doğrultusunda uygulanarak iki aşamada gerçekleştirilir. Ayrıntılı bilgi Ek-2'de verilmiştir. Satırlar ve sütunlar üzerinde aşağı örnekleme sonucunda Şekil 2.5-b'de olduğu gibi 4 resim oluşur. Bu algoritma blok temelli DCT ayrıştırmasına göre tüm imge üzerinde yapıldığından farklıdır. Ayrıca alçak frekanslı kısmın (LL alt bandı) dalgacık dönüşümü Şekil 2.6'da gösterildiği gibi daha fazla bitle kodlanarak daha az kayıplı kodlanması sağlanır. Orijinal imge, ters dönüşüm 3 kere alınarak yine de tam olarak elde edilebilir.

Alçak frekans bandı (L)	Yüksek frekans bandı (H)
-------------------------------	--------------------------------

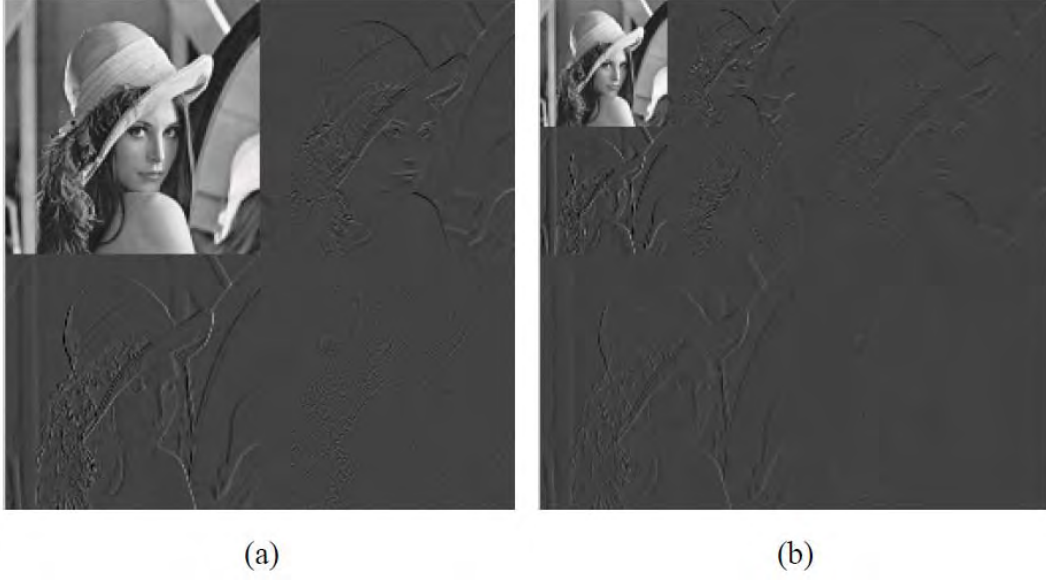
Alçak-Alçak frekans bandı (LL)	Yüksek-Alçak frekans bandı (HL)
Alçak-Yüksek frekans bandı (LH)	Yüksek-Yüksek frekans bandı (HH)

Şekil 2.5. İmge üzerinde 1-Düzye 2B DWT: (a) Satır doğrultusunda DWT; (b) Sütun doğrultusunda DWT

İmge, 1-Düzye ayrışım sonrasında LL, LH, HL ve HH olmak üzere 4 frekans bandına ayrılır. Ayrışım işlemi devam ettirilerek sırasıyla 2-Düzye ve 3-Düzye ayrışımına karşılık gelen frekans bantları elde edilir. N-düzye ayrışım için $3N + 1$ farklı frekans bandı elde edilmektedir. Bunlardan $3N$ tanesi yüksek frekans bandı iken yalnızca bir tanesi LL frekans bandı olacaktır. Şekil 2.6' da 3-Düzye'e kadar olan DWT ayrışım işlemi gösterilmiştir.

LL³	LH³	LH²	LH¹
HL³	HH³		
HL²		HH²	
HL¹		HH¹	

Şekil 2.6. 2B DWT'nin Piramidal hiyerarşisi.



Şekil 2.7. "Lena" imgesi üzerinde DWT (a) 1-Düzye Ayrışım; (b) 2-Düzye Ayrışım.

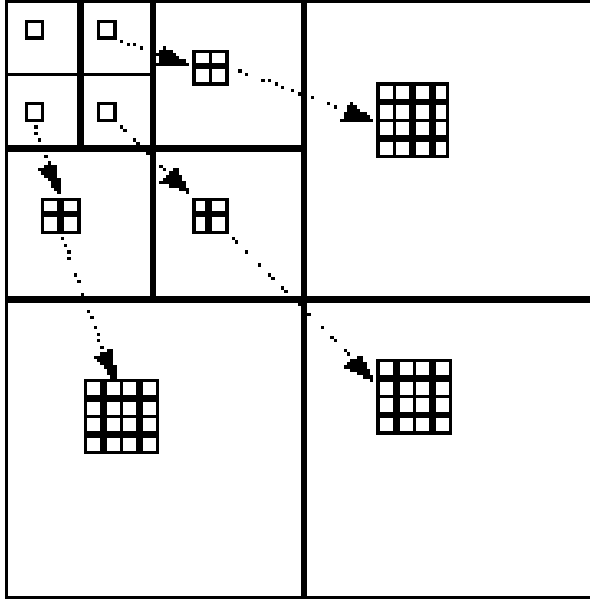
Şekil 2.7.(a)'da Lena imgesinin 1-düzye ayrışımı gösterilmiştir. 2B dalgacık dönüşümü, şekilde görüldüğü gibi Lena imgesini dört frekans bandına ayırmıştır. Bunlardan her biri orijinal imgenin boyutunun çeyreği kadardır. Sol üstte yer alan en düşük çözünürlük bandı olan LL aynı zamanda sinyal enerjilerinin çoğunun bulunduğu alçak frekanslı bölüm karşılık gelmektedir. Diğer tüm alt bantlar detay bilgilerini yüksek frekanslarda tutmaktadır. Yüksek frekans bantları kolay sıkıştırılabilen alçak genliklere sahiptir. İnsan gözü yüksek frekanslı detay bilgilere hassas olmadığından yüksek frekansa sahip diğer bölümler atılabilir. Yüksek frekanslı bilgiler olmadan da imge geri çatıldığında orijinaline yakın bir görüntü verecektir.

Ayrık dalgacık dönüşümü LL alt imgesine özyinelemeli olarak uygulandığında Şekil 2.7.(b)'de olduğu gibi imgenin bir alt düzeydeki ayrışımı elde edilir. Bu şekilde imgenin hiyerarşik bir ayrışımı elde edilir. Ayrışım düzeyinin sayısı orijinal imgenin boyutuna bağlıdır. Bazı durumlarda tek bir katsayı ile tüm imgenin temsili mümkün olabilmektedir.

Sıkıştırma işlemini tamamlamak için, nicemleyici ve entropi kodlayıcının dalgacık dönüşümüne göre tasarlanması gerekecektir. Gelişmiş bir dalgacık kodlama algoritması olan "Zerotree kodlaması" çoğu dalgacık tabanlı imgelerin

kodlama şemasında kullanılmaktadır. İmge, Şekil 2.8’de olduğu gibi ağaç yapısında gösterilmektedir. Kök düğüm, burada üç düğümün üstünde bulunmakta olup, en düşük frekans bandındaki ölçekleme fonksiyonu katsayısını temsil etmektedir. Ağaçtaki her bir düğüm, bulunduğu yerin yüksekliğini ölçek alan dalgacık katsayılarına karşılık gelmektedir. Bu katsayıların her biri, sonraki ölçekte aynı konuma sahip dalgacıklara karşılık gelen dört katsayıya sahiptir. Veri yapısının altında bulunan alt düğümü olmayan yaprak düğümler, önceden belirlenen eşik değerine göre önemsiz düğüm değeri olarak yok sayılmaktadır. Bu yok sayılan katsayılar kod çözücü tarafından sıfır (zero) olarak çözülmektedir. Eğer düğüm değeri önemli ise, her dört alt düğümü ile birlikte yeni bir veri ağacının kökü kabul edilir. Bu süreç tüm ağaçlar elde edilene kadar devam edilir [18].

“Gömülü Zerotree Dalgacık algoritması” (EZW) “zerotrees” yöntemini kullanan ilk imge kodlayıcıdır [19]. Daha sonra Hiyerarşik Ağaçlarda Küme Bölüntülemeli (SPIHT) algoritması, katsayılardan oluşan kümeler listesini zerotree kavramı üzerine inşa ederek oluşturulmuştur. EZW ve SPIHT algoritmalarının her ikisi de 2B imgeler üzerine geliştirilmiştir. EZW algoritmasının 3B’ye genişletilmesi daha sonraları ele alınmıştır [20] [21]. EZW’nin kodlama performansı SPIHT ve SPIHT’tan türetilmiş farklı kodlayıcılarla iyileştirilmiştir [18].



Şekil 2.8. Zerotree kodlama yapısı

2.3.3. DCT ile DWT'nin karşılaştırılması

DCT'de kontrol için orijinal sinyal ile sabit temel kosinüs fonksiyonu karşılaştırılırken, dalgacık dönüşümünde dalgacıklar ile orijinal sinyaller arasında kontrol edilmektedir. Bu durum ise, DWT tabanlı şemalara avantajlar sağlamaktadır. Bir başka deyişle temel fonksiyonlar seçilen imgenin durumuna göre (yumuşak, keskin kenarlı v.b.) değiştirilebilmektedir.

Doğal olarak, DCT tabanlı imgelerde kodlayıcının imgeleri 8×8 boyutundaki bloklara bölmelerinden kaynaklanan rahatsız edici ve göze batan görüntü hataları bulunmaktadır. Bu durum DWT ile yapılan testlerde geri çatılan imgelerde gözlenmemiştir. İlâveten PSNR değerleri olarak da DCT kadar iyi değerler elde edilmiştir. İki dönüşüm PSNR değerleriyle de karşılaştırılmış ve 30 dB değerine sahip DCT ile DWT'nin 2-Düzyey ayrışımının birbirine yakın değerler aldığı görülmüştür [12].

Bölüm 2.3.'den buraya kadar imge/video üzerindeki uzamsal fazlalığı atma teknikleri verilmiştir. Tezde, DWT nicemleme düzeyinin kolaylıkla belirlenebildiği SPIHT kodlama üzerine yoğunlaşmıştır.

2.4. Zamansal Fazlalık

Zamansal fazlalığın kaldırılması, video sıkıştırma algoritmaları ile ilişkili bir konudur. Video sinyalleri, saniyede 25 ya da 30 çerçeve ile örneklenmektedir. Bu durumda video dizisindeki ardışık çerçeveler hemen hemen birbirinin aynısı gibi olmaktadır. Özellikle, görüntüdeki kamera hareketleri ve aydınlatma değişikliklerinin çok az olduğu durumlarda arka plan bilgileri aynı olacaktır. Bu durum, çerçeveler arası zamansal fazlalığa neden olmaktadır.

Zamansal fazlalığı kaldırmak amacıyla belki de en kolay yaklaşım; kamera hareketinin olmadığını varsayarak, mevcut çerçeveden önceki çerçeveyi çıkartmak yoluyla aktarılan bilginin azaltılmasıdır. Diğer bir algoritma olarak, video dizisindeki çerçeveleri bloklara ayırarak, her bir bloğun hareket bilgisinin mevcut çerçevedeki yerinin önceki çerçevedeki yerinden yola çıkarak tahmin edilmeye çalışılmasıdır. Bu hareket bilgisi (=hareket vektörü) gönderilerek, zamansal fazlalık azaltılırken aynı zamanda kamera hareketleri ve nesnelerin video dizisindeki hareketleri dikkate alınabilmektedir. Bu işleme hareket dengeleme denir. Ayrıntılı olarak Bölüm 3.1'de anlatılacak olan bu işlem, kısaca hareketin tahmininden sonra mevcut çerçevenin bu hareket bilgisinin kullanılarak önceki çerçeveden elde edilmesi ile gerçekleşir. Bu yöntemle, kod çözücüyü bütün imge yerine bloklar arasındaki farkın gönderilmesiyle bit oranında azalma sağlanır.

2.5. İstatistiksel Fazlalık

Önceki bölümlerde belirtilen fazlalık türlerine ek olarak, sıkıştırılmış sinyallerde istatistiksel fazlalıklar da bulunabilmektedir. Video sinyallerinin istatistiksel özellikleri bu tür fazlalıklara neden olabilmektedir. Bu fazlalıkları atma amacıyla, *Run-length*, *Huffman*, veya aritmetik kodlama teknikleri gibi bazı istatistiksel kodlama şemaları kullanılmaktadır [6].

3. HAREKET DENGEMELİ VE LİFTİNG TABANLI VIDEO KODLAYICI TASARIMI

3.1. Hareket Dengeleme ve Hareket Kestirimi

Hareket dengeleme (MC) [22] 1960'lı yıllarda video kodlayıcıların verimliliğini arttırmak için kullanılmıştır. Hareket dengelemeli video kodlayıcılar üç kademeli olarak uygulanmaktadır. İlk aşamada önceden geri çatılmış çerçeve ile mevcut çerçeve arasındaki hareketin tahmini (ME = Hareket Kestirimi) yapılır. İkinci aşamada ise mevcut çerçeve öngörüsünün, önceden geri çatılmış çerçeve ve hareket tahmini kullanılarak oluşturulması sağlanır. Son aşamada, öngörü ile mevcut çerçeve arasındaki farkın öngörü farkı olarak kodlanması işlemi yapılır. Bu nedenle, alıcı mevcut imgeyi yalnızca değişken uzunluklu kodlayıcının (VLC) uzamsal hareket öngörülerini ve bu öngörülerin kodlanması sırasında oluşan öngörü farklarını kullanarak geri çatar.

Hareket kestirimi ve hareket dengelemesi (ME/MC) video sinyalini zamansal doğrultuda kodlamak için kullanılan ortak yöntemlerdir. Bu yöntemlerde çeşitli boyutlarda bloklardan yararlanılmaktadır. Blok dönüşümleri, uzamsal bilginin nicemlemeye ve VLC kodlamaya olanak sağlayan önemli bir teknik olarak kullanılmaktadır. Blok dönüşümleri, blok enerjilerini az sayıdaki katsayı ile karşılayarak kodlamadan kazanç elde edilmesine olanak sağlar. Video kodlamanın nicemleme aşaması bir video sıkıştırma sisteminin bozulma oranı karakteristiğinin belirlenmesinde merkezi bir faktördür. Blok dönüşüm katsayıları, bit hızı ve bozulma belirtimine bağlı olarak nicemlenir. Hareket dengeleme, zamansal düzlemdeki belirgin bilgi fazlalığının getirdiği avantajı kullanır ve önceki çerçeveyle mevcut çerçeve arasında blokların hareketini eşleştirerek mevcut çerçeve için öngörüler yapar. Hareket dengeleme çoğunlukla uzamsal kodlama üzerinde video kodlama verimliliğini belirgin bir biçimde arttırmaktadır.

Video sıkıştırma sistemlerinde kullanılan blok tabanlı hareket dengeleme ve hareket kestirim teknikleri ham sinyalin bit oranındaki en büyük azalmayı sağlamaktadır. Tipik uygulamalar tamamıyla uzamsal kodlamalardan en az üç kat daha iyi performans gösterirler [15]. Sayısal bir imge dizisinin zamansal boyutta

yer alan çerçeveler-arası fark, video kodlayıcılara etkili bir sinyal sıkıştırma olanağı sunar. Çerçeveler-arası fark, potansiyel zamansal sıkıştırmayı gerçeklemek için, statik arka-planlar ve hareketli ön-planlar gösterimi ile basitçe modellenenbilir. İmge dizileri kısa bir süre içerisinde, ön-planda hareketli nesnelere birlikte statik bir arka-plan olarak tanımlanabilir. Dizi içerisindeki ardışık iki çerçeve arasındaki arka-plan değişmiyorsa farkları sıfır olarak alınır ve bu iki çerçeve tek bir çerçeve olarak kodlanabilir. Bundan dolayı sıkıştırma oranının artışı, ilk çerçevede elde edilen uzamsal sıkıştırmanın iki katıyla orantılıdır [15]. Genel olarak, değişmeyen veya statik arka-planlar kodlama kazancına katkı sağlamaktadır [15].

$$\begin{aligned} & \text{Statik arka – plan kodlama kazancı} \\ & \propto \\ & N \times (\text{Arka – plan çerçevedeki Uzamsal Sıkıştırma Oranı}) \end{aligned} \tag{3.1}$$

Burada N kodlanmış statik arka-plan çerçeve sayısını ifade etmektedir. Video imgeleri üzerinde statik arka-planlar büyük yer işgal ederler. Rastsal ve sistematik nedenlerden dolayı meydana gelen dalgalanmalar yüzünden arka-planda daima bazı değişimler meydana gelir. Bu, elde edilebilecek arka plan kodlama kazancını azaltma eğilimindedir [15].

Hareketli ön-planlar, arka-plandan bağımsız hareket eden dönel olmayan katı nesnelere olarak modellenmiştir. Hareketli nesnelere iki çerçeve arasındaki ön-planda olan nesnelere eşlenmesi ile tespit edilebilir. Eşleme sırasında iki çerçeve arasındaki farkın sıfır olması mükemmel sonucu verecektir. Teoride, hareketli ön-planlar kodlama kazancına katkı sağladığı öngörülmüştür [23]. Pratikte elde edilebilen kodlama kazancını azaltabilen nesne hareketleri mevcuttur. Bu nesne hareketlerine örnek olarak, nesnenin esnek hareketini (genişleme ve büzülme), nesnenin kendi eksenine etrafında dönmesini, nesnenin kendisinin ya da arka-plandaki ışığının değişmesini sayabiliriz. Hareket kestirim yöntemlerini kullanan hareket dengelemeli sistemler hem arka-plan hem de ön-plan kodlama kazancından yararlanır. Arka planlar statik olduklarında, yani hesaplanan hareket vektörü (0,0) olduğu durumda, çerçeveler arasında farka dayalı kodlama sağlarlar

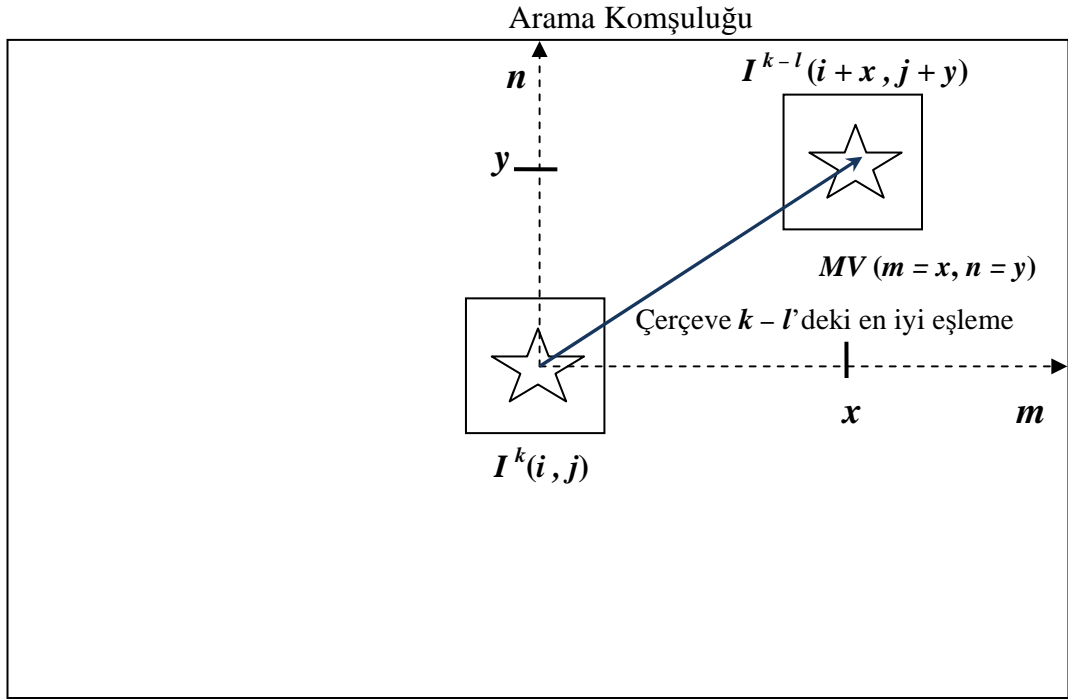
ve hareketli ön-planlar ele alındığında hesaplanan hareket kestirimi, en küçük bozulum öngörüsünü üretir.

Hareket kestirimi, pel-özyinelemeli (pel-recursive = Picture element-recursive) algoritmalar [24] ve blok-eşleştirmeli algoritmalar (BMA) [25] olarak iki genel kategoriye giren çerçeveler arası bir öngörü sürecidir. Pel-özyinelemeli yöntemler çok karmaşık oldukları için video kodlayıcılarda sınırlı durumlarda kullanılmıştır [26]. Sayısal imge dizileri, pel-özyinelemeli algoritmaların yakınsama özellikleri nedeniyle videoda bazı nesnelerin hareketlerini belirsiz bir şekilde göstermektedir. Ortaya çıkan bu resim dizilerini kodlamak için blok-eşleştirmeli hareket kestirimi yöntemlerinin geliştirilmesine neden olmuştur. Blok-eşleştirmeli hareket kestirimi imgelerdeki hareketli nesnelerin dönel olmadığı ve esnek olmadığını kabul ederek hareket eder. Aksi durumda eşleştirilecek bloklarda farklılıklar olacağından yöntemin başarımında azalmalar olacaktır. MPEG, H261 ve H263 kodlayıcılarda BMA için blok boyutu 16×16 piksel olarak belirlenmiştir. Ayrıca MPEG-2, 16×8 piksel blokları da desteklemektedir.

BMA'lar, imge dizilerindeki 2 çerçeve arasındaki piksel bloklarının hareketini öngörür. Öngörü, arama komşuluğu sınırlandırılmış bir alanda pikselin yer değişimini bulmaya çalışmaktadır. Arama komşuluğunun boyutu algoritmanın karmaşıklık derecesini belirler. Arama komşuluğu içerisinde en iyi blok eşleştirmesi elde edildiğinde en iyi öngörü yakalandığından aramada sona ermiş olur. En iyi eşleştirme, tam bir aramada her bloğun arama komşuluğunda en küçük Ortalama Kare Hatası (OKH) seçilerek bulunabilir.

$$\text{En iyi Eşleştirme}_{\text{OKH}} = \min_{m,n} \frac{1}{N^2} \sum_{i=1}^M \sum_{j=1}^N [I^k(i,j) - I^{k-l}(i+m,j+n)]^2 \quad (3.2)$$

Burada k çerçeve indisini, l çerçevelerdeki zamansal değişimi, M ve N imge bloğunun sırasıyla yatay ve dikey eksenlerindeki piksel sayılarını, i ve j imge bloğundaki piksel indislerini, m ve n ise yatay ve dikey doğrultudaki arama komşuluğu indislerini göstermektedir. Buna bağlı olarak, en iyi eşleştirme hareket vektörü tahmini, yani $MV(m = x, n = y)$, k indisli çerçevedeki $I^k(i,j)$ bloğuyla $k-l$ indisli ötelenmiş çerçevedeki en iyi eşleşen $I^{k-l}(i+x,j+y)$ bloğu arasındaki piksel farkıdır. En iyi eşleşme, Şekil 2.1'de resmedilmiştir.



Şekil 3.1. En iyi eşleme hareket tahmini

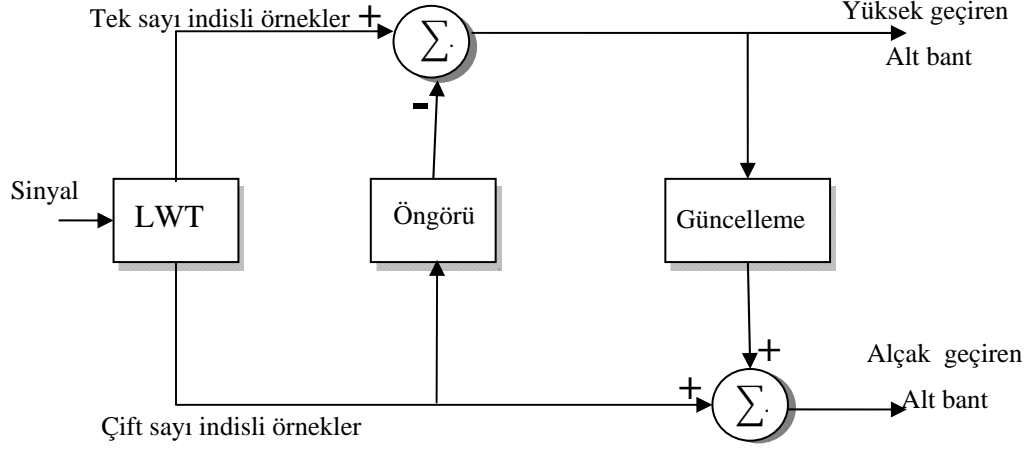
Sahne değişimi ya da düzgün olmayan blok hareketi durumlarında, bloğun uzamsal kodlamasına bağlı olarak hareket tahmininin bit oranı değeri artacaktır. Hareket tahminin etkili olmadığı durumda, video kodlayıcı hareket kestirimi yerine bloğu uzamsal kodlama kullanarak kodlamaktadır.

Arama alanının boyutu, hareket kestirimi algoritmasının karmaşıklığında etkilidir. Gerçek-zamanlı video kodlayıcılarda, tam bölge arama yöntemleri işlemci ve zaman maliyetlerinin yüksekliği nedeniyle tercih edilmemektedir. Hızlı arama teknikleri, görüntü bütünlüğünü koruyarak hesaplamadaki karışıklığı azaltmaktadırlar. Bu tekniklerde kullanılan algoritmalar, arama sürecini birkaç sıralı adıma düşürerek arama yönünü mevcut adımın durumuna göre tercih ederek azaltmaktadır.

3.2. Zamansal Lifting Şeması

Yükseltme şeması, sinyallerin çok çözünürlüklü temsilini, sinyalin uzay ve frekans ekseninde dekorelasyonuna olanak sağlayan ikinci nesil dalgacık dönüşümüdür [27]. 2-kanal yükseltme şeması, orijinal veriyi 2 alt banda ayırır.

Bunlardan birinci alt bant tek sayı indisli örneklerden, ikincisi ise çift sayı indisli örneklerden oluşmaktadır (Şekil 3.3).



Şekil 3.2. Lifting yöntemiyle 1B DWT

$x[m, n]$ iki boyutlu bir sinyal olsun. Genel yapı itibariyle öncelikle bu sinyali 1B dalgacık dönüşüm ile yatay doğrultuda sonrasında ise dikey doğrultuda ayrıştırılır. Her 1B dalgacık dönüşümüne bir ya da birden fazla yükseltme şeması uygulanabilir. Tipik bir yükseltme şeması şu üç kısımdan oluşmaktadır: Ayırma, Öngörü ve Güncelleme (Şekil 3.3).

Ayırma işleminde; $x[m, n]$ sinyali, $x_c[m, n] = x[2m, n]$ ve $x_t[m, n] = x[2m + 1, n]$ olmak üzere $x_c[m, n]$ çift altkümesi ve $x_t[m, n]$ tek alt kümelerine ayrılır.

Öngörü işleminde; $x_c[m, n]$ altkümesi, komşuluğundaki $x_t[m, n]$ alt kümesi üzerinden öngörülür. $P(.)$ öngörü operatörü, komşu tek altkümenin doğrusal bir kombinasyonudur.

$$P(x_t)[m, n] = \sum_i p_i x_t[m + i, n] \quad (3.3)$$

Burada p_i yüksek geçiren filtre katsayısıdır. $x[m, n]$ 'nin diğer bir gösterimi $x_c[m, n]$ ile öngörü farkı kullanılarak elde edilir.

$$d[m, n] = x_c[m, n] + P(x_t)[m, n] \quad (3.4)$$

Eğer sinyal lokal olarak düzgün ise $d[m, n]$ öngörü farkı değeri küçük olacaktır. Buna ilave olarak yeni gösterimle elde edilen $x_t[m, n]$ alt kümesi ve öngörü farkı $d[m, n]$ kullanılarak $x_c[m, n]$ alt kümesi elde edilebilir. Bu durumda orijinal sinyalin bilgisi de korunmuş olacaktır.

$$x_c[m, n] = d[m, n] - P(x_t)[m, n] \quad (3.5)$$

Güncelleme işleminde; $x_t[m, n]$ alt kümesi alçak geçiren süzgeçten geçirilerek $x[m, n]$ sinyalinin bir alt örneği elde edilmektedir. Bu kaba yaklaşım $d[m, n]$ öngörü farkının bir doğrusal kombinasyonunun güncellenmesiyle elde edilmektedir. $x_t[m, n]$ sinyali;

$$c[m, n] = x_t[m, n] + U(d)[m, n] \quad (3.6)$$

ile yer değiştirir. Burada $U(.)$ komşu d değerlerinin doğrusal bir kombinasyonudur.

$$U(d)[m, n] = \sum_j u_j d[m + j, n] \quad (3.7)$$

u_j alçak geçiren süzgeç katsayıdır. Her yükseltme adımının her zaman tersi alınabilir. Bu işlem sonrası bilgi kaybı olmayacaktır. Analiz ve sentez için aynı P ve U değerlerinin seçildiğini varsayalım. Yükseltme yapısı herhangi P ve U değeri için mükemmel bir geri çatmayı garanti etmektedir. $d[m, n]$ ve $c[m, n]$ verildiğinde,

$$x_t[m, n] = c[m, n] - U(d)[m, n] \quad (3.8)$$

ve denklem (3.9) kullanılarak $x_c[m, n]$ elde edilir.

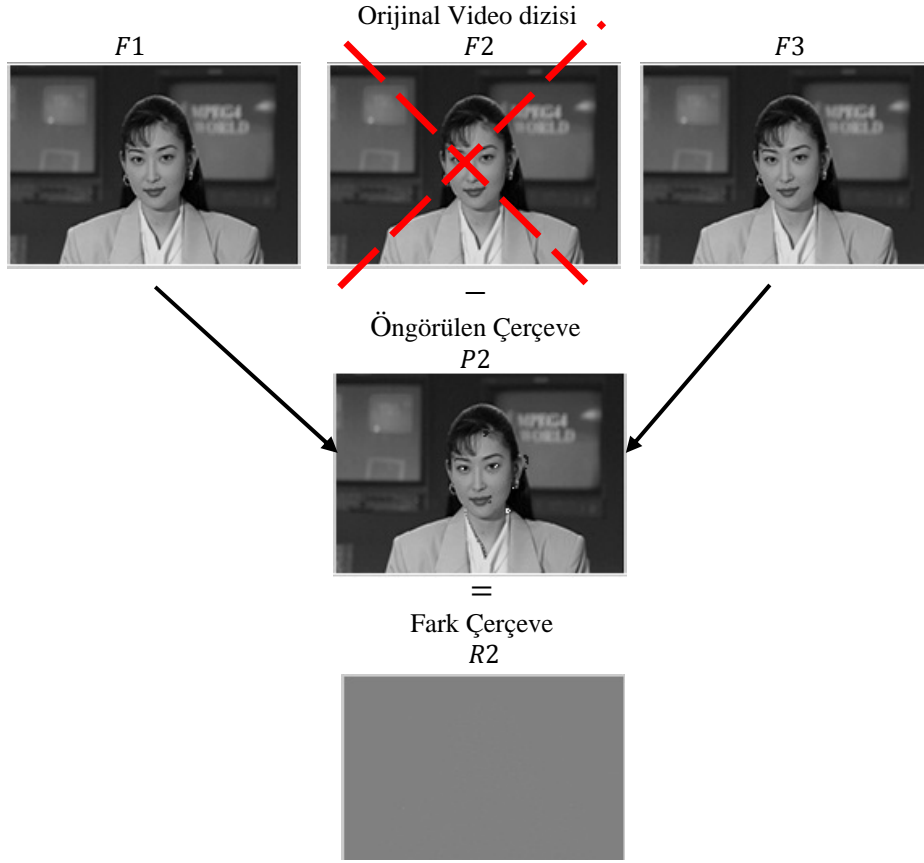
Özet olarak, 1B Yükseltme şemasında öngörü operatörü yüksek frekanslı (HF) detay (dalgacık katsayıları) alt bandını elde etmek için kullanılmaktadır. Güncelleme operatörü ise alçak frekanslı (LF) sinyalleri elde etmek yani verinin kaba temsilini elde etmede kullanılmaktadır.

LF sinyalleri ve dalgacık katsayılarını elde etmede kullanılan farklı boyutlara bağlı olarak çok sayıda yükseltme şeması bulunmaktadır. Bu boyutlar; öngörü operatörünün boyutu j ve güncelleme operatörünün boyutu k olarak tanımlanır. Bir yükseltme şeması genellikle bu boyutların oluşturduğu $[j, k]$ ikilisiyle gösterilir [28].

Bu tezde, yukarıda anlatılan zamansal yükseltme şemasının, ayırma ve öngörü bölümleri kullanılmıştır. Öngörü işlemi içerisinde hareket dengeleme yöntemleri geliştirilerek, çerçeveler arasındaki bloklar üzerinden Bölüm 3.4'te verilen gradyan yaklaşımları ile en yakın kestirimler elde edilmiştir.

3.3. Fark Verinin Kodlanması

Video dizisi içerisinde bulunan imgeler Bölüm 3.2'de anlatıldığı gibi ayırma işlemi sonucunda $x_c[m, n]$ ve $x_t[m, n]$ olmak üzere iki farklı kısma bölünürler ve hareket dengeleme yöntemleri kullanılarak tek indisli çerçevelerden çift indisli öngörü çerçeveleri oluşturulur. Çift indisli orijinal çerçeveler ile bunların öngörülerini arasındaki fark imgeleri kodlanır ve iletilir. Kod çözücü tarafında, kodlayıcı tarafından kullanılan öngörü algoritması ile tek indisli çerçevelerden çift indisli çerçevelerin öngörü imgeleri oluşturulur ve bunlara fark imgeleri eklenerek çift indisli çerçeveler geri çatılmış olur.



Şekil 3.3. Video dizisinde $F1$, $F2$, $F3$ Çerçeveleri kullanılarak öngörü ile $R2$ kalan çerçevenin bulunması

Şekil 3.3'te $F1$ ve $F3$ çerçevelerinden $P2$ çerçevesi öngörülmekte ve $P2$ ile $F2$ arasındaki fark olan $R2$ çerçevesi SPIHT ile kodlanmaktadır.

3.4. Önerilen Hareket Dengelemeli ve Zamansal Lifting Hibrit Modeli

Tez çalışmasında önerilen, hareket dengelemeli ve zamansal yükseltme yöntemini kullanan hibrit bir kodlayıcı için geliştirilen 3 farklı yöntem ele alınmıştır.

3.4.1. Hareket dengelemede kenar-uyarlamalı yükseltme yöntemi

Hareket dengelemede BMA'da kullanılan farklı yöntemler bulunmaktadır. Video sinyalinin çift indisli çerçevelerini tahmini için Gerek ve Çetin tarafından [3]'de anlatılan polifaz ayrıştırma yöntemi kullanılır. Bir x video sinyalinin i 'inci

çerçevesinin $[m, n]$ konumundaki pikseli $x_i[m, n]$ ile gösterilir. Kullanılan yöntemde öncelikle tüm video sinyal dizisinin $\sum_{i=1}^n x_i$ çerçeveleri;

$$x_t = \sum_{i=1}^n x_{2i-1} \quad (3.9)$$

$$x_{\zeta} = \sum_{i=1}^n x_{2i} \quad (3.10)$$

Olarak iki ayrı çerçeve kümesine bölünür. Bu noktada, x_{ζ} tüm video sinyali dizisinden çıkartıldıktan sonra x_t çerçeveleri ile öngörü çerçeveleri elde edilmeye çalışılmaktadır. Burada Şekil 3.4'te $2i - 1$. çerçeve, Şekil 3.5'te $2i$. çerçeve ve Şekil 3.6'da $2i + 1$. çerçeve ve çerçevelerdeki piksellerin gösterimleri verilmiştir. Bu üç çerçevenin 3B gösterimi ise Şekil 3.7'de gösterilmiştir.

	$x_{2i-1}[m-1, n-1]$	$x_{2i-1}[m-1, n]$	$x_{2i-1}[m-1, n+1]$
	$x_{2i-1}[m, n-1]$	$x_{2i-1}[m, n]$	$x_{2i-1}[m, n+1]$
	$x_{2i-1}[m+1, n-1]$	$x_{2i-1}[m+1, n]$	$x_{2i-1}[m+1, n+1]$

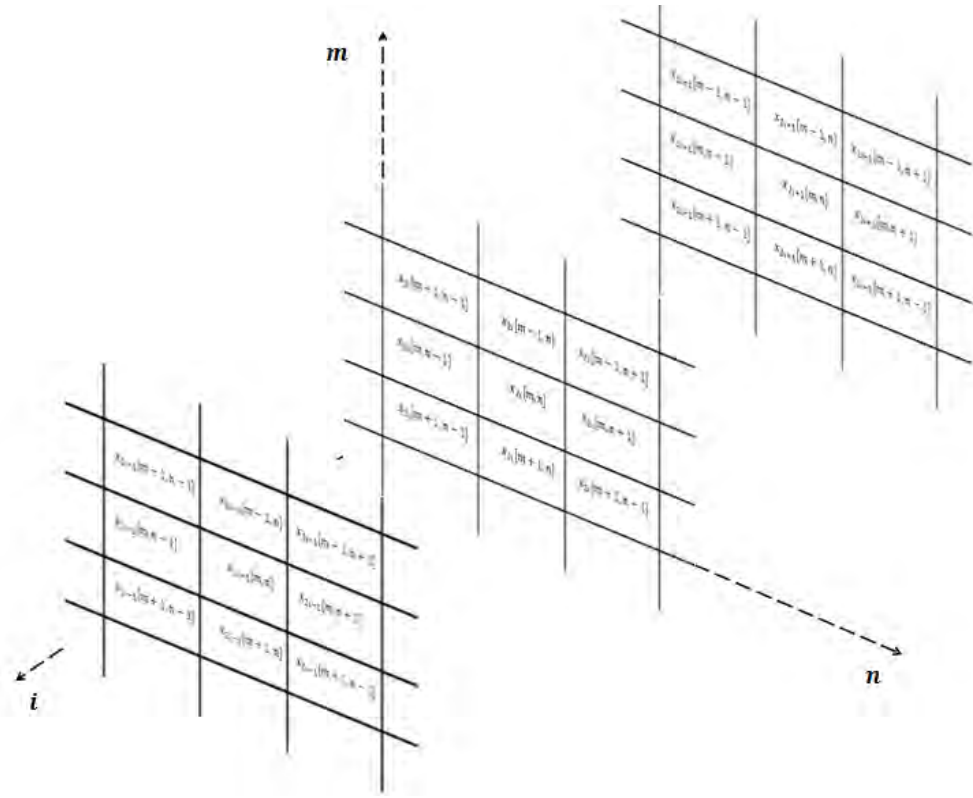
Şekil 3.4. $2i - 1$. çerçeve ve piksellerinin gösterimi

	$x_{2i}[m-1, n-1]$	$x_{2i}[m-1, n]$	$x_{2i}[m-1, n+1]$
	$x_{2i}[m, n-1]$	$x_{2i}[m, n]$	$x_{2i}[m, n+1]$
	$x_{2i}[m+1, n-1]$	$x_{2i}[m+1, n]$	$x_{2i}[m+1, n-1]$

Şekil 3.5. $2i$. çerçeve ve piksellerin gösterimi

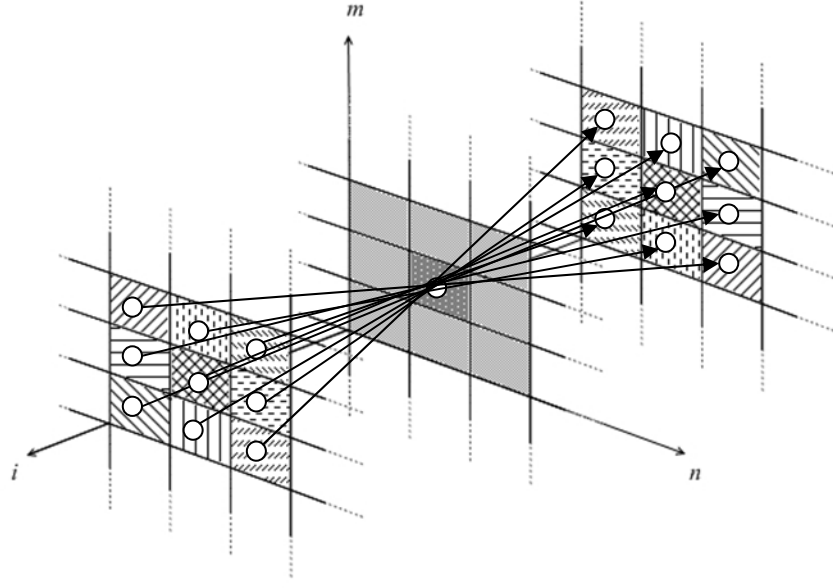
	$x_{2i+1}[m-1, n-1]$	$x_{2i+1}[m-1, n]$	$x_{2i+1}[m-1, n+1]$
	$x_{2i+1}[m, n-1]$	$x_{2i+1}[m, n]$	$x_{2i+1}[m, n+1]$
	$x_{2i+1}[m+1, n-1]$	$x_{2i+1}[m+1, n]$	$x_{2i+1}[m+1, n-1]$

Şekil 3.6. $2i+1$. çerçeve ve piksellerin gösterimi



Şekil 3.7. Ardışık üç çerçevenin zamansal doğrultuda gösterimi

3×3 blok boyutundaki ardışık 3 çerçevenin ortasındaki kalan $x_{2i}[m, n]$ çift sayı indisli çerçevenin merkez pikselinin tahmini için bu pikseli aralarına alan karşılıklı çapraz piksel çiftleri kullanılır (Şekil 3.8).



Şekil 3.8. Merkez pikseli aralarına alan 9 farklı gradyan yaklaşımı

Buna göre on yedi farklı gradyan yaklaşımı tanımlanır. Bunlardan sekiz tanesi tahmin edilecek piksel ile aynı çerçeve olduğu için kullanılabılır değildir. Geriye kalan geçerli dokuz farklı gradyan yaklaşımı şöyledir:

$$\Delta_1 = |x_{2i-1}[m-1, n-1] - x_{2i+1}[m+1, n+1]|$$

$$\Delta_2 = |x_{2i-1}[m-1, n] - x_{2i+1}[m+1, n]|$$

$$\Delta_3 = |x_{2i-1}[m-1, n+1] - x_{2i+1}[m+1, n-1]|$$

$$\Delta_4 = |x_{2i-1}[m, n-1] - x_{2i+1}[m, n+1]|$$

$$\Delta_5 = |x_{2i-1}[m, n] - x_{2i+1}[m, n]|$$

$$\Delta_6 = |x_{2i-1}[m, n+1] - x_{2i+1}[m, n-1]|$$

$$\Delta_7 = |x_{2i-1}[m+1, n-1] - x_{2i+1}[m-1, n+1]|$$

$$\Delta_8 = |x_{2i-1}[m+1, n] - x_{2i+1}[m-1, n]|$$

$$\Delta_9 = |x_{2i-1}[m+1, n+1] - x_{2i+1}[m-1, n-1]|$$

Bu gradyan yaklaşımları yardımıyla dokuz farklı $\hat{x}_{2i}[m, n]$ değeri tahmin edilebilir. Burada doğrusal bir yaklaşımla karşılıklı piksellerin konumlarının orta noktası (yarısı) aranan merkez pikseli verecektir.

$$\hat{x}_{2i}^1[m, n] = (x_{2i-1}[m-1, n-1] + x_{2i+1}[m+1, n+1])/2$$

$$\hat{x}_{2i}^2[m, n] = (x_{2i-1}[m-1, n] + x_{2i+1}[m+1, n])/2$$

$$\hat{x}_{2i}^3[m, n] = (x_{2i-1}[m-1, n+1] + x_{2i+1}[m+1, n-1])/2$$

$$\hat{x}_{2i}^4[m, n] = (x_{2i-1}[m, n-1] + x_{2i+1}[m, n+1])/2$$

$$\hat{x}_{2i}^5[m, n] = (x_{2i-1}[m, n] + x_{2i+1}[m, n])/2$$

$$\hat{x}_{2i}^6[m, n] = (x_{2i-1}[m, n+1] + x_{2i+1}[m, n-1])/2$$

$$\hat{x}_{2i}^7[m, n] = (x_{2i-1}[m+1, n-1] + x_{2i+1}[m-1, n+1])/2$$

$$\hat{x}_{2i}^8[m, n] = (x_{2i-1}[m+1, n] + x_{2i+1}[m-1, n])/2$$

$$\hat{x}_{2i}^9[m, n] = (x_{2i-1}[m+1, n+1] + x_{2i+1}[m-1, n-1])/2$$

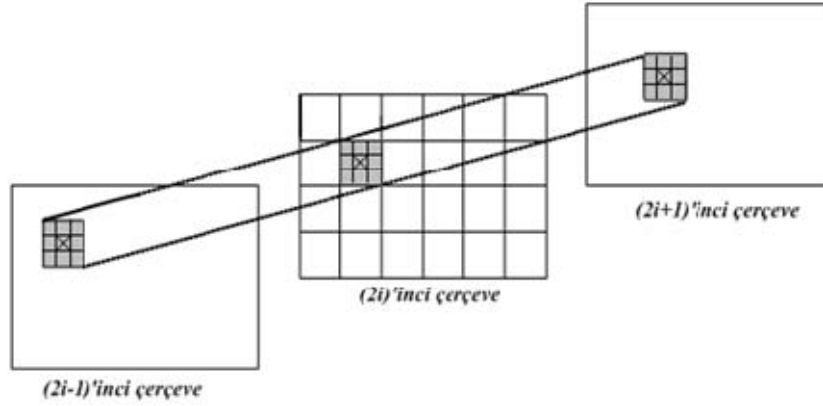
Merkez pikseli elde etmek için dokuz farklı gradyan içerisinde en az farka sahip değer aranan $\hat{x}_{2i}[m, n]$ değerini verecektir. Bu durumda çift indisli çerçevedeki 3×3 'lük blok merkez piksel yardımıyla, dolayısıyla bu işlem tüm çerçeveye döngülü olarak işletilirse çerçevedeki imge elde edilmiş olacaktır.

Video sinyali üzerindeki bir kenar oluşumunun ardı ardına gelen çerçevelerdeki yer değiştirmesi, yukarıdaki yöntemlerde kullanılan 3×3 'lük blok sınırını aşabilir. Bu durumda iki çerçeve sonra bu sınırın dışına çıkmış olan hareketli nokta polifaz ayrıştırma analizinin ve blok karşılaştırma yönteminin tahmin kalitesini düşürür. Bunun için $(2i-1)$ 'inci çerçevedeki 3×3 'lük bloğun $(2i+1)$ 'inci çerçevedeki karşılığının konumunun bulunmaya çalışılması ve tahmin edilecek pikselin bu yeni konuma göre $(2i)$ 'inci çerçevede konumlandırılması gerekir. Oluşan $3 \times 3 \times 3$ 'lük dörtgen prizma üzerinde yöntemlerin uygulanması, sıkıştırmanın neden olduğu kaybı azaltacaktır.

Hareket dengelemeli yükseltme şemasının matematiksel gösterimi için $W_{2i-1,2i+1}$ operatörü tanımlanır. $W_{2i-1,2i+1}$ operatörü, seçilen hareket dengeleme şemasına göre $2i-1$ çerçevesini $2i+1$ çerçevesinin koordinat düzlemine bağlar [11].

$$h[m, n] = \frac{1}{2}(x_{2i+1}[m, n] - W_{2i-1,2i+1}(X_{2i-1})[m, n]) \quad (3.11)$$

Denklemdede; tahmin edilecek olan $(2i)$ çerçevesi hesap dışı olduğu için $(2i - 1)$ ve $(2i + 1)$ çerçeveleri tahmin sırasında ardışık kabul edilirler ve $x_{2i-1}[m, n]$ ve $x_{2i+1}[m, n]$, ardışık iki geçerli video çerçevesi üzerinde $[m, n]$ konumlu iki noktayı temsil etmektedirler. $W_{2i-1, 2i+1}$ operatörünün kullanacağı hareket dengeleme şeması blok-eşleştirme yöntemi ile bulunur. Blok eşleştirmede $(2i)$ çerçevesindeki merkez pikselin konumunu bulabilmek için $(2i + 1)$ çerçevesindeki 3×3 'lük bir bloğun konumu $(2i - 1)$ çerçevesi üzerinde işaretlenir ve bu komşuluktaki sınırlı bir arama bölgesinde yapılan blok taraması sonucunda merkez pikseller arasındaki minimum farka sahip karşılıklı bloklar seçilir. Burada ardışık çerçevelerdeki merkez pikselleri birleştiren doğru parçasının orta noktası her bir blok için merkez pikselin konumunu verecektir.

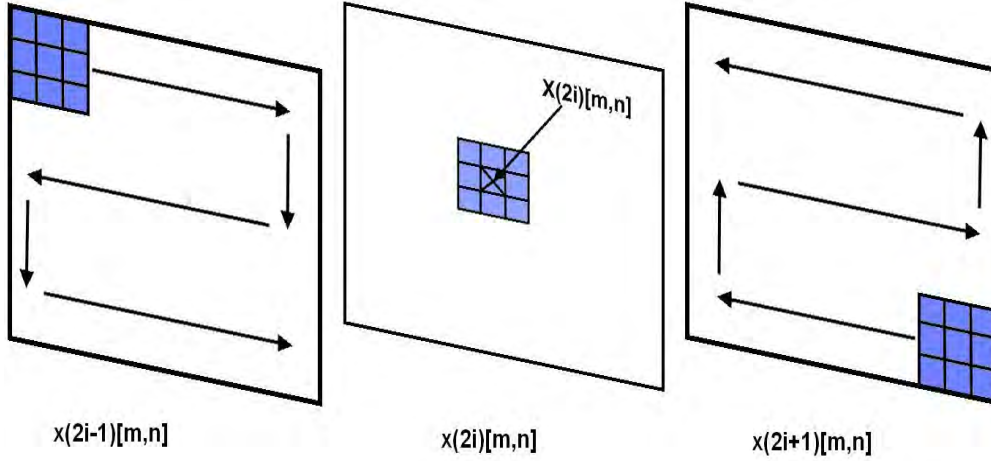


Şekil 3.9. Bloklar üzerindeki merkez piksellerin eşleştirme ile tahmin edilecek olan pikselin konumlandırılması

Bu yöntem kullanılarak blok eşleştirme uygulanan $(2i + 1)$ 'inci çerçevedeki her bir 3×3 'lük bloğun $(2i - 1)$ 'inci çerçevedeki eşleştirmesi tamamlandığında $(2i)$ 'inci blokta değerleri hesaplanmamış bağlantısız pikseller olabilir. Hesaplanmamış olan piksellerin değerleri yukarıda anlatılan gradyan yöntemiyle bulunur. Böylece hareket eden blokların arasında kalan bağlantısız pikseller de doldurulmuş olur [29].

3.4.2. Hareket dengelemede blok karşılaştırma yöntemi

Bir önceki bölümde anlatılan $3 \times 3 \times 3$ 'lük kübün merkezindeki $x_{2i}[m,n]$ piksel değerinin tahmini için farklı bir yöntem olarak ardışık x_t 'de Şekil 2.8.'de gösterildiği gibi 3×3 boyutundaki bloklar karşılaştırılmaktadır.



Şekil 3.10. Blok karşılaştırma ile tahmin edilecek piksel için blokların konumlandırılması

$(2i - 1)$ 'deki blok sol üst köşeden başlayarak sağ alt köşeye kadar hareket ettirilirken, $(2i + 1)$ 'deki blok sağ alt köşeden sol üst köşeye doğru zıt doğrultuda hareket ettirilmektedir. Bu sırada bloklar içerisinde var olan karşılıklı piksellerin, farklarının kareleri toplamının en küçük olduğu değer tarama esnasında saptanır. Bulunan yerin $(2i - 1)$ ve $(2i + 1)$ çerçevelerindeki konumlarının geometrik ortalaması aranan merkez pikselin konumunu verecektir. Bu işlem $2i$. çerçevedeki tüm piksel değerleri hesaplanana kadar sürecektir. Aynı yaklaşımı video dizisindeki tüm x_c için uygulandığında, x_c için bir öngörü elde edilmiş olacaktır [30].

Bölüm 3.4.1 ve Bölüm 3.4.2'de anlatılan, YUV biçimli CIF (352×288) boyutundaki “bus”, “coastguard” ve “container” videolarının ilk beş çerçevesi üzerinde deneysel olarak uygulanmıştır. 3×3 'lük bloklar kullanılarak elde edilen deneysel sonuçlar Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Kenar Uyarlamalı Yükseltme ve Blok karşılaştırma yöntemlerinin OKH değerleri (%)

Çerçeve No	<i>bus</i>		<i>coastguard</i>		<i>container</i>	
	3.4.1 Yöntemi	3.4.2 Yöntemi	3.4.1 Yöntemi	3.4.2 Yöntemi	3.4.1 Yöntemi	3.4.2 Yöntemi
1	---	---	---	---	---	---
2	1,36	1,16	0,32	0,15	0,32	0,08
3	---	---	---	---	---	---
4	1,43	1,25	0,3	0,14	0,32	0,09
5	---	---	---	---	---	---

Tek indisli çerçeveler kodlama sırasında tutulan çerçevelerdir. Bu çerçeveler geri çatma sırasında değişmeden tutulduklarından karşılaştırma yapılırken dikkat edilecek çerçeveler çift indisli çerçeveler olacaktır. Yöntemlerle elde edilen çerçevelerin *Y* kanallarının özgün çerçevelerle karşılaştırmalı öngörülerinin OKH sonuçları incelendiğinde, her bir videoda 3.4.2 yöntemi ile elde edilen değerlerde iyileşmeler görülmüştür. Görsel iyileşmeler Şekil 3.11’de verilmiştir.



(a)



(b)



(c)



(d)



(e)



(f)

Şekil 3.11. Kenar uyarlamalı yükseltme ve blok karşılaştırma yöntemleri görselleri

(a) *bus* videosunun kenar uyarlamalı yükseltme yöntemi ile elde edilmiş 2. çerçeve öngörüsü

(b) *bus* videosunun blok karşılaştırma yöntemi ile elde edilmiş 2. çerçeve öngörüsü

(c) *coastguard* videosunun kenar uyarlamalı yükseltme ile elde edilmiş 2. çerçeve öngörüsü

(d) *coastguard* videosunun blok karşılaştırma yöntemi ile elde edilmiş 2. çerçeve öngörüsü

(e) *container* videosunun kenar uyarlamalı yükseltme ile elde edilmiş 2. çerçeve öngörüsü

(f) *container* videosunun blok karşılaştırma yöntemi ile elde edilmiş 2. çerçeve öngörüsü

Elde edilen deney sonuçlarından yola çıkarak, blok karşılaştırma yönteminin daha başarılı sonuçlar verdiği görülmüştür. Bunun dışında, imge üzerindeki hareketli nesnelerin farklı boyutlardaki bloklar ($3 \times 3, 5 \times 5, 7 \times 7$) ve bu blokların farklı arama komşuluğu değerleri (2, 4, 6 piksel) ile tespiti yapılmıştır. Yapılan çalışmada, deney videolarının çift indisli çerçevelerinin OKH değerleri hesaplanmıştır (Çizelge 3.2, Çizelge 3.3, Çizelge 3.4).

Çizelge 3.2. “Bus” videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%)

Arama Komşuluğu/ Çerçeve No	<i>bus</i>								
	<i>Blok boyutu = 3 × 3</i>			<i>Blok boyutu = 5 × 5</i>			<i>Blok boyutu = 7 × 7</i>		
	2	4	6	2	4	6	2	4	6
1	---	---	---	---	---	---	---	---	---
2	1,59	1,53	1,59	1,60	1,64	1,64	1,61	1,62	1,66
3	---	---	---	---	---	---	---	---	---
4	1,71	1,63	1,63	1,73	1,74	1,76	1,74	1,75	1,80
5	---	---	---	---	---	---	---	---	---

Çizelge 3.3. “Coastguard” videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%)

Arama Komşuluğu/ Çerçeve No	<i>coastguard</i>								
	<i>Blok boyutu = 3 × 3</i>			<i>Blok boyutu = 5 × 5</i>			<i>Blok boyutu = 7 × 7</i>		
	2	4	6	2	4	6	2	4	6
1	---	---	---	---	---	---	---	---	---
2	0,19	0,24	0,32	0,15	0,18	0,24	0,12	0,13	0,16
3	---	---	---	---	---	---	---	---	---
4	0,19	0,27	0,35	0,14	0,19	0,24	0,10	0,13	0,17
5	---	---	---	---	---	---	---	---	---

Çizelge 3.4. “Container” videosu üzerinde farklı blok boyutu ve arama komşuluğu değerlerinde OKH değerleri (%)

	<i>container</i>								
	<i>Blok boyutu = 3 × 3</i>			<i>Blok boyutu = 5 × 5</i>			<i>Blok boyutu = 7 × 7</i>		
Arama Komşuluğu/ Çerçeve No	2	4	6	2	4	6	2	4	6
1	---	---	---	---	---	---	---	---	---
2	0,032	0,071	0,13	0,020	0,025	0,035	0,018	0,019	0,02
3	---	---	---	---	---	---	---	---	---
4	0,021	0,054	0,11	0,015	0,017	0,026	0,014	0,014	0,015
5	---	---	---	---	---	---	---	---	---

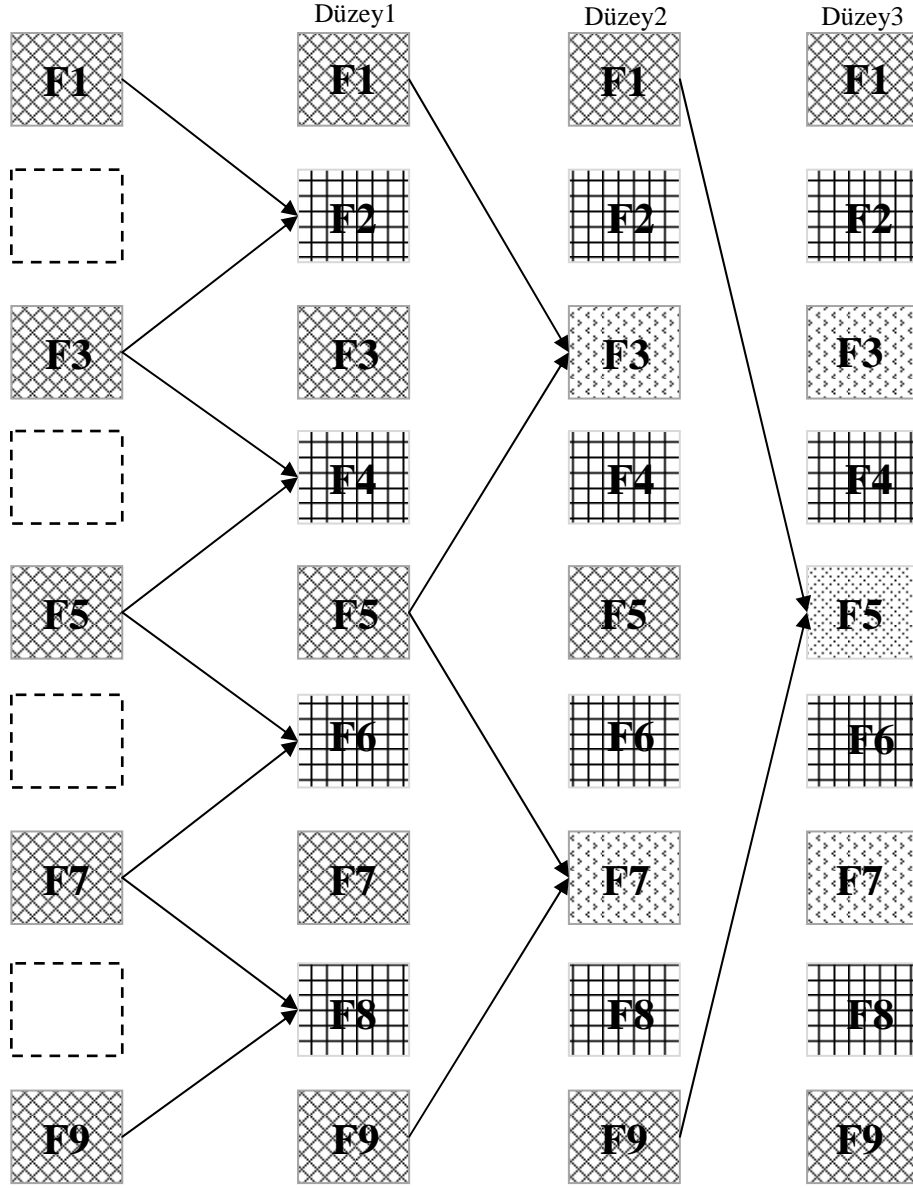
3.4.3. Hareket dengelemede farklı düzeylerde blok karşılaştırma yöntemi

Bölüm 3.4.2’de verilen Blok karşılaştırma yöntemi ile elde edilen deneysel sonuçlar kaynak video dizisi içerisinde nesnelerin yatay ekseninde hızlı kaydığı durumlarda düşük blok boyutu ile elde edilen öngörülerde daha başarılı sonuçlar vermiştir. Buna karşın video dizisi içerisinde daha az hareketin olduğu, kayma hareketlerinin daha yavaş olduğu ya da arka plan görüntüsünün değişiminin olmadığı durumlarda daha büyük boyutta blokların kullanılmasının kodlayıcı performansını artırıcı etkisinin olduğunu göstermektedir. Söz konusu performans artışı hareket dengelemenin simetrik olmadığı durumlarda daha sınırlı kalmaktadır.

Blok karşılaştırma yönteminde; $(2i)$ çerçevesindeki merkez pikselin konumunu bulabilmek için $(2i + 1)$ çerçevesindeki $n \times n$ ’lik bir bloğun konumu $(2i - 1)$ çerçevesi üzerinde işaretlenir ve bu komşuluktaki sınırlı bir arama bölgesinde yapılan blok taraması sonucunda merkez pikseller arasındaki minimum farka sahip karşılıklı bloklar seçilmektedir [31].

Bu çalışmada, bundan önce yapılan Bölüm 3.4.2.’deki çalışmadaki, hareket dengelemenin en yakın çerçeve çiftlerinden yapılan öngörüsü Düzey1 blok karşılaştırma yöntemi olarak isimlendirilmiştir. Şekil 3.12’de gösterildiği üzere,

hareket dengeleme ile öngörüsü yapılacak çerçeveye olan uzaklıkların artırılmasıyla Düzey2 ve Düzey3 blok karşılaştırma yöntemleri tanımlanmıştır.



Şekil 3.12. Hareket dengelemede Düzey1, Düzey2 ve Düzey3 blok karşılaştırma yöntemi

Düzey1 blok karşılaştırma yönteminde; Bölüm 3.4.2.'de olduğu gibi $(2i)$. çerçeveyi elde etmek için $(2i - 1)$ ve $(2i + 1)$ numaralı çerçeveler kullanılmaktadır.

Düzey2 blok karşılaştırma yönteminde ise; $(4i - 1)$. çerçeveyi elde etmek için $(4i - 3)$ ve $(4i + 1)$ numaralı çerçeveler kullanılmaktadır.

Son olarak Düzey3 blok karşılaştırma yönteminde ise; $(8i - 3)$. çerçeveyi elde etmek için bu kez $(8i - 7)$ ve $(8i + 1)$ numaralı çerçeve çiftleri kullanılmaktadır. $(i = 1,2,3, \dots n)$

Her bir düzeyde elde edilen farklı video dizisi görüntüleri ve bunların tümünün OKH değerleri hesaplanmıştır.

“*bus*” video dizisinin Düzey1, Düzey2 ve Düzey3 Blok Karşılaştırma Yöntemi’nden elde edilen ilk 9 çerçevesinin görselleri sırasıyla, Şekil 3.13, Şekil 3.14 ve Şekil 3.15’te gösterilmiştir. Çizelge 3.5’te ise “*bus*” videosuna ait her bir düzey için hesaplanmış çerçevelerin OKH değerleri verilmiştir.

Benzer olarak, “*coastguard*” video dizisinin Düzey1, Düzey2 ve Düzey3 Blok Karşılaştırma Yöntemi’nden elde edilen ilk 9 çerçevesinin görselleri sırasıyla, Şekil 3.16, Şekil 3.17 ve Şekil 3.18’de gösterilmiştir. Çizelge 3.6’da ise “*coastguard*” videosuna ait her bir düzey için hesaplanmış çerçevelerin OKH değerleri verilmiştir.

Son olarak, “*container*” video dizisinin Düzey1, Düzey2 ve Düzey3 Blok Karşılaştırma Yöntemi’nden elde edilen ilk 9 çerçevesinin görselleri sırasıyla, Şekil 3.19, Şekil 3.20 ve Şekil 3.21’de gösterilmiştir. Çizelge 3.7’de ise “*container*” videosuna ait her bir düzey için hesaplanmış çerçevelerin OKH değerleri verilmiştir.

Her bir video dizisi için elde edilen çizelgelerde, çerçeve sütunlarındaki değerlerin her düzeyde aynı kaldığı görülmektedir. Bu durum her üst ayrışım seviyesinden sonra kayıpsız kodlamanın varsayıldığı için elde edilmiş bir sonuçtur.



(a)



(b)



(c)



(d)



(e)



(f)

Şekil 3.13. Bus videosu Düzey1 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F4 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F6 çerçevesi



(g)



(h)



(i)

Şekil 3.12. Bus videosu Düzey1 Blok Eşleştirme Yöntemi Görselleri (devamı)

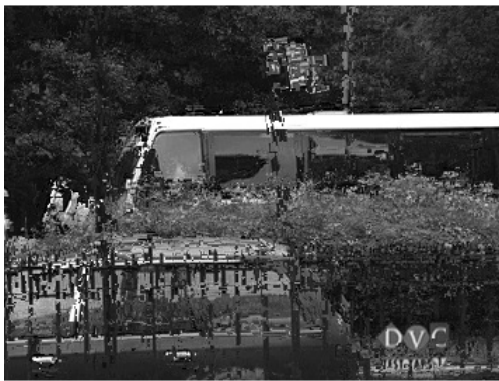
- (g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F8 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.14. *Bus* videosu Düzey2 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F4 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F6 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F6 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(f)



(h)



(i)

Şekil 3.15. *Bus* videosu Düzey3 Blok Eşleştime Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F4 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F6 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F6 çerçevesi
(i) F9 çerçevesi

Çizelge 3.5. *Bus* videosunun ilk 9 çerçevesi üzerinde 3x3 arama bloğu ile 4 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%)

	<i>Bus</i>								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
Düzey1	---	1,53	---	1,63	---	1,65	---	1,73	---
Düzey2	---	1,53	1,73	1,63	---	1,65	1,84	1,73	---
Düzey3	---	1,53	1,73	1,63	0,243	1,65	1,84	1,73	---



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.16. Coastguard videosu Düzey1 Blok Eşleştime Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F4 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F6 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F6 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.17. *Coastguard* videosu Düzey2 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F2 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F2 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F2 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.18. *Coastguard* videosu Düzey3 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F2 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F2 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F2 çerçevesi
(i) F9 çerçevesi

Çizelge 3.6. *Coastguard* videosunun ilk 9 çerçevesi üzerinde 7×7 arama bloğu ile 2 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%)

	<i>Coastguard</i>								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
Düzey1	---	0,12	---	0,10	---	0,15	---	0,12	---
Düzey2	---	0,12	0,40	0,10	---	0,15	0,43	0,12	---
Düzey3	---	0,12	0,40	0,10	0,63	0,15	0,43	0,12	---



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.19. *Container* videosu Düzey1 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F2 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F2 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F2 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.20. Container videosu Düzey2 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F2 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F2 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F2 çerçevesi
(i) F9 çerçevesi



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Şekil 3.21. *Container* videosu Düzey3 Blok Eşleştirme Yöntemi Görselleri

- (a) F1 çerçevesi (b) F1 ve F3 çerçevelerinden elde edilen F2 çerçevesi
(c) F3 çerçevesi (d) F3 ve F5 çerçevelerinden elde edilen F2 çerçevesi
(e) F5 çerçevesi (f) F5 ve F7 çerçevelerinden elde edilen F2 çerçevesi
(g) F7 çerçevesi (h) F7 ve F9 çerçevelerinden elde edilen F2 çerçevesi
(i) F9 çerçevesi

Çizelge 3.7. *Container* videosunun ilk 9 çerçevesi üzerinde 7×7 arama bloğu ile 2 arama komşuluğu üzerinden Düzey1, Düzey2 ve Düzey3 blok karşılaştırma OKH değerleri (%)

	<i>Container</i>								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
Düzey1	---	0,018	---	0,014	---	0,017	---	0,015	---
Düzey2	---	0,018	0,055	0,014	---	0,017	0,050	0,015	---
Düzey3	---	0,018	0,055	0,014	0,054	0,017	0,050	0,015	---

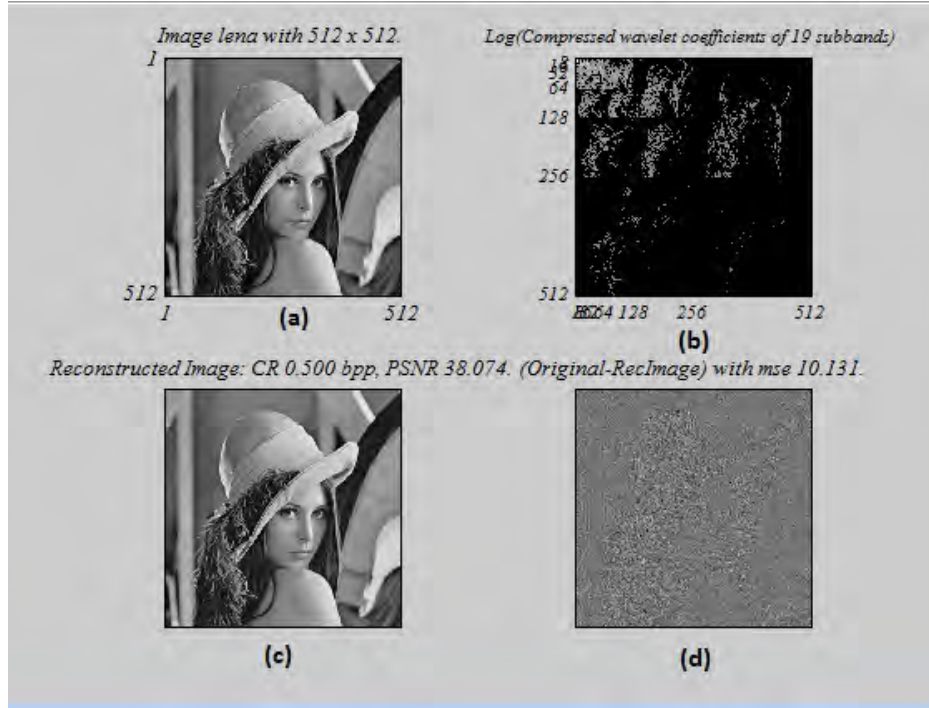
4. DENEYSEL SONUÇLAR

Bölüm 3.4’te önerilen yöntemlerle elde edilen çerçeveler “*Hiyerarşik Ağaçlarda Küme Bölüntüleme*” (SPIHT) diye bilinen dalgacık kodlama algoritmasını kullanan dalgacık katsayılarının bit gösterimini veren bir kodlayıcı ile kodlanmıştır [32]. Uygulama deneyleri SPIHT kodlayıcı ile YUV biçimli “*bus*”, “*coastguard*” ve “*container*” deney videolarının 145 çerçevelik CIF (352×288) boyutundaki imgeleri üzerinde yapılmıştır. Her bir yönteme ait sıkıştırma oranına (CR) karşılık gelen farklı durumlardaki PSNR değerleri ve deney videoları MPEG2 sıkıştırma algoritması ile sıkıştırılmasıyla elde edilen PSNR değerleri verilerek bunlar ayrıntılı bir şekilde çizelgelerde toplanmıştır. Böylelikle deney videoları önerilen yöntemler üzerinden ve referans olarak MPEG2 üzerinden karşılaştırılmıştır.

$$CR = \frac{O}{C} \quad (4.1)$$

Sıkıştırma oranının elde edilmesinde, C sıkıştırılmış veri miktarını O ise orijinal veri miktarını göstermektedir.

512×512 boyutundaki Lena imgesi için 0.5 bpp sıkıştırma oranı ile elde edilen sonuçlar Şekil 4.1’de verilmiştir.

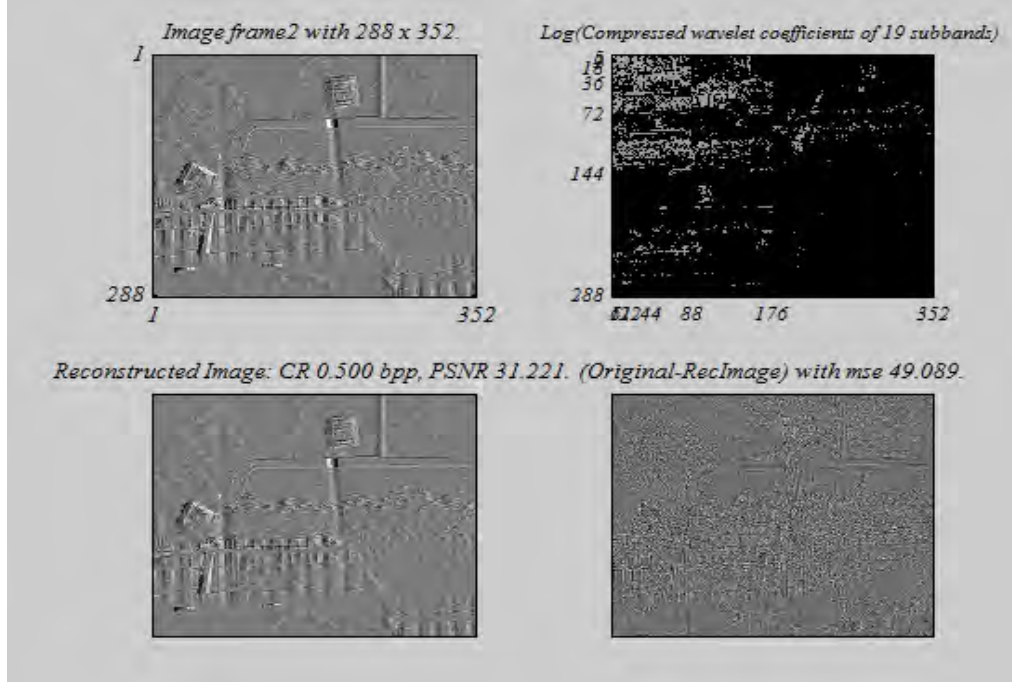


Şekil 4.1. "Lena" imgesinin SPIHT kodlayıcı ile 0.5 bpp oranında sıkıştırılması ile elde edilen sonuçlar

- (a) Orijinal imge (b) Sıkıştırılmış dalgacık katsayıları (c) Geri çatılmış imge
(d) Orijinal imge ile geri çatılan imge arasındaki fark görüntüsü

Kodlayıcının ürettiği sıkıştırılmış "Lena" imgesi orijinaline göre 1/16 oranında küçülmüştür. Orijinal imgede her bir piksel için 8 bit gerekli iken bu durum 0.5 bpp sıkıştırma oranı ile her piksel için 0.5 bite düşürülmüştür. Elde edilen dalgacık katsayıları ile geri çatılan imgenin PSNR değeri 38.074 dB gibi bir değer almıştır. Orijinal imge ile geri çatılan imge arasındaki farkın OKH değeri ise 10.131 olarak elde edilmiştir.

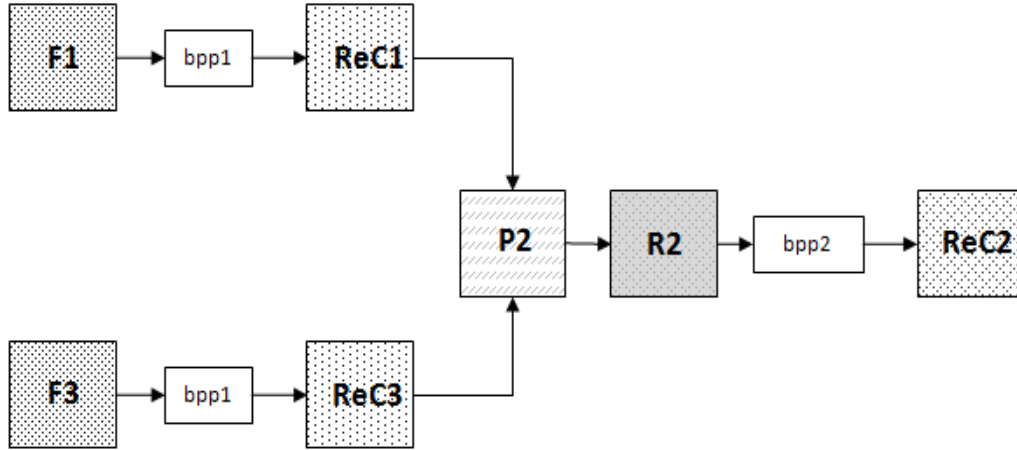
Benzer olarak, 352×288 boyutundaki "bus" videosunun hareket dengelemede blok karşılaştırma yöntemini kullanılarak elde edilen ikinci çerçevenin orijinali ile arasındaki farkının kodlanması sonucunda elde edilen sonuçlar ve görüntüler Şekil 4.2'de verilmiştir.



Şekil 4.2. "bus" videosunun 2. çerçevesinin SPIHT kodlayıcı ile 0.5 bpp oranında sıkıştırılması ile elde edilen sonuçlar

4.1. Hareket Dengelemede Kenar Uyarlamalı Yükseltme Yöntemi'nin Deneysel Sonuçları

Deney çalışmalarında, YUV biçimli "bus", "coastguard" ve "container" videolarının (30 fps.) Y kanalı üzerinde çalışılmıştır. Tek indisli video imgeleri, "bpp1" piksel başına bit sayısı değeri ile kodlandıktan sonra geri çatılmaktadır. Bu noktada arada kalan çift indisli çerçeveyi tahmin etmek için Bölüm 3.4.1'de anlatılan yöntem kullanılmakta ve çift indisli öngörüler (P) elde edilmektedir. Orijinal olan çift indisli çerçeveler ile öngörülerden elde edilen farklar (R) "bpp2" değeri ile kodlanarak geri çatılmış çift indisli çerçeveler (ReC) elde edilir. $ReC2$ çerçevesinin elde edilmesi Şekil 4.3'de gösterilmiştir.

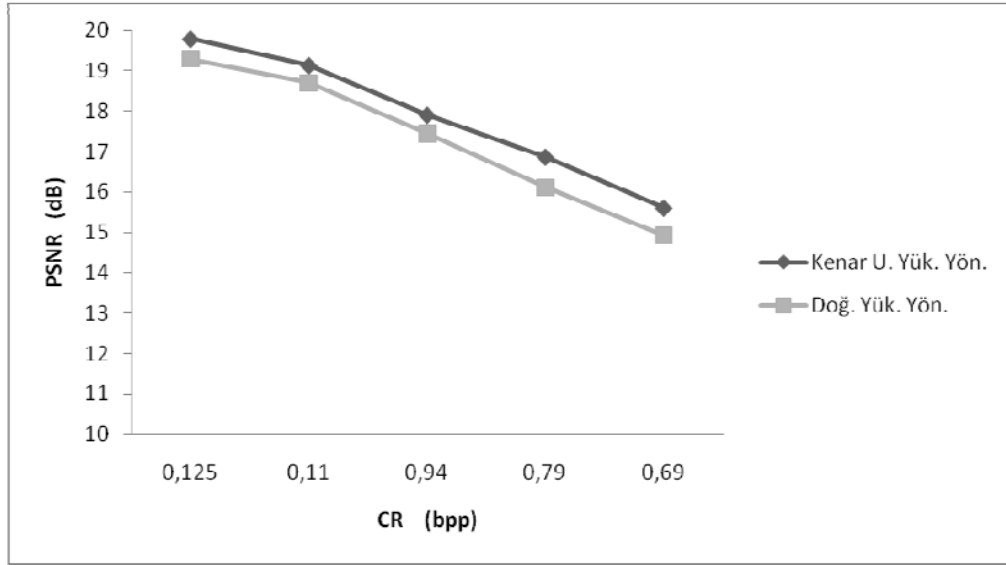


Şekil 4.3. Düzey1 ayrışımının ve geri çatılmasının F1 ve F3 çerçeveleriyle gösterimi

Yapılan deneylerde farklı “*bpp1*” ve “*bpp2*” değerleri için CR ve PSNR değerleri elde edilmiştir. Ayrıca tüm çerçeveler, kenar uyarlama yönteminin uygulanmadığı, doğrudan yükseltme durumundaki PSNR değerleri ile aynı tabloda verilerek yöntemin verimliliği daha kolay karşılaştırılabilmiştir. CR değerleri, geri çatılan çerçevelerin dosya büyüklüklerinin toplamının orijinal çerçeve dosyalarının büyüklüklerinin toplamına oranı olarak hesaplanmıştır.

Çizelge 4.1. “*bus*” videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri.

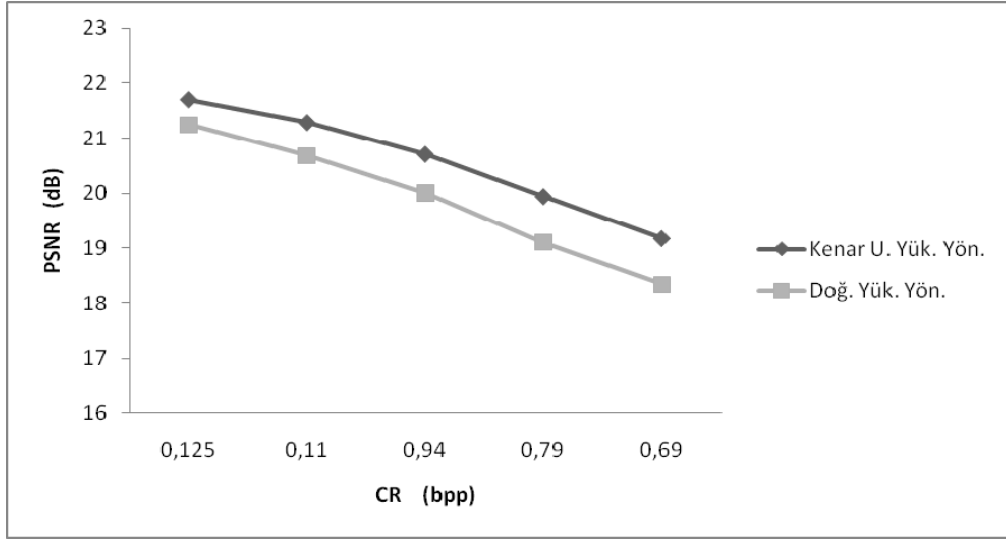
bpp1, bpp2	“ <i>bus</i> ”		
	CR	Kenar Yük. Yön. PSNR	Doğrudan Yük. Yön. PSNR
bpp1 = 1, bpp2 = 1	8 : 1	19,7917	19,2959
bpp1 = 1, bpp2 = 0,75	9,09 : 1	19,1359	18,6998
bpp1 = 1, bpp2 = 0,5	10,64 : 1	17,9106	17,4388
bpp1 = 1, bpp2 = 0,25	12,66 : 1	16,8692	16,1095
bpp1 = 1, bpp2 = 0,1	14,49 : 1	15,6051	14,9285



Şekil 4.4. "bus" videosu için PSNR – (1/CR) grafiği.

Çizelge 4.2. "coastguard" videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri.

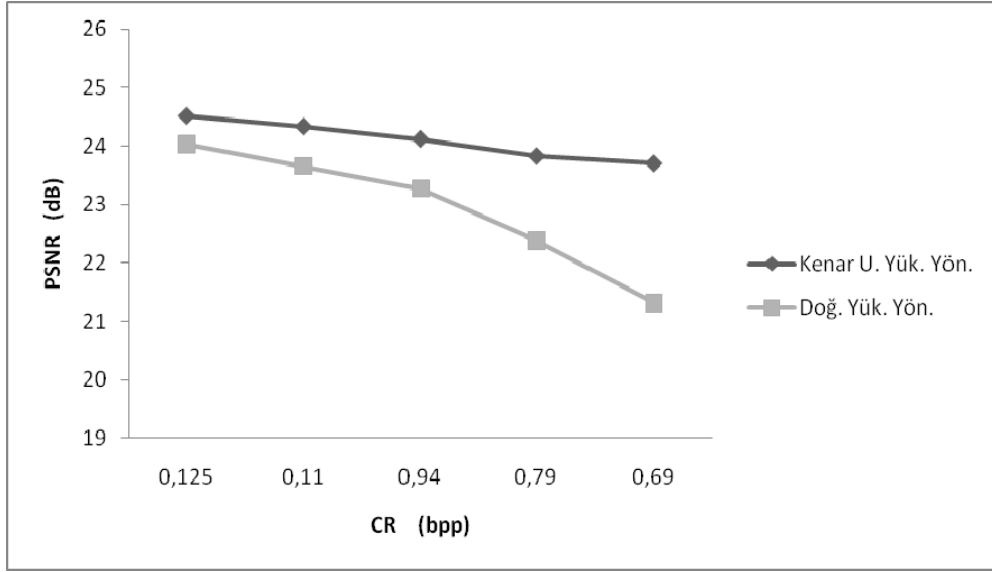
<i>"coastguard"</i>			
bpp1, bpp2	CR	Kenar Yük. Yön.	Doğrudan Yük. Yön.
		PSNR	PSNR
bpp1 = 1, bpp2 = 1	8 : 1	21,7015	21,2431
bpp1 = 1, bpp2 = 0,75	9,09 : 1	21,2868	20,6817
bpp1 = 1, bpp2 = 0,5	10,64 : 1	20,7094	20,0034
bpp1 = 1, bpp2 = 0,25	12,66 : 1	19,9425	19,1055
bpp1 = 1, bpp2 = 0,1	14,49 : 1	19,1780	18,3445



Şekil 4.5. "coastguard" videosu için PSNR – (1/CR) grafiği

Çizelge 4.3. "container" videosu için Kenar Yük. Yön. ve Doğrudan Yük. Yön. PSNR değerleri

<i>"container"</i>			
bpp1, bpp2	CR	Kenar Yük. Yön.	Doğrudan Yük. Yön.
		PSNR	PSNR
bpp1 = 1, bpp2 = 1	8 : 1	24,5099	24,0360
bpp1 = 1, bpp2 = 0,75	9,09 : 1	24,3299	23,6541
bpp1 = 1, bpp2 = 0,5	10,64 : 1	24,1185	23,2752
bpp1 = 1, bpp2 = 0,25	12,66 : 1	23,8260	22,3842
bpp1 = 1, bpp2 = 0,1	14,49 : 1	23,7025	21,3084



Şekil 4.6. "container" videosu için PSNR – (1/CR) grafiği.

4.2. Hareket Dengelemede Blok Karşılaştırma Yöntemi'nin Deneysel Sonuçları

Geliştirilen blok karşılaştırma yöntemi deneyde kullanılan video imgelerine uygulanmıştır. Bu uygulama da yöntem, Şekil 4.3'te gösterilen öngörü öncesine dâhil edilmiştir. Deney videolarının kodlanması aşamasında Çizelge 3.2, Çizelge 3.3 ve Çizelge 3.4'te elde edilen sonuçlar göz önünde bulundurularak, sırasıyla "bus" için blok boyutu 3, arama komşuluğu 4; "coastguard" için blok boyutu 7, arama komşuluğu 2; "container" için ise blok boyutu 7, arama komşuluğu 2 seçilmiştir.

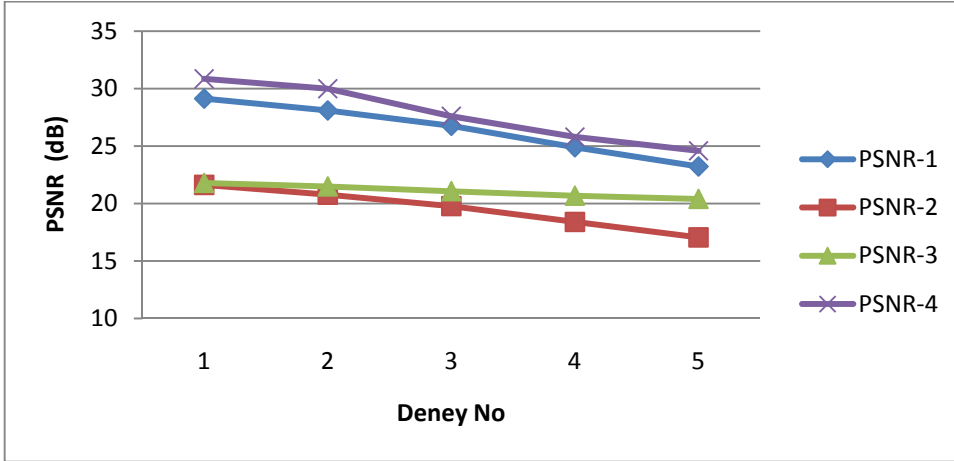
Deney videoları, blok karşılaştırma yöntemini öngöründe kullanan, farklı "bpp1" ve "bpp2" değerleri ile SPIHT kodlayıcı kullanılarak kodlanmıştır. Çizelgelerde gösterilen PSNR-1 değerleri ilgili satırlarda verilmiş "bpp1" ve "bpp2" ile elde edilen değerlerdir. PSNR-2 değerleri, ilgili satırda elde edilen CR değeri ile kodlanarak elde edilen değerlerdir. PSNR-3 değerleri ise, tüm çerçevelerin hiçbir yöntem kullanılmadan doğrudan o satırdaki CR değeri ile kodlanmasından elde edilmiştir. Bunların yanında referans olarak MPEG2

sıkıştırması imge guruplarına IBI sırasıyla uygulanarak PSNR-4 değerleri elde edilmiştir. "bus" videosunun Düzey2'de elde edilen PSNR grafiği

Çizelge 4.4. "bus" videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri

"bus"						
bpp1	bpp2	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	1	8 : 1	29,1281	21,6289	21,7944	30,8591
1	0,75	9,09 : 1	28,0962	20,7733	21,4890	29,9932
1	0,5	10,64 : 1	26,7581	19,7792	21,0737	27,5991
1	0,25	12,66 : 1	24,9027	18,4008	20,6826	25,7961
1	0,1	14,49 : 1	23,2227	17,0526	20,4052	24,5924

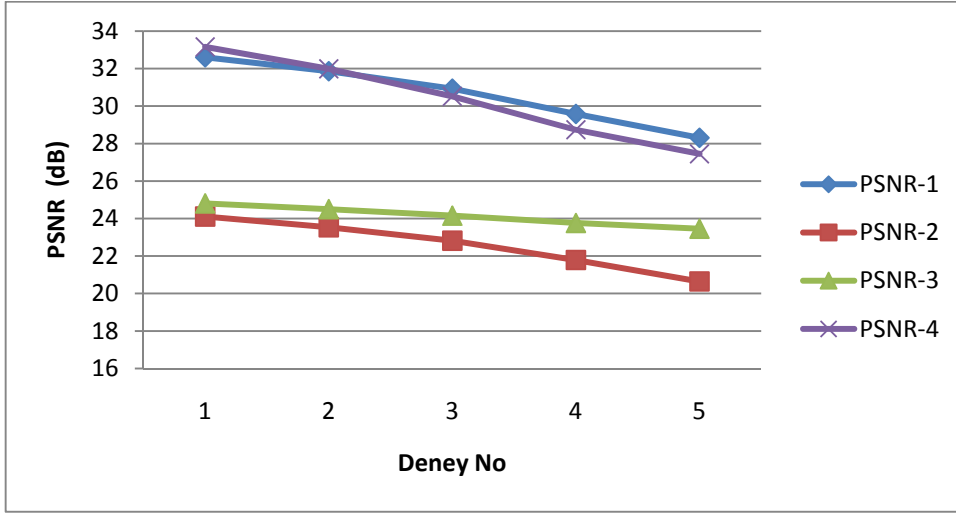
Çizelge 4.5. "bus" videosunun Düzey1'de elde edilen PSNR grafiği



Çizelge 4.6. "coastguard" videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri

"coastguard"						
bpp1	bpp2	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	1	8 : 1	32,6047	24,1047	24,8111	33,1602
1	0,75	9,09 : 1	31,8618	23,5350	24,5020	31,9920
1	0,5	10,64 : 1	30,9295	22,8152	24,1544	30,5214
1	0,25	12,66 : 1	29,5803	21,7869	23,7632	28,7396
1	0,1	14,49 : 1	28,3052	20,6434	23,4559	27,4510

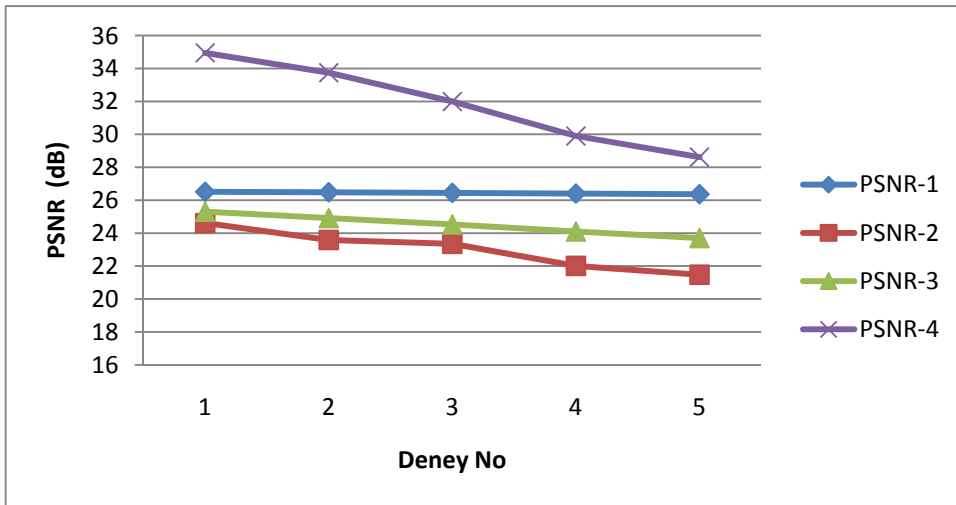
Çizelge 4.7. "coastguard" videosunun Düzey1'de elde edilen PSNR grafiği



Çizelge 4.8. "container" videosunun farklı bpp oranları ile blok karşılaştırma yöntemi kullanılarak kodlanması ile elde edilen PSNR değerleri

"container"						
bpp1	bpp2	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	1	8 : 1	26,5069	24,6091	25,3007	34,9491
1	0,75	9,09 : 1	26,4819	23,5859	24,9156	33,7414
1	0,5	10,64 : 1	26,4431	23,3499	24,5257	31,9907
1	0,25	12,66 : 1	26,4011	22,0109	24,0947	29,8967
1	0,1	14,49 : 1	26,3593	21,4721	23,7006	28,6175

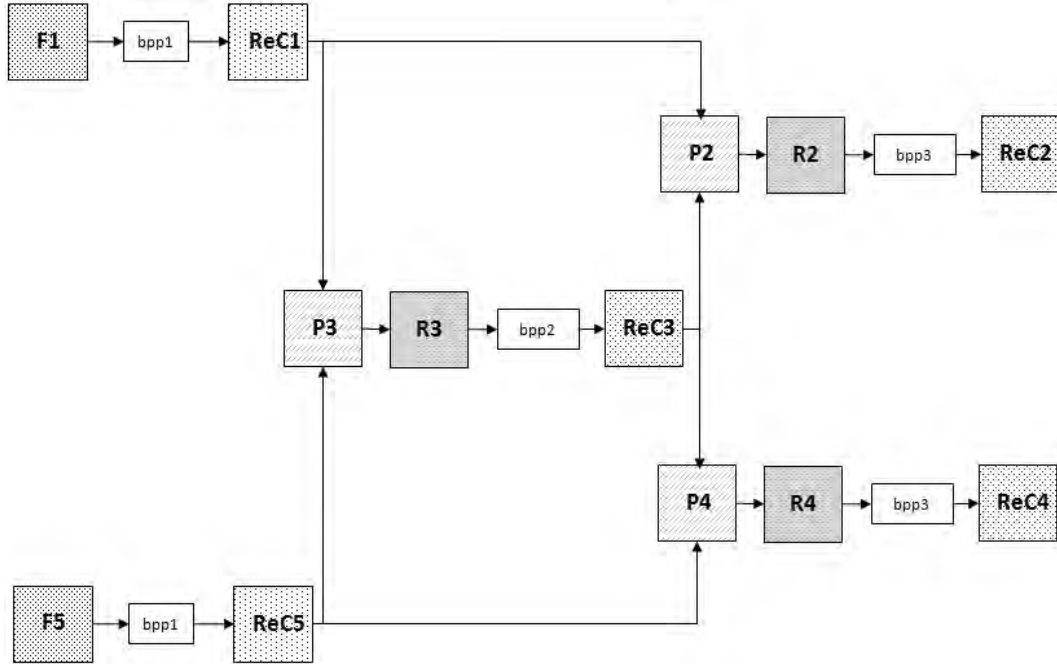
Çizelge 4.9. "container" videosunun Düzey1'de elde edilen PSNR grafiği



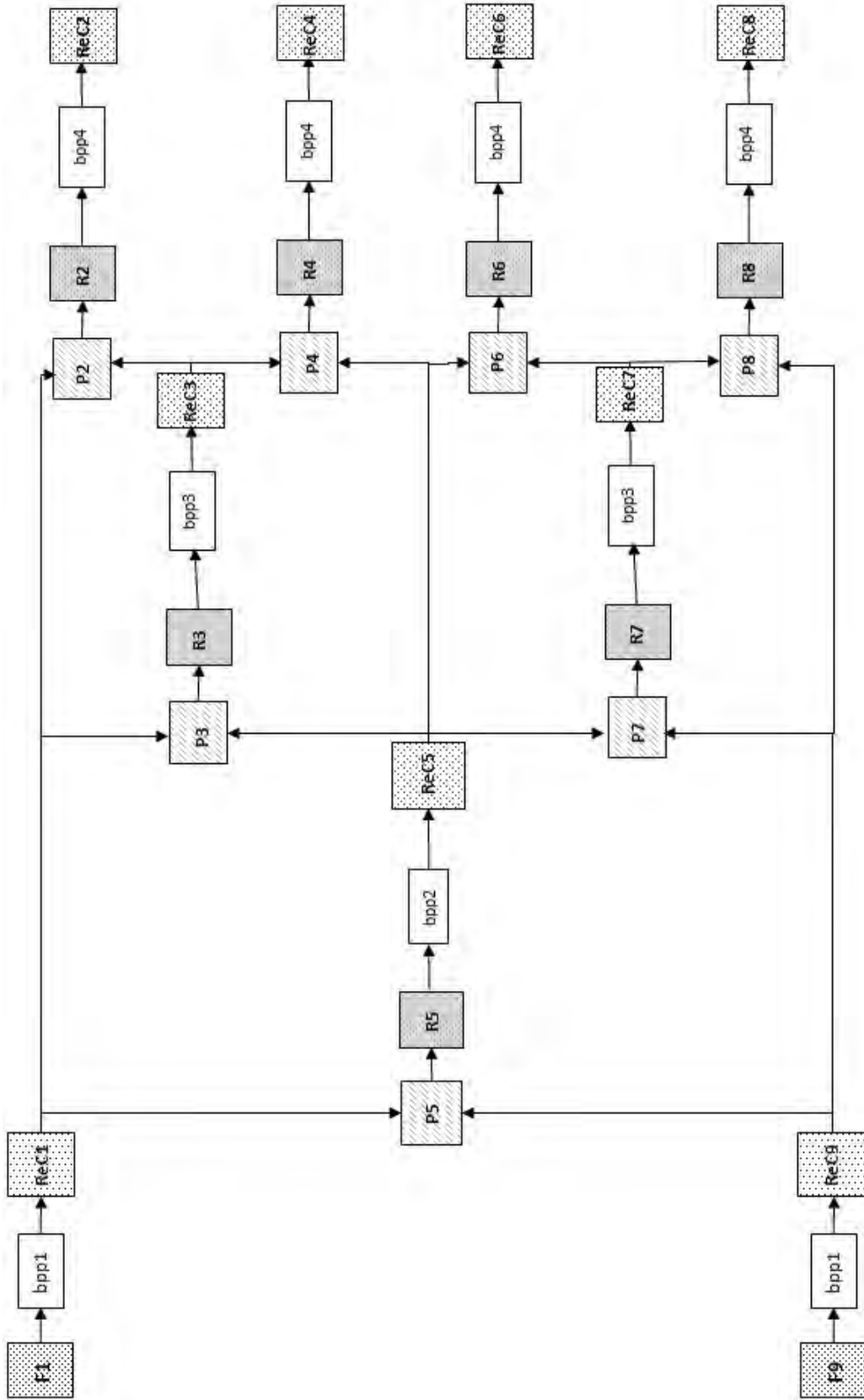
4.3. Hareket Dengelemede Farklı Düzeylerde Blok Karşılaştırma Yöntemi'nin Deneysel Sonuçları

SPIHT kodlayıcıda Düzey2 ayrışımı ve geri çatımında izlenen algoritma Şekil 4.7'de verilmiştir. $(4i - 3)$ ve $(4i + 1)$ indisli video imgeleri, "*bpp1*" piksel başına bit sayısı değeri ile kodlandıktan sonra geri çatılmaktadır. Bu noktada arada kalan $(4i - 1)$ indisli çerçeveyi tahmin etmek için bölüm 3.4.2'de anlatılan yöntem kullanılmakta ve $(4i - 1)$ indisli öngörüler (*P*) elde edilmektedir. Orijinal olan $(4i - 1)$ indisli çerçeveler ile öngörülerden elde edilen farklar (*R*) "*bpp2*" değeri ile kodlanarak geri çatılmış $(4i - 1)$ indisli çerçeveler (*ReC*) elde edilir. $(4i - 3)$ ve $(4i - 1)$ indisli geri çatılmış çerçevelerden aynı yöntem ile çift indisli çerçeveler elde edilir. Elde edilen çift indisli çerçevelerin orijinal çift indisli çerçevelerden farkı olan (*R*)'ler "*bpp3*" değeriyle kodlanıp geri çatılarak çift indisli (*ReC*) elde edilir.

Benzer algoritma Düzey3 için, Şekil 4.8'de F1-F9 çerçeveleri için gösterilmiş olduğu gibi $(8i - 1)$ ve $(8i + 1)$ indisli çerçeveler üzerinden ayrıştırma ve geri çatılım yapılır.



Şekil 4.7. Düzey2 ayrışımının ve geri çatılmasının F1 ve F5 çerçeveleriyle gösterimi



Şekil 4.8. Düzey3 ayrışımının ve geri çatılmasının F1 ve F9 çerçeveleriyle gösterimi

Her iki düzey için aynı deney videoları SPIHT ile farklı “*bpp*” değerleri ile kodlanarak CR, PSNR-1,PSNR-2 ve PSNR-3 değerleri elde edilmiştir. PSNR-2 değerleri ilgili satırdaki CR değerini kullanarak tüm adımların bu değer ile kodlanmasıyla elde edilirken, PSNR-3, tüm video çerçevelerinin ilgili satırdaki CR bit oranını kullanarak hiçbir yöntem kullanılmadan SPIHT ile kodlanmasıyla elde edilen değeridir. Önerilen yöntemlerin referans bir sıkıştırma yöntemi ile karşılaştırılması için MPEG2 kullanılmıştır. MPEG2, düzey2 için imge guruplarına IPBPI (Intra-Predicted-BiPredicted-Predicted-Intra) çerçeve örüntüsüyle, düzey3 için ise imge guruplarına IPBBBBBPI örüntüsüyle uygulanmıştır. Bu sıkıştırmalardan elde edilen PSNR-4 değerleri de tablolatırılmıştır.

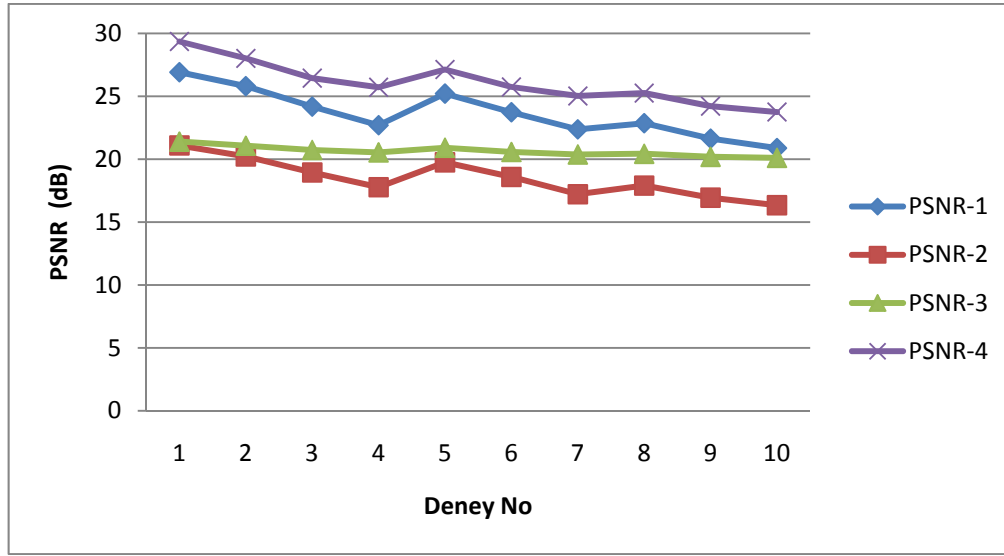
Düzyey2 üzerinde uygulanan hareket dengelemeli blok karşılaştırma yönteminde deney videolarındaki nesnelere hareketine göre PSNR değerleri her üç deney videosu için farklılık göstermiştir. Düzyey3 için ise, öngörüsü yapılan çerçeveye olan uzaklık bir adım daha arttığından ve geri çatılmış öngörüler üzerinden hareket dengelemeleri yapıldığından PSNR değerlerinde azalmalar görülmüştür. Buna paralel olarak, toplam sıkıştırılmış bit miktarında azalmalar vardır. Düzyey3 için elde edilen sonuçlar Çizelge 4.10, Çizelge 4.11 ve Çizelge 4.12’de gösterilmiştir.

Çizelge 4.10. “*bus*” videosunun farklı *bpp* oranları ile Düzyey2 ile elde edilen PSNR değerleri

“ <i>bus</i> ”							
bpp1	bpp2	bpp3	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	9,43 : 1	26,9128	21,0920	21,4002	29,3588
1	0,75	0,5	10,64 : 1	25,8047	20,2194	21,0737	28,0157
1	0,75	0,25	12,35 : 1	24,1815	18,9413	20,7348	26,4438
1	0,75	0,1	13,51 : 1	22,6978	17,7730	20,5461	25,7149

1	0,5	0,5	11,36 : 1	25,2140	19,7543	20,9123	27,1271
1	0,5	0,25	13,33 : 1	23,7326	18,5878	20,5772	25,7483
1	0,5	0,1	14,71 : 1	22,3677	17,2130	20,3784	25,0267
1	0,25	0,25	14,29 : 1	22,8604	17,9010	20,4328	25,2563
1	0,25	0,1	16,39 : 1	21,6373	16,9379	20,1926	24,2177
1	0,1	0,1	17,24 : 1	20,8808	16,3422	20,1113	23,7527

Çizelge 4.11. "bus" videosunun Düzey2'de elde edilen PSNR grafiği

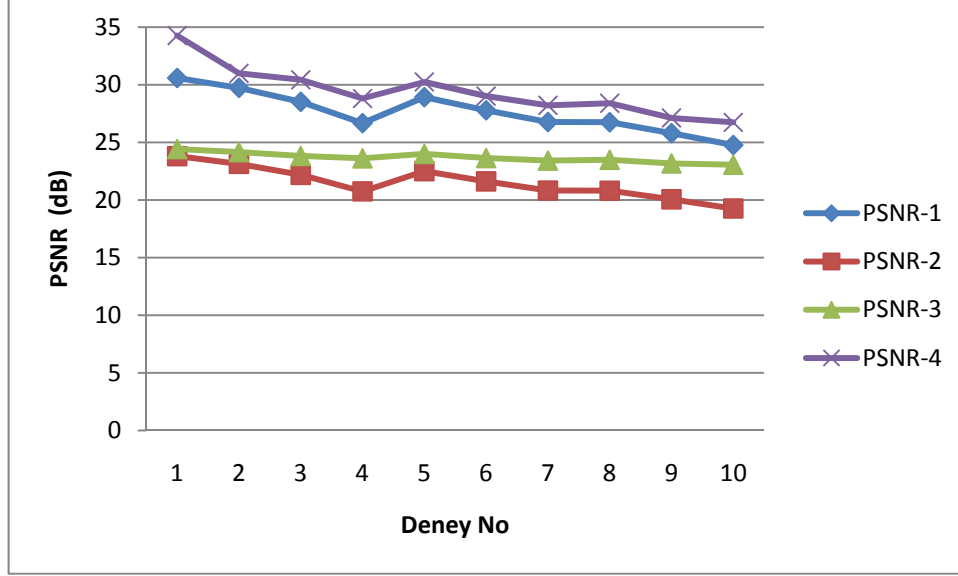


Çizelge 4.12. "coastguard" videosunun farklı bpp oranları ile Düzey2 ile elde edilen PSNR değerleri

"coastguard"							
bpp1	bpp2	bpp3	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	9,43 : 1	30,5897	23,8084	24,4182	34,2731
1	0,75	0,5	10,64 : 1	29,7338	23,1394	24,1544	30,9904
1	0,75	0,25	12,35 : 1	28,5237	22,1936	23,8211	30,4335
1	0,75	0,1	13,51 : 1	26,6787	20,7516	23,6168	28,8077
1	0,5	0,5	11,36 : 1	28,9215	22,5046	24,0118	30,2507
1	0,5	0,25	13,33 : 1	27,7849	21,6162	23,6462	29,0161
1	0,5	0,1	14,71 : 1	26,7687	20,8220	23,4226	28,2128
1	0,25	0,25	14,29 : 1	26,7513	20,8084	23,4877	28,3977
1	0,25	0,1	16,39 : 1	25,7997	20,0646	23,1741	27,1256

1	0,1	0,1	17,24 : 1	24,7752	19,2639	23,0605	26,7395
---	-----	-----	-----------	---------	---------	---------	---------

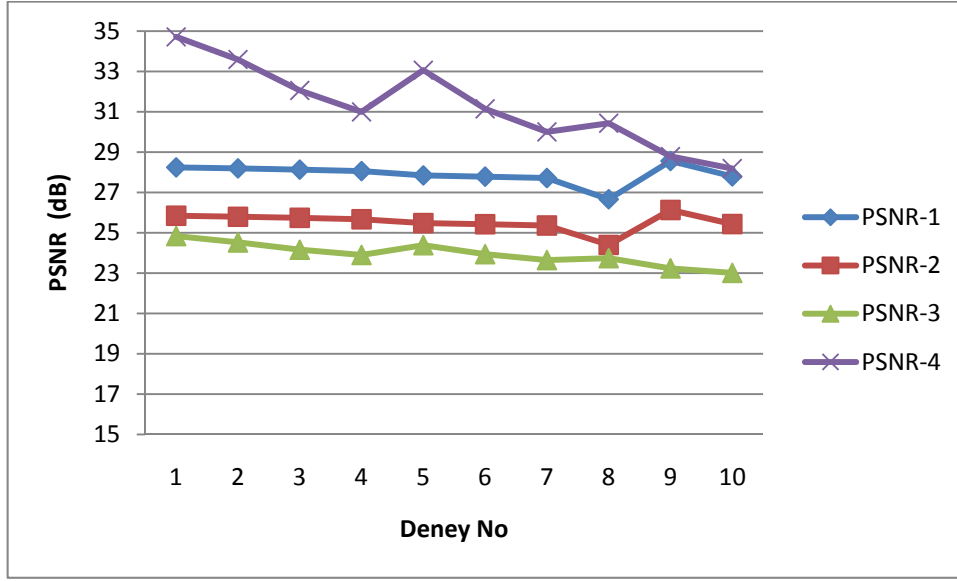
Çizelge 4.13. "coastguard" videosunun Düzey2'de elde edilen PSNR grafiği



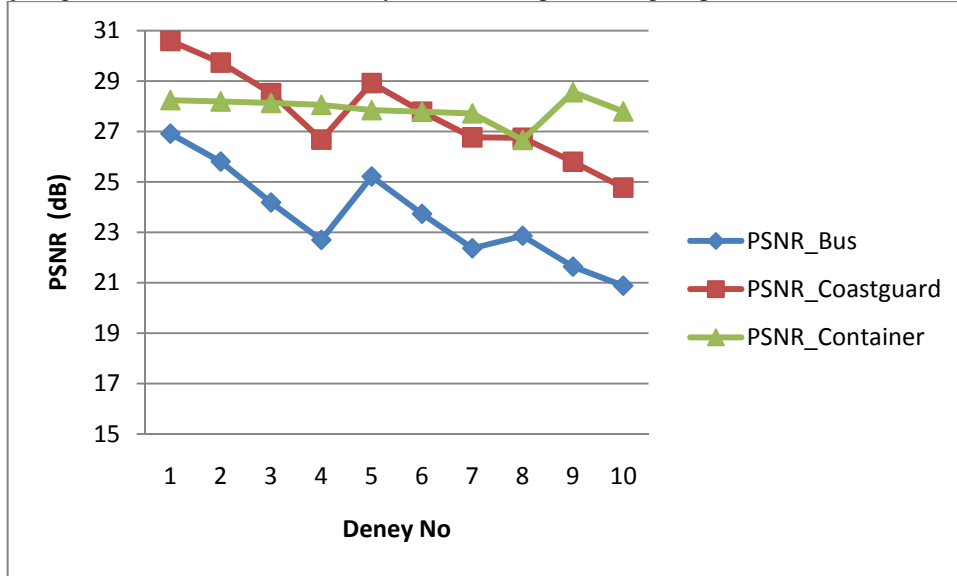
Çizelge 4.14. "container" videosunun farklı bpp oranları ile Düzey2 ile elde edilen PSNR değerleri

"container"							
bpp1	bpp2	bpp3	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	9,43 : 1	28,2444	25,8463	24,8423	34,7159
1	0,75	0,5	10,64 : 1	28,1921	25,7958	24,5257	33,5944
1	0,75	0,25	12,35 : 1	28,1334	25,7421	24,1632	32,0618
1	0,75	0,1	13,51 : 1	28,0582	25,6733	23,9030	30,9968
1	0,5	0,5	11,36 : 1	27,8456	25,4787	24,3898	33,0651
1	0,5	0,25	13,33 : 1	27,7860	25,4242	23,9422	31,1486
1	0,5	0,1	14,71 : 1	27,7165	25,3606	23,6492	30,0043
1	0,25	0,25	14,29 : 1	26,6654	24,3988	23,7448	30,4316
1	0,25	0,1	16,39 : 1	28,5594	26,1319	23,2386	28,7877
1	0,1	0,1	17,24 : 1	27,7962	25,4335	23,0153	28,1891

Çizelge 4.15. "container" videosunun Düzey2'de elde edilen PSNR grafiği



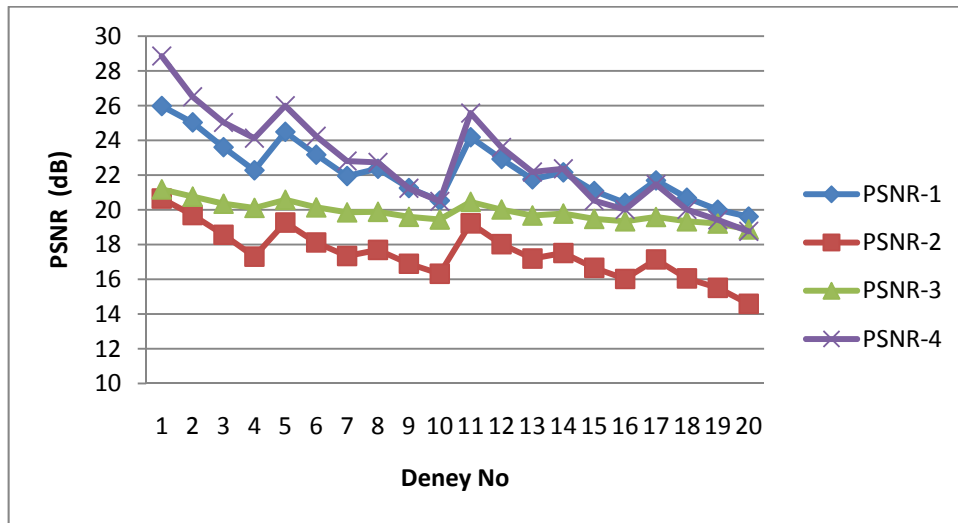
Çizelge 4.16. Tüm videoların Düzey2 PSNR-1 değerlerinin grafiği



Çizelge 4.17. "bus" videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri

"bus"								
bpp1	bpp2	bpp3	bpp4	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	0,75	10,20 : 1	25,9748	20,6485	21,1789	28,8567
1	0,75	0,75	0,5	12,20 : 1	25,0311	19,6983	20,7607	26,5114
1	0,75	0,75	0,25	14,93 : 1	23,6042	18,5640	20,3520	25,0336
1	0,75	0,75	0,1	17,24 : 1	22,2766	17,3086	20,1113	24,1274
1	0,75	0,5	0,5	13,33 : 1	24,4887	19,2671	20,5772	25,9953
1	0,75	0,5	0,25	16,95 : 1	23,1718	18,1203	20,1406	24,2410
1	0,75	0,5	0,1	20 : 1	21,9448	17,3449	19,8546	22,8117
1	0,75	0,25	0,25	19,61 : 1	22,3665	17,6901	19,8881	22,7358
1	0,75	0,25	0,1	23,81 : 1	21,2547	16,8963	19,5967	21,2388
1	0,75	0,1	0,1	27,03 : 1	20,5316	16,3215	19,4462	20,4911
1	0,5	0,5	0,5	14,09 : 1	24,1781	19,2202	20,461	25,5723
1	0,5	0,5	0,25	18,18 : 1	22,9253	18,0243	20,0188	23,5888
1	0,5	0,5	0,1	21,74 : 1	21,7545	17,1936	19,6748	22,1779
1	0,5	0,25	0,25	20,83 : 1	22,1567	17,5133	19,7869	22,3653
1	0,5	0,25	0,1	26,32 : 1	21,0872	16,6631	19,4767	20,5318
1	0,5	0,1	0,1	29,41 : 1	20,4055	16,0212	19,3408	20,0187
1	0,25	0,25	0,25	22,73 : 1	21,6921	17,1440	19,5879	21,4677
1	0,25	0,25	0,1	29,41 : 1	20,694	16,0506	19,3408	20,0187
1	0,25	0,1	0,1	33,33 : 1	20,0091	15,5061	19,2089	19,4255
1	0,1	0,1	0,1	37,04 : 1	19,6093	14,5883	18,8712	18,7613

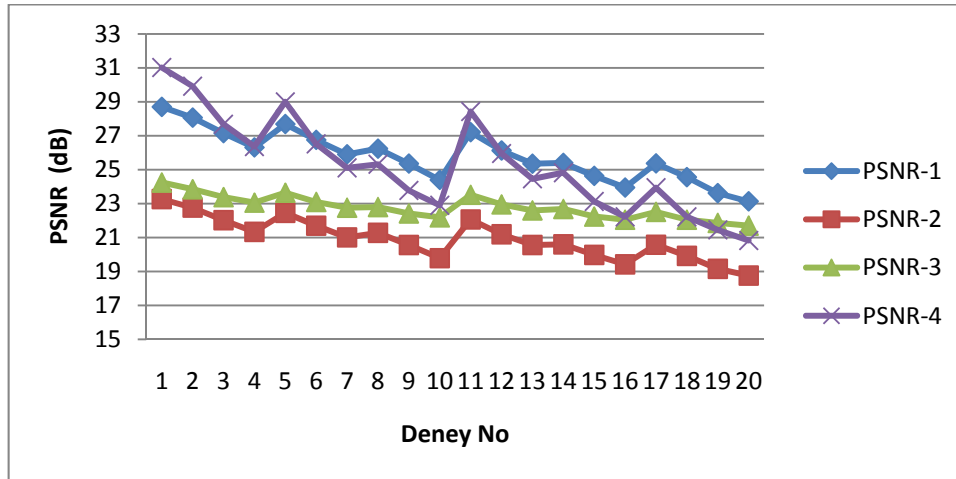
Çizelge 4.18. "bus" videosunun Düzey3'te elde edilen PSNR grafiği



Çizelge 4.19. "coastguard" videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri

"coastguard"								
bpp1	bpp2	bpp3	bpp4	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	0,75	10,20 : 1	28,7025	23,2701	24,2455	31,0248
1	0,75	0,75	0,5	12,20 : 1	28,0670	22,7623	23,8498	29,9213
1	0,75	0,75	0,25	14,93 : 1	27,1545	22,0223	23,3879	27,6801
1	0,75	0,75	0,1	17,24 : 1	26,3087	21,3364	23,0605	26,3633
1	0,75	0,5	0,5	13,33 : 1	27,6880	22,4550	23,6462	29,0004
1	0,75	0,5	0,25	16,95 : 1	26,7534	21,6970	23,099	26,5247
1	0,75	0,5	0,1	20 : 1	25,9025	21,0069	22,7618	25,1090
1	0,75	0,25	0,25	19,61 : 1	26,2384	21,2793	22,7988	25,3189
1	0,75	0,25	0,1	23,81 : 1	25,3549	20,5628	22,4220	23,7762
1	0,75	0,1	0,1	27,03 : 1	24,3928	19,7826	22,1921	22,8971
1	0,5	0,5	0,5	14,09 : 1	27,2072	22,0650	23,5216	28,4258
1	0,5	0,5	0,25	18,18 : 1	26,1283	21,1901	22,9547	25,9524
1	0,5	0,5	0,1	21,74 : 1	25,3498	20,5587	22,6009	24,4600
1	0,5	0,25	0,25	20,83 : 1	25,4000	20,5994	22,6852	24,8214
1	0,5	0,25	0,1	26,32 : 1	24,6312	19,9749	22,2388	23,1268
1	0,5	0,1	0,1	29,41 : 1	23,9444	19,4189	22,0532	22,2226
1	0,25	0,25	0,25	22,73 : 1	25,3697	20,5748	22,5143	23,9345
1	0,25	0,25	0,1	29,41 : 1	24,5587	19,9171	22,0532	22,2226
1	0,25	0,1	0,1	33,33 : 1	23,615	19,1518	21,8646	21,4501
1	0,1	0,1	0,1	37,04 : 1	23,1294	18,7579	21,7041	20,8253

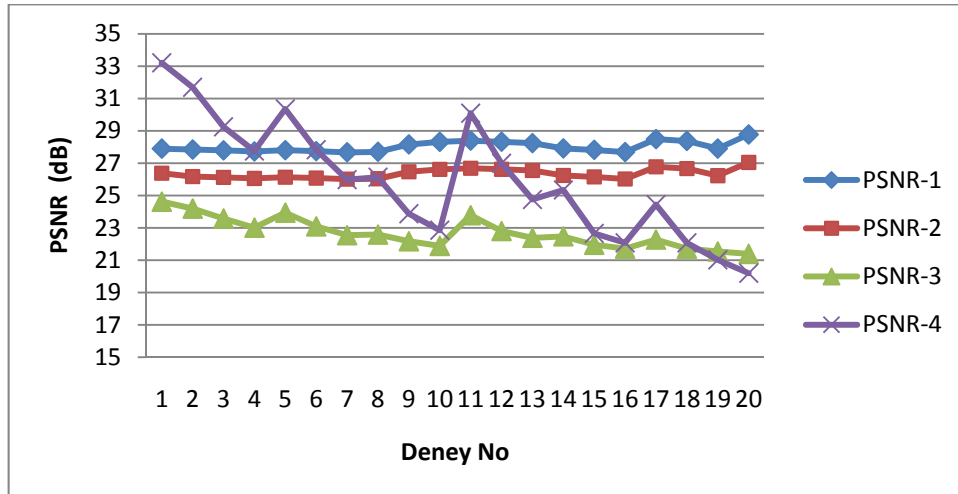
Çizelge 4.20. "coastguard" videosunun Düzey3'te elde edilen PSNR grafiği



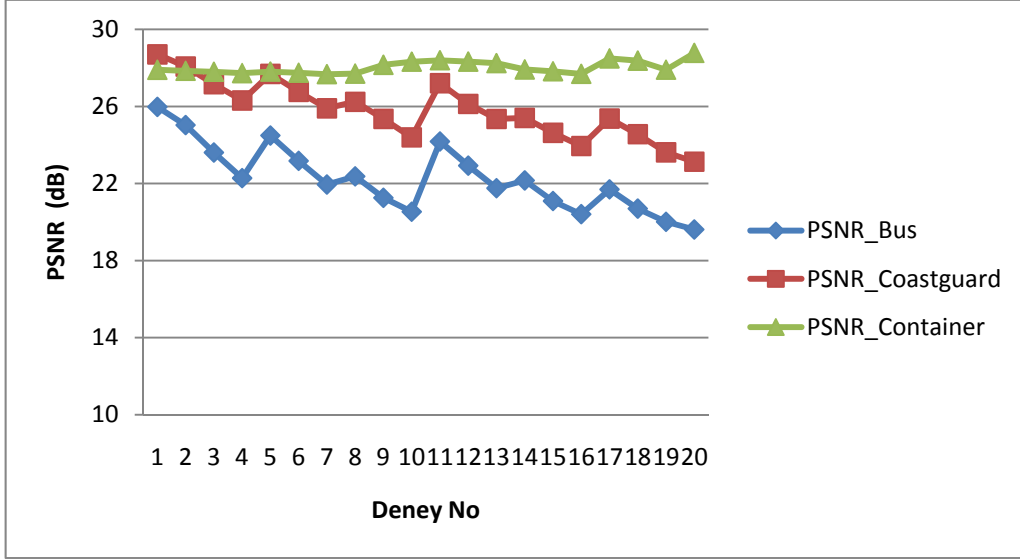
Çizelge 4.21. "container" videosunun farklı bpp oranları ile Düzey3 ile elde edilen PSNR değerleri

"container"								
bpp1	bpp2	bpp3	bpp4	CR	PSNR-1	PSNR-2	PSNR-3	PSNR-4
1	0,75	0,75	0,75	10,20 : 1	27,8997	26,3802	24,6253	33,2147
1	0,75	0,75	0,5	12,20 : 1	27,8443	26,1736	24,1979	31,7017
1	0,75	0,75	0,25	14,93 : 1	27,794	26,1264	23,5824	29,2523
1	0,75	0,75	0,1	17,24 : 1	27,7253	26,0618	23,0153	27,7541
1	0,75	0,5	0,5	13,33 : 1	27,8039	26,1357	23,9422	30,3517
1	0,75	0,5	0,25	16,95 : 1	27,7455	26,0808	23,0968	27,8388
1	0,75	0,5	0,1	20 : 1	27,6733	26,0129	22,5466	25,9876
1	0,75	0,25	0,25	19,61 : 1	27,7023	26,0402	22,5888	26,1358
1	0,75	0,25	0,1	23,81 : 1	28,1589	26,4694	22,1722	23,8795
1	0,75	0,1	0,1	27,03 : 1	28,3213	26,6220	21,8889	22,8557
1	0,5	0,5	0,5	14,09 : 1	28,3953	26,6916	23,7855	30,0971
1	0,5	0,5	0,25	18,18 : 1	28,3254	26,6259	22,8052	26,9991
1	0,5	0,5	0,1	21,74 : 1	28,2434	26,5488	22,3870	24,7575
1	0,5	0,25	0,25	20,83 : 1	27,9158	26,2409	22,4757	25,3469
1	0,5	0,25	0,1	26,32 : 1	27,8222	26,1529	21,9542	22,6629
1	0,5	0,1	0,1	29,41 : 1	27,689	26,0277	21,7189	22,0748
1	0,25	0,25	0,25	22,73 : 1	28,4889	26,7796	22,2779	24,4395
1	0,25	0,25	0,1	29,41 : 1	28,3751	26,6726	21,7189	22,0748
1	0,25	0,1	0,1	33,33 : 1	27,9028	26,2286	21,5389	21,0243
1	0,1	0,1	0,1	37,04 : 1	28,776	27,0494	21,389	20,1973

Çizelge 4.22. "container" videosunun Düzey3'te elde edilen PSNR grafiği



Çizelge 4.23. Tüm videoların Düzey3 PSNR-1 değerlerinin grafiği



5. SONUÇLAR

Tez çalışmasındaki ana motivasyon, günlük yaşamda video görüntülerinin her türlü mobil aygıtlar üzerinden paylaşımının yaygınlaşması yönündeki ilerlemelerdir. Bu amaçla iletimi yapılacak videonun kalite kaybını minimize edecek, öte yandan iletimini daha kolay kılacak düşük boyutta videoların elde edilmesi için videonun sıkıştırılması önemli hale gelmektedir.

Bu tezde video görüntülerinin sıkıştırılması için hareket dengelemeli ve yükseltme tabanlı hibrit bir video kodlayıcı tasarlanmıştır. Video kodlayıcı tasarlanırken, görüntü kalitesinden ödün vermeden sıkıştırma oranının artırılması hedeflenmiştir. Bu amaçla, video çerçeveleri, bilinen yükseltme adımlarıyla zamansal doğrultuda gruplara ayrılmıştır. Her grup içerisinde belli bazı çerçeveler içsel çerçeveler olarak alınırken diğer çerçeveler bunlardan öngörülmeyle çalışılmıştır. Öngörü için içsel çerçeveler arasında kenar-uyarlamalı yükseltme ile blok karşılaştırma yöntemi, üç farklı düzeyde kullanılmıştır. Her bir yöntemde öngörüsü yapılan çerçevelerin orijinal çerçevelerden farkları hesaplanarak bunların kodlanması, hedeflenen bit oranının belirlenebildiği EZW dönüşümlerini kullanan SPIHT kodlayıcı ile yapılmıştır. Kodlama sonunda, hedeflenen bit oranlarında elde edilen geri çatılmış çerçevelerin sinyal gürültü seviyeleri ile farklı durumlardaki sinyal gürültü seviyeleri karşılaştırılmıştır.

İlk önce video görüntüleri tek indisli ve çift indisli çerçeveler halinde gruplandırılmıştır. Video dizisindeki çift indisli çerçeveler tek indisli çerçevelerden tahmin edilmeye çalışılmıştır. Bunun için ardışık tek indisli çerçevelerde 3×3 boyutundaki bloklar, belirli bir arama komşuluğunda taranarak kenar-uyarlamalı yükseltme yöntemini kullanarak çift indisli çerçeveler tahmin edilmiştir. Doğrudan yükseltme yönteminde çift indisli çerçeveler, komşu tek indisli çerçevelerin ortalaması ile elde edilmektedir. Kenar-uyarlamalı yükseltme yönteminin PSNR değerleri, doğrudan yükseltme yönteminin değerleri ile karşılaştırılmıştır. Deneysel sonuçlar, kenar-uyarlamalı yöntemin daha başarılı olduğunu göstermiştir. Tek indisli çerçevelerde bulunan blokların zıt yönde hareket ettirilerek eşleştirilmesine dayalı blok karşılaştırma yöntemi ile OKH değerlerinin daha da iyileştirilmesi sağlanmıştır. Bunun yanında blok

karşılaştırma yöntemi 3×3 , 5×5 , 7×7 , v.b. gibi farklı boyutlardaki bloklar üzerinde denenmiştir. “bus” deney videosunda yatay doğrultuda hızlı hareket eden otobüs görüntüsünün hareketinin 3×3 boyutlu blok ile kodlanması daha iyi sonuçlar verirken, “coastguard” ve “container” gibi daha yavaş nesne hareketlerinin olduğu videolarda daha büyük boyutlu bloklarla öngörü işleminin yapılması daha iyi sonuçlar vermiştir. Kodlama aşamasında tek indisli çerçeveler önceden belirlenen bir bit/piksel oranıyla kodlanıp geri çatıldıktan sonra elde edilen çerçevelerle hareket dengelemeli blok karşılaştırma yöntemi kullanılarak çift indisli çerçeveler öngörülmüştür. Öngörülen çerçevelerle orijinalleri arasındaki farklar, başka bir bit/piksel oranıyla ile kodlanıp geri çatılmıştır. Geri çatılan video ile orijinal video arasındaki PSNR değerleri hesaplanmış ve bu işlem içsel çerçevelerle öngörülen çerçeveler için çeşitli bit/piksel oranları kullanılarak tekrarlanmıştır. İçsel ve öngörü çerçeveleri için farklı bit/piksel oranları kullanıldığında elde edilen ortalama bit/piksel oranı kullanılarak ve aynı öngörü işlemi kullanılarak ikinci bir PSNR değeri hesaplanmıştır. Hiçbir öngörü işlemi yapılmadan ortalama bit/piksel oranıyla bütün çerçeveler kodlanarak üçüncü bir PSNR değeri hesaplanmıştır. Çalışmaların daha önce yapılmış olan referans sıkıştırma yöntemlerinden MPEG2 ile karşılaştırılmasının sağlanması için dördüncü bir PSNR değeri hesaplanmış ve öncekilerle karşılaştırılmıştır. Elde edilen ilk üç PSNR değerlerine bakıldığında blok karşılaştırmalı yöntemin diğerlerine oranla daha başarılı olduğu görülmüştür. MPEG2 sıkıştırmasıyla yapılan karşılaştırmada ise, sıkıştırma arttığında blok karşılaştırma yönteminin başarımının, özellikle ikinci ve üçüncü düzeylerde videodaki nesnelerin hareketlerinin az olduğu durumlarda daha iyi olduğu görülmüştür.

Yapılan deneylerden sonra, blok karşılaştırma yöntemi farklı düzeylerde denenmiştir. Düzeyler arttıkça daha uzaktaki çerçeveler kullanılarak ortada kalan çerçeveler öngörülme çalışılmıştır. İkinci düzeyde PSNR değerleri bir alt ayrışımaya göre daha küçük çıkmış olmasına rağmen kodlamadan elde edilen kazançta göre bu fark kabul edilir düzeyde kalmıştır. Üçüncü düzey ayrışımında “container” videosu için optimum sıkıştırma, 37,04 : 1 oranında 29 dB’lik PSNR değeriyle bulunmuştur. Diğer deney videoları için üçüncü düzeyde elde edilen PSNR değerleri, alt düzeylerinkine oranla daha düşük olsa da daha fazla

sıkıştırma elde edilebilmiştir. Elde edilen bütün düzeylerdeki PSNR değerlerine göre, spor müsabakaları gibi nesnelerin hareketlerinin çok ve hızlı olduğu videolarda düşük düzeyde blok karşılaştırma yönteminin daha başarılı olacağı; diğer durumlarda ise yüksek düzeylerde blok karşılaştırma yöntemi ile öngörü yapılmasının daha uygun olduğu görülmüştür.

İleriki çalışmalar olarak video çerçevelerinin SPIHT yerine JPEG ve JPEG2000 ile sıkıştırılarak aynı deneylerin tekrarlanması planlanmaktadır. Ayrıca video görüntülerinde hareket eden nesnelerin hızlarını göz önünde bulundurarak kullanılacak düzeyi adaptif olarak belirleyen hareket dengelemeli blok karşılaştırma yöntemlerinin geliştirilmesi düşünülmektedir.

KAYNAKLAR

- [1] Manning, C.E., <http://newmediarepublic.com/dvideo/>, 2004.
- [2] Gerek, Ö.N. ve Çetin, A.E., "Adaptive Polyphase Subband Decomposition Structures for Image Compression," *Proceedings of the IEEE Transactions on Image Processing*, **9**, no. 10, pp. 1649-1660, Ekim 2006.
- [3] Gerek, Ö.N. ve Çetin, A.E., "A 2-D Orientation-Adaptive Prediction Filter in Lifting Structures For Image Coding," *Proceedings of the IEEE Transactions on Image Processing*, **15**, no. 1, pp. 106-108, Ocak 2006.
- [4] Sayood, K., *Introduction to Data Compression*, San Francisco, A.B.D., 2006.
- [5] Anonim, *MPEG Committee*, 2004. <http://www.mpeg.org>.
- [6] Aksay, A., *Motion Wavelet Video Compression*. Elektrik-Elektronik Mühendisliği Böl. ODTÜ, Ankara, Yüksek Lisans Tezi, Ocak 2001.
- [7] Karlsson, G. ve Vetterli, M., "Three-Dimensional Subband Coding of Video," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, New York, A.B.D., Nisan 1998, pp. 1100-1103.
- [8] Taubman, D. ve Zakhor, A., "Multirate 3-D Subband Coding of Video," *IEEE Trans. Image Processing*, **3**, pp. 572-588, Eylül 1994.
- [9] Ohm, J.-R., "Three Dimensional Subband Coding with Motion Compensation," *IEEE Trans. Image Processing*, **3**, no. 5, pp. 559-571, Eylül 1994.
- [10] Woods, W. ve Choi, S.J., "Motion Compensated 3-D Subband Coding of Video," *IEEE Trans. Image Processing*, **8**, no. 2, pp. 155-167, Şubat 1999.
- [11] Secker, A. ve Taubman, D., "Motion-Compensated Highly Scalable Video Compression using an Adaptive 3D Wavelet Transform based on Lifting," *Proceedings of the International Conference on Image Processing*, **2**, pp. 1029-1032, Ekim 2001.
- [12] Biçer, A., *Motion Compensated Three Dimensional Wavelet Transform Based Video Compression and Coding*. Elektrik-Elektronik Mühendisliği Böl. ODTÜ, Ankara, Yüksek Lisans Tezi, Ocak 2006.

- [13] Anonim, *en.wikipedia.org/wiki/YUV*, Haziran 2010.
- [14] Akyol, E., *Adaptive Scalable Video Coding*, Koç Üniversitesi, Yüksek Lisans Tezi, Eylül 2005.
- [15] Barnett, B., *Handbook of Image and Video Processing*, 2nd ed., Elsevier Inc., A.B.D., 2008.
- [16] Tekalp, A.M., *Digital Video Processing*, Prentice Hall, New Jersey, A.B.D., 1995.
- [17] Vetterli, M. ve Kovacevic, J., *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, A.B.D., 1995.
- [18] Shapiro, J.M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, **41**, no. 12, pp. 3445-3462, 1993.
- [19] Chia, W.C., Ang, Lii-M. ve Seng, K.P., "Embedded Descendent-Only Zerotree Wavelet Coding for Image Compression," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, **1**, pp. 922-926, Mart 2009.
- [20] Chen, C. ve Pearlman, A., "Three-Dimensional Subband Coding of Video using the Zero-Tree Method," *Cisual Communications and Image Processing*, pp. 1302-1312, Ocak 1996.
- [21] Campisi, P., Gentile, M. ve Neri, A., "Three Dimensional Wavelet Based Approach for a Scalable Video Conference System," *Proceedings of the International Conference on Image Processing*, **3**, pp. 802-806, Ekim 1999.
- [22] Limb, B. ve Haskell, G., "Predictive Video Encoding Using Measured Subjective Velocity," Patent No: 3,632,865, Ocak 1972.
- [23] Koga, T. ve ark., "Motion compensated interframe coding for video conferencing," *NTC'81*, New Orleans, LA, Nov. 1981, pp. G5.3.1-G5.3.5.
- [24] Robbins, A.N. ve Netravali, J.D., "Motion-Compensated Television Coding: Part I," *Bell Syst. Tech. J.*, no. 58, pp. 631-670, Mart 1979.

- [25] Jain, J.R. ve Jain, A.K., "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Trans. Commun.*, no. COM-29, pp. 1799-1808, Aralık 1981.
- [26] Netravali, A.N. ve Robbins, J.D., "Motion Compensated Television Coding: Part I," *Bell Syst. Tech. J.*, **58**, pp. 631-670, Mart 1979.
- [27] Sweldens, W., "The Lifting Scheme: A Construction of Second Generation Wavelets.," *SIAM Journal on Mathematical Analysis*, **29**, no. 2, pp. 511-546, 1998.
- [28] Payan, F., Boulfani, Y. ve Antonini, M., "Temporal Lifting Scheme for the Compression of Animated Sequences of Meshes," *Proceeding of IEEE International Workshop VLBV 2005*, Sardinia, İtalya, Eylül 2005.
- [29] Görgülü, S., Telçeken, S. ve Gerek, Ö.N., "Lifting Based Video Coding Using Motion Compensation," *IKECCO'2008 5. International Conference on Electronics & Computer*, Bişkek, Kırgızistan, 2008, pp. 96-100.
- [30] Telçeken, S., Görgülü, Ş. ve Gerek, Ö.N., "Hareket Dengeleme Katkılı 3B Yükseltmeye Dayalı Video Kodlayıcı," *IEEE 17. Sinyal İşleme ve İletişim Kurultayı 2009 (SIU2009)*, Side, Antalya, 2009.
- [31] Telçeken, S., Görgülü, Ş. ve Gerek, Ö.N., "A 3D Lifting Based Method Augmented by Motion Compensation for Video Coding," *9th International Conference on Intelligent Systems Design and Applications*, Pisa, İtalya, 2009.
- [32] Kim, B-Jo., Xiong, Z. ve A., PearlmanW., "Low Bit-Rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees (3-D SPIHT)," *IEEE Trans. Circuit Syst Video Tech.*, **10**, pp. 1374-1387, Aralık 2000.
- [33] *MPEG-2 Video*. ITU-T Recommendation H.262-ISO/IEC 13818-2, Ocak 1995.
- [34] *Video Coding for Low Bitrate Communication*. ITU-T Recommendation H.263, Aralık 1995.
- [35] Committee, JPEG, *JPEG2000-Links*. <http://www.jpeg.org>, 2004.

- [36] Gall, D.L., "MPEG: A video Compression Standard for Multimedia Applications," *Commun. ACM*, **34**, no. 4, pp. 47-58, Nisan 1991.
- [37] Ohm, J.-R., "Advances in Scalable Video Coding," *Proceedings of the IEEE*, **93**, no. 1, pp. 42-56, Ocak 2005.
- [38] Pau, G., Tillier, C., Pesquet-Popescu, B. ve Heijmans, H., "Motion Compensation and Scalability in Lifting-Based Video Coding," *Signal Processing: Image Communication*, **19**, no. 7, pp. 577-600, Ağustos 2004.
- [39] Pau, G., Tillier, C. ve Pesquet-Popescu, B., "Optimization of the Predict Operator in Lifting-Based Motion Compensated Temporal Filtering," *Visual Communications and Image Processing*, **5308**, pp. 712-720, Ocak 2004.
- [40] Taubman, E., "Successive Refinement of Video: Fundamental Issues, Past Efforts and New Directions," *Visual Communications and Image Processing*, Lugano, İsviçre, Temmuz 2003, pp. 649-663.
- [41] Pesquet-Popescu, B. ve Bottreau, V., "Three-Dimensional Lifting Schemes for Motion Compensated Video Compression," *Speech and Signal Processing*, **3**, pp. 1793-1796, Mayıs 2001.

Ek-1 RGB ↔ YUV Renk Uzayı Dönüşümü

RGB ana renk uzayı olarak kabul edilir. Bir renk kırmızı, yeşil ve mavi renk bileşenlerinden çeşitli değerlerde kullanılarak elde edilmektedir. YUV renk uzayı Avrupa'daki televizyonlarda kullanılmaktadır. Y, parlaklık bileşenini, U ve V ise renk bileşenini göstermektedir. 1. denklem RGB uzayından YUV'ye dönüşümü, 2. denklem ise YUV uzayından RGB'ye dönüşümü göstermektedir.

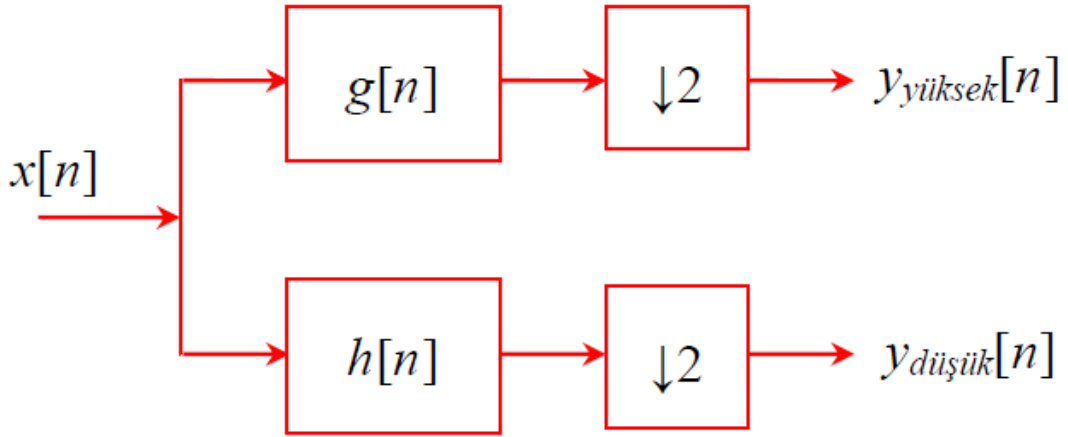
$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.14 \\ 1 & -0.396 & 0.581 \\ 1 & 2.029 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Ek-2 Ayrık Dalgacık Dönüşümü

Ayrık Dalgacık Analizi / Ayrışımı

DWT, sinyali detaylı bilgiye ayrışımını yaparak farklı frekans bandında farklı çözünürlüklerde çözümlemesini yapar. Şekil A.1’de $x[n]$ sinyalinin farklı frekans bantlarına 1-Düzye ayrışımı gösterilmektedir.

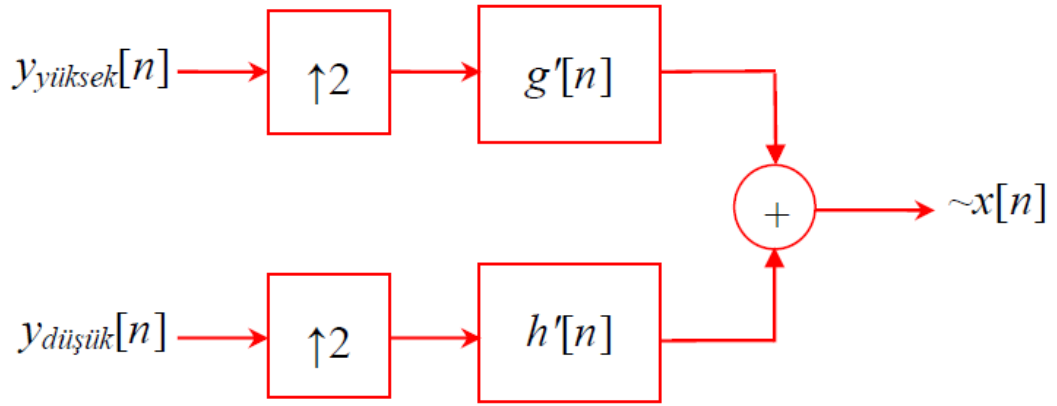


Şekil A.1. 1-Düzye Dalgacık Analizi

Şekil A.1’de $x[n]$ orijinal sinyali, sırasıyla $g[n]$ ve $h[n]$ olmak üzere yüksek geçiren ve alçak geçiren filtrelerden geçirildikten sonra 2 ile aşağı örneklenir. $x[n]$ sinyalinin ayrışımı sonucunda $y_{yüksek}[n]$ ve $y_{düşük}[n]$ dalgacık katsayıları çıktı olarak elde edilmektedir. Bu katsayıları matematiksel olarak aşağıdaki gibi gösterebiliriz:

$$y_{yüksek}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[2n - k]$$
$$y_{düşük}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[2n - k]$$

Geri çatmada ise DWT katsayıları 2 ile yukarı örneklenip, $g'[n]$ ve $h'[n]$ filtrelerinden geçirilerek sentezlenir. Bu filtrelerin çıktıları toplanarak $\tilde{x}[n]$ sinyali elde edilir. İşlemler Şekil A.2’de gösterilmiştir.

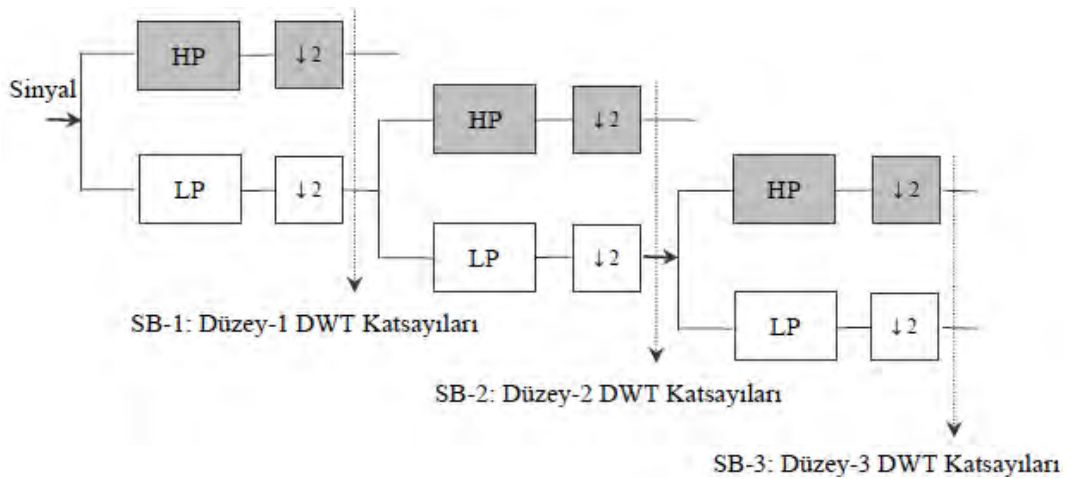


Şekil A.2. 1-Düzye Dalgacık Sentezi

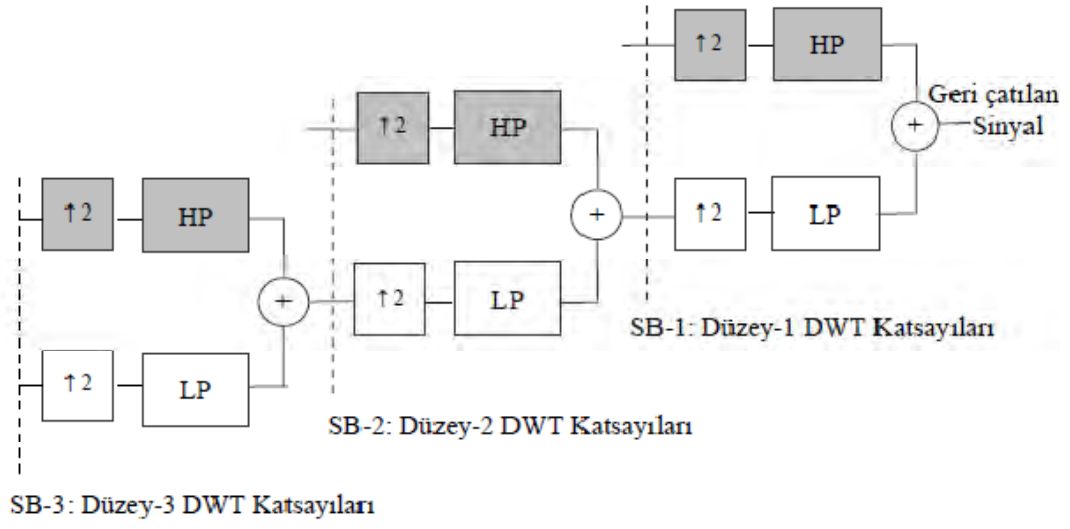
Geri çatma formülü ise aşağıdaki gibi olacaktır:

$$\tilde{x}[n] = \sum_{k=-\infty}^{\infty} [(y_{yüksek}[k] \cdot g'[-n + 2k]) + (y_{düşük}[k] \cdot h'[-n + 2k])]$$

Şekil A.1 ve Şekil A.2 de gösterilen 1-Düzye çözümlemesi ardarda iki kez yapılırsa 2-Düzye ve 3-Düzye dalgacık katsayıları elde edilir. 3-Düzye katsayılarından sentezleme yapılarak sinyal yeniden geri çatılır. Ayrıntılı gösterim Şekil A.3 ve Şekil A.4'te verilmiştir.



Şekil A.3 3-Düzye Dalgacık Çözümleme



Şekil A.4 3-Düzey Dalgacık Sentezleme