

GÖRÜNTÜ DİZİLERİNDEKİ ARTIKLIKLARIN İŞLENMESİ

Mustafa ATALAR

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MART 2008

ANKARA

Mustafa ATALAR tarafından hazırlanan GÖRÜNTÜ DİZİLERİNDEKİ ARTIKLIKLARIN İŞLENMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hasan Şakir BİLGE
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. İrfan KARAGÖZ
(Elektrik ve Elektronik Mühendisliği, Gazi Üniversitesi)

Yrd. Doç. Dr. Hasan Şakir BİLGE
(Bilgisayar Mühendisliği, Gazi Üniversitesi)

Yrd. Doç. Dr. Reza HASSANPOUR
(Bilgisayar Mühendisliği, Çankaya Üniversitesi)

Tarih: 24/03/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nermin ERTAN
Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Mustafa ATALAR

GÖRÜNTÜ DİZİLERİNDEKİ ARTIKLIKLARIN İŞLENMESİ

(Yüksek Lisans Tezi)

Mustafa ATALAR

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mart 2008

ÖZET

Anlaşılabilir kalitede görüntü saklanabilmesi için sayısal hafıza üzerinde geniş alana, gerçek zamanlı aktarımı için, yüksek hat kapasiteli ağlara ihtiyaç duyulmaktadır. Saklama ve taşıma ortamlarının maliyetlerinin yüksek olmasından dolayı görüntü sıkıştırma yöntemleri önem taşımaktadır. Görüntü sıkıştırma işleminde, sıkıştırma oranı görüntü kalitesiyle ters orantılıdır. Farklı sıkıştırma yöntemlerinde, bu iki parametrenin eniyilenmesi amaçlanmaktadır. Görüntü kalitesi, sinyaldeki değişimlerinin ayırt edilebilmesine bağlıdır. Bu yüzden sıkıştırma işlemi, sinyalin frekans düzleminde yapılmaktadır. Bu çalışmada görüntü ile görüntünün Ayrık Kosinüs Dönüşümü (AKD) ve 3 Boyutlu (3B) AKD'si arasındaki ilişki araştırılmış, 3B AKD ile video sıkıştırma uygulamalarında alt işlemler ve bunların sıkıştırma oranı ve kalite üzerindeki etkilerini farklılaştırabilmek için yeni yöntemler incelenmiştir. Bu yöntemlerin en önemlisi, hareket tanımaya dayalı sıkıştırma yöntemidir. Özellikle yaygın kullanılan dosya biçimleri üzerinde daha yüksek performans ile hareket tanıma uygulayabilmek için, frekans düzleminde hareket tanıma yöntemleri incelenmiştir. Sözde faz hesaplaması ile hareket tanıma yönteminin başarısını artırabilmek için geliştirme yapılmıştır. Yapılan incelemelerin doğrulanması amacıyla, Matlab'da 3B AKD video sıkıştırma uygulaması hazırlanarak farklı çerçeve sayıları, budama değerleri ile tüm görüntü üzerinde işlem yapılarak incelenmiştir. Ayrıca tüm görüntü küçük bloklara bölünerek sıkıştırma işlemi uygulanmıştır. İçeriklerindeki değişim ve hareket hızı farklı iki video görüntüsü

üzerinde yapılan denemelerde, her iki video görüntüsü aynı sayıda AKD değeri ile ifade edildiğinde yavaş hareketlere sahip videodaki bozulmanın daha az olduğu saptanmıştır. Bu sonuçtan, enerji sıkıştırmanın ve hareket tanınmanın önemi ortaya konulmuştur.

Bilim Kodu : 902.1.067

Anahtar Kelimeler : Ayrık Kosinüs Dönüşümü, 3 Boyutlu Ayrık Kosinüs Dönüşümü, video sıkıştırma, hareket tanıma, sözde faz hesaplaması

Sayfa Adedi : 92

Tez Yöneticisi : Yrd. Doç. Dr. Hasan Şakir BİLGE

PROCESSING OF REDUNDANCY IN IMAGE SEQUENCES**(M.Sc. Thesis)****Mustafa ATALAR****GAZİ UNIVERSITY****INSTITUTE OF SCIENCE AND TECHNOLOGY****March 2008****ABSTRACT**

High amount of memory is needed to save images at understandable quality, and high bandwidth networks are needed to transfer images in real time. Image compression is important because memory and link capacity are expensive. Compression ratio is inversely proportional to image quality in compression process. It is targeted to optimize these two parameters at different studies. Image quality is related to differing frequencies of the signal. Therefore, compression is done in frequency space of signal. In this thesis, the relation between image and its Discrete Cosine Transform (DCT), 3 dimension (3D) DCT is studied and also subprocesses of 3D DCT compression are studied to differentiate the quality of image by new methods on these subprocesses. The most important method to do this is the motion estimation based coding. Specially motion estimation in frequency domain is researched to increase the compression effect while working with standard file formats. Some improvements are offered for motion estimation with pseudo phase calculation. For this purpose motion estimation is studied. To verify these researches, by preparing 3D compression application for Matlab, the results are experienced with different frame numbers and pruning values on full images. And by dividing full image into small blocks compression is done.

When two video sequences are analyzed, it is seen that the video image which has slower motions has less degradation at images. This result showed the importance of energy compression and motion estimation.

Science Code : 902.1.067
Key Words : Discrete Cosine Transform, 3 Dimension Discrete Cosine Transform, video compression, motion estimation, pseudo phase calculation
Page Number : 92
Adviser : Assist. Prof. Dr. Hasan Şakir BİLGE

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Hocam Hasan Őakir Bilge'ye, maddi ve manevi desteęiyle beni hiçbir zaman yalnız bırakmayan sevgili niőanlım Havva Çoban'a teőekkürü bir borç bilirim.

İÇİNDEKİLER**Sayfa**

ÖZET	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xvii
1. GİRİŞ	1
2. AYRIK KOSİNÜS DÖNÜŞÜMÜ (AKD)	5
2.1. AKD Hesaplama Yöntemleri	5
2.2 AKD Matrisi	7
2.3 Taban Görüntüler	11
2.4 Görüntü İşleme Uygulamalarında AKD	14
2.5 AKD Düzleminde Görüntü İşleme	16
2.6. AKD Özellikleri.....	21
2.6.1. Dekorelasyon (İlişkilerin azaltılması).....	21
2.6.2. Enerji sıkıştırma	22
2.6.3. Ayrılabilirlik.....	26
2.6.4. Simetri.....	26
2.6.5. Ortogonallik	27
3. 3B AKD İLE GÖRÜNTÜ SIKIŞTIRMA.....	28

Sayfa

3.1. Orijinal Video Görüntüsü	29
3.2. Orijinal Video Matrisi Üzerinde 3B AKD Hesaplanması	29
3.3. 3B AKD Katsayılarının Nicemlenmesi	31
3.4. 3B AKD Katsayılarının Sıralanması.....	35
3.5. Sıkıştırma	36
4. HAREKET TANIMA.....	39
4.1. MPEG	40
4.2. Blok Eşleştirme Algoritması (BEA).....	42
4.2.1 Tüm arama algoritması	42
4.2.2 Logaritmik arama.....	44
4.3 AKD Düzleminde Hareket Tanıma	45
4.4 Sözde Faz Kullanarak Hareket Tanıma	49
4.4.1. 1B sinyaller için sözde faz hesaplama	50
4.4.2. 2B sinyaller için sözde faz hesaplama	54
4.4.3. Yapılan iyileştirmeler.....	58
5. UYGULANAN YÖNTEM.....	72
6. SONUÇ VE ÖNERİLER.....	85
KAYNAKLAR	88
EKLER.....	90
EK-1 Türkçe-İngilizce terimler karşılığı	91
ÖZGEÇMİŞ	92

ÇİZELGELERİN LİSTESİ

Çizelge		Sayfa
Çizelge 2.1.	Doğrudan Algoritma'nın sonuçlarının Geleneksel Algoritma'yla karşılaştırılması	9
Çizelge 2.2.	Pao ve Sun'un çalışmalarının sonuç tablosu	9
Çizelge 2.3.	Shen ve Sethi'nin AKD işlem tablosu	19
Çizelge 2.4.	Yatay piksel sayısı 128 olan üç farklı görüntünün enerji dağılımı/AKD katsayısı çizelgesi	26
Çizelge 3.1.	Budama işlemi sonuçları	33
Çizelge 3.2.	Lakhani'nin çalışmasının sonuçları	37
Çizelge 4.1.	Sözde faz ile hareket belirleme çizelgesi	57
Çizelge 5.1.	Çerçeve sayısı 3 alındığında uygulama sonuçları	73
Çizelge 5.2.	Çerçeve sayısı 5 alındığında uygulama sonuçları	73
Çizelge 5.3.	Çerçeve sayısı 12 alındığında uygulama sonuçları	74

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. 8x8'lik AKD matrisi.....	7
Şekil 2.2. Vektör tabanlı 3B AKD hesaplama sonuçları.....	10
Şekil 2.3. Taban görüntüler	11
Şekil 2.4. 8x8'lik görüntüde taban görüntü matrisi.....	12
Şekil 2.5. 8x8'lik görüntüde (5,2) elemanının taban görüntü grafiği.....	12
Şekil 2.6. 8x8'lik görüntüde (7,7) elemanının taban görüntü grafiği.....	13
Şekil 2.7. Farklı iki görüntü.....	14
Şekil 2.8. (a) Görüntünün sinyal gösterimi (b) a görüntüsünün AKD katsayılarının dağılımı.....	16
Şekil 2.9. AKD boyutunda görüntü döndürme.....	20
Şekil 2.10. (a) İçeriği ilişkisiz bir görüntünün, uzay düzleminde ve AKD sinyalinin normalize edilmiş otokorelasyon sonucu (b) İçeriği ilişkili bir görüntünün, uzay düzleminde ve AKD sinyalinin normalize edilmiş otokorelasyon sonucu	22
Şekil 2.11. (a) İçeriği ilişkisiz görüntünün AKD sinyali, (b) içeriği ilişkili görüntünün AKD sinyali	23
Şekil 3.1. 3B AKD sıkıştırma ve açma gösterimi	29
Şekil 3.2. AKD fonksiyonu enerjisinin yoğunlaşma bölgesi	30
Şekil 3.3. Bir resmin 2B AKD grafiği	30
Şekil 3.4. Video görüntüsünün küçük bloklara bölünmesi.....	31
Şekil 3.5. Video görüntüsünde çerçevelerin gruplanması	31
Şekil 3.6. MPEG için 8x8'lik nicemleme matrisi.....	32
Şekil 3.7. 8x8'lik matrisin budama değeri=3 olarak budanması.....	33
Şekil 3.8. 3B AKD fonksiyonu üzerinde $u+v+w=SABIT$ fonksiyonu yüzeyi.....	33
Şekil 3.9. MPEG'de kullanılan iki farklı sıralama matrisi	35

Şekil	Sayfa
Şekil 3.10. Huffman kodlama.....	36
Şekil 3.11. Run-length sıkıştırma	36
Şekil 4.1. MPEG’de görüntü tiplerinin dizilimi	41
Şekil 4.2. MPEG (a) sıkıştırma ve (b) çözme fonksiyonları akış şeması.....	42
Şekil 4.3. BEA tüm arama uzayı	43
Şekil 4.4. Logaritmik arama noktaları.....	44
Şekil 4.5. AKD fark alma ile hareket tanıma akış şeması.....	46
Şekil 4.6. AKD fark alma ile hareket tanıma (a) önceki görüntü (b) referans görüntü (c) DC katsayı farkı haritası (d) hareket görüntüsü	48
Şekil 4.7. Sözde faz ile hareket tanıma akış şeması	50
Şekil 4.8. (a) x1 sinyali, (b) x2 sinyali, (c) x1 sinyalinin AKD’si, (d) x2 sinyalinin AKD’si, (e) x1 sinyali üzerinde Koc ve Liu’nun yöntemiyle öteleme hesaplaması, (f) x2 sinyali üzerinde Koc ve Liu’nun yöntemiyle öteleme hesaplaması.....	52
Şekil 4.9. (a) x1 sinyali üzerinde DXT-ME sonucu (b) x2 sinyali üzerinde DXT-ME sonucu	53
Şekil 4.10. İki boyutlu sinyallere sözde faz uygulanması	54
Şekil 4.11. (a) x1 görüntüsü (b) x2 görüntüsü (c) x1’in histogram eşitleme yapılmış hali, (d) x2’nin histogram eşitleme yapılmış hali, (e) $AKSD^{-1}(f_{mu,mv}(k,l))$ grafiği, (f) $ASKD^{-1}(g_{mu,mv}(k,l))$	58
Şekil 4.12. (a) x1 görüntüsü (b) x2 görüntüsü, x’in ötelenmiş hali (c) Fourier faz hesaplama sonucu (d) korelasyon sonucu (e) sözde faz hesaplama $AKSD^{-1}$ sonucu (f) sözde faz hesaplama $ASKD^{-1}$ sonucu	60
Şekil 4.13. (a) x1 görüntüsü, arka plan 0-10 arası, nesne 20-30 arası rastgele değerlerden üretilmiştir (b) x2 görüntüsü, x1 görüntüsündeki nesnenin ötelenmiş hali (c) x1’in histogram eşitleme yapılmış hali (d) x2’nin histogram eşitleme yapılmış hali (e) Fourier faz hesaplama yöntemi sonucu (f) korelasyon sonucu (g) sözde faz hesaplama yöntemi $AKSD^{-1}$ sonucu (h) sözde faz hesaplama yöntemi $ASKD^{-1}$ sonucu	61

Şekil	Sayfa
Şekil 4.14. (a) x1 görüntü grafiği (b) x2 görüntü grafiği, x1'in üzerine gürültü eklenerek ötelenmiş hali (c) Fourier faz hesaplama yöntemi (d) korelasyon sonucu (e) sözde faz hesaplama AKSD ⁻¹ sonucu (f) sözde faz hesaplama sonucu ASKD ⁻¹ sonucu	63
Şekil 4.15. (a) x1 görüntü grafiği (b) x2 görüntü grafiği, x1'in üzerine gürültü eklenerek ötelenmiş hali (c) Fourier faz hesaplama yöntemi (d) korelasyon sonucu (e) sözde faz hesaplama AKSD ⁻¹ sonucu (f) sözde faz hesaplama sonucu ASKD ⁻¹ sonucu	64
Şekil 4.16. (a) x1 görüntüsü (b) x2 görüntüsü, x1'in +3,+5 ötelenmiş hali (c) x3 görüntüsü, x1'in +10,+8 ötelenmiş hali (d,e) x1 ve x2 üzerinde sözde faz hesaplama yöntemi AKSD ⁻¹ ve ASKD ⁻¹ sonuçları (f,g) x1 ve x3 üzerinde sözde faz hesaplama yöntemi AKSD ⁻¹ ve ASKD ⁻¹ sonuçları	65
Şekil 4.17. (a) x1 görüntüsünün eşiklenmiş hali (b) x2 görüntüsünün eşiklenmiş hali (c,d) a ve b görüntülerinin sözde faz hesaplaması sonucu ortaya çıkan AKSD ⁻¹ ve ASKD ⁻¹ grafikleri	67
Şekil 4.18. (a) x1 görüntüsü (b) x2 görüntüsü, x1'in ötelenmiş hali (c) x1 ve x2'nin gri seviyesi aralığı (d,e) x1 ve x2 üzerinde sözde faz hesaplaması AKSD ⁻¹ ve ASKD ⁻¹ sonuçları (f) y1, x1'in üssel fonksiyon ile gri seviyesi dönüşümü uygulanması sonucu (g) y2, x2'in üssel fonksiyon ile gri seviyesi dönüşümü uygulanması sonucu (h) y1 ve y2'nin gri seviyesi aralığı (i,i) y1 ve y2 üzerinde sözde faz hesaplaması AKSD ⁻¹ ve ASKD ⁻¹ sonuçları.....	68
Şekil 4.19. (a) x1 görüntüsü, (b) x2 görüntüsü, x1'in gürültü eklenmiş ve ötelenmiş hali (c), (d) x1 ve x2'ye sözde faz hesaplaması uygulamasının AKSD ⁻¹ ve ASKD ⁻¹ sonucu (e), (f) x1 ve x2'ye medyan filtre uygulandıktan sonra sözde faz hesaplamasının AKSD ⁻¹ ve ASKD ⁻¹ sonucu.....	70
Şekil 5.1. Çerçeve sayısı ile sıkıştırma oranı arasındaki ilişkiyi gösterir grafik	75
Şekil 5.2. Çerçeve sayısı ile MSE hata değeri arasındaki ilişkiyi gösterir grafik	76
Şekil 5.3. Budama sayısı ile sıkıştırma oranı arasındaki ilişkiyi gösterir grafik	77
Şekil 5.4. Budama sayısı ile MSE hata değeri arasındaki ilişkiyi gösterir grafik	78
Şekil 5.5. Sıkıştırma, küçük küpler üzerinde yapıldığında, sıkıştırma oranı ile budama sayısı arasındaki ilişkiyi gösterir grafik.....	79

Şekil	Sayfa
Şekil 5.6. Sıkıştırma, küçük küpler üzerinde yapıldığında, MSE hata değeri ile budama sayısı arasındaki ilişkiyi gösterir grafik.....	80
Şekil 5.7. Sıkıştırma, küçük küpler üzerinde yapıldığında, Sıkıştırma oranı ile blok kenar ölçütü arasındaki ilişkiyi gösterir grafik.....	81
Şekil 5.8. Sıkıştırma, küçük küpler üzerinde yapıldığında, MSE hata değeri ile blok kenar ölçütü arasındaki ilişkiyi gösterir grafik.....	82

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. (a) orijinal görüntü (b) düşük frekansların silinmesi sonucu ortaya çıkan görüntü (c) yüksek frekansların silinmesi sonucu ortaya çıkan görüntü	17
Resim 2.2. (a) Orijinal görüntü (b) AKD katsayıları 2 ile çarpılmış durum (c) AKD katsayıları 0,5 ile çarpılmış durum	18
Resim 2.3. (a) Orijinal görüntü (b) AKD katsayılarının satır ve sütunlarından ilk yarıları ters dönüşümde kullanılarak elde edilmiş görüntü (c) AKD katsayılarının satırların ilk yarısından örneklenerek ters dönüşümde kullanılarak elde edilmiş görüntü	19
Resim 2.4. AKD konvolüsyon ile görüntü karıştırma (a) logo görüntüsü (b) video görüntüsü (c) birleştirilmiş görüntü.....	21
Resim 2.5. (a) Satürn ve AKD'si (b) Çocuk ve AKD'si (c) Devre ve AKD'si (d) Ağaç ve AKD'si (e) Maymun ve AKD'si (f) Bir sinüs dalgası ve AKD'si	24
Resim 5.1. Uygulanan yöntemin hata analizi.....	83

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
2B	2 Boyutlu
3B	3 Boyutlu
AFD	Ayrık Fourier Dönüşümü
AKD	Ayrık Kosinüs Dönüşümü
AKSD	Ayrık Kosinüs Sinüs Dönüşümü
ASD	Ayrık Sinüs Dönüşümü
ASKD	Ayrık Sinüs Kosinüs Dönüşümü
BEA	Blok Eşleştirme Algoritması
DXT-ME	Discrete Cosine/Sine Based Motion Estimation
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio

1. GİRİŞ

Bilginin ifade edildiği düzlemde bir başka düzleme çevrilerek, bu düzlemde gösterilmesine dönüşüm denir. Daha çok sinyal işlemede kullanılan dönüşüm yöntemleri için bilgi, sürekli veya kesikli bir fonksiyonla ifade edilebilen sinyal anlamına gelmektedir. Bilginin ifade edilmesinde, zaman, genlik, frekans bilgileri kullanılmaktadır.

Sinyalin, frekans düzlemine dönüştürülmesinin temelinde, Joseph Fourier tarafından ortaya konulan teori bulunmaktadır. Bu teoriye göre, sürekli ve periyodik her fonksiyon, periyodik olan sinüs ve kosinüs fonksiyonlarından oluşmaktadır. Fourier dönüşümü, sinyal işlemede en çok kullanılan dönüşüm yöntemlerindedir. Sinyali, sinüsoidal alt fonksiyonlar (periyodik sinüs ve kosinüs fonksiyonları) ile ifade eder.

Teorinin esasında sürekli fonksiyonlar olsa da, kesikli fonksiyonların, sürekli fonksiyonların örneklenmiş hali olduğu düşünülerek, aynı dönüşüm yöntemlerinin kesitli fonksiyonlar üzerinde uygulanması sağlanmaktadır. Buna ise Ayrık Fourier Dönüşümü (AFD) denilmektedir. Ayrık sinyallerin alt kosinüs bileşenlerince gösterilmesi, Ayrık Kosinüs Dönüşümü (AKD) olarak adlandırılmaktadır.

Sinyalin zaman düzlemindeki gösterimi, sinyalin yapısına ve ifade ettiği bilgiye bağlı olarak incelenmesi ve işlenmesi için uygun olmayabilir. Bu yüzden frekans düzlemine dönüştürülerek, sinyal içerdiği değişimler olarak gösterilir. Özellikle görüntü ve video gibi, sinyalin zamansal ve uzaysal düzlemde gösterdiği değişimlerin anlam taşıdığı sinyaller için, frekans düzleminde incelenmesi ve işlenmesi daha kolay ve işlevsel olmaktadır. 3 boyutlu (3B) AKD sıkıştırma yönteminin temelinde, sinyalin frekans düzleminde incelenerek önemli anlam taşımayan değişimlerin yok edilmesi, böylece sinyali gösteren verinin azaltılması işlemi bulunmaktadır.

Görüntüyü ifade eden verinin daha az veriyle ifade edilebilmesi için görüntü içerisindeki artıklıklar işlenmektedir. Temel olarak artıklık oluşturan iki unsur

bulunmaktadır. Birincisi; görüntüyü oluşturan pikseller arasındaki gri seviyesi benzerliğidir. Uzaysal düzlemde birbirlerine yakın bölgeler yakın renklere sahiptirler. Bu yakın gri seviyeleri, görüntünün frekans düzleminde işlenerek verinin sıkıştırılmasını sağlamaktadır. İkinci unsur ise zaman düzlemindeki hareket unsurudur. Artarda gelen zamanlara ait görüntülerde, hareket bulunmayan alanlardaki gri seviyeleri birbirlerine yakın değerler taşımaktadırlar. Hareket olan alanlarda ise, hareket eden cismin yer değiştirmesinden dolayı gri seviyelerinin oluşturduğu bölgeler yer değiştirmektedir. Bir görüntü dizisi içerisindeki artarda gelen görüntülerdeki benzerlik frekans düzleminde işlenerek veri sıkıştırılmaktadır. Bunun yanı sıra artarda gelen görüntüler tekrar ifade edilmek yerine, hareket belirlenerek iki görüntü arasındaki fark hesaplanmakta ve sadece referans görüntü ve üzerindeki fark işlenerek, ardışık görüntüler sıkıştırılabilmektedir.

Özet olarak literatür taramasında, 3B AKD sıkıştırma uygulamalarına dair aşağıdaki konuların incelendiği ve amaçlandığı görülmektedir:

İşlem zamanının azaltılabilmesi amacıyla; (a) 3B AKD hesaplamasında yeni yöntemler geliştirilmektedir. Bu yöntemlerde, hesaplama içerisindeki toplama, çarpma işlem sayıları azaltılmaya çalışılmaktadır. (b) Sıkıştırma yöntemlerinin işlem miktarını azaltmak amacıyla optimize edilmeye çalışılmaktadır.

Hafızada daha az yer kaplayan ve aktarım hızı düşük ortamlarda daha yüksek kaliteli görüntüler elde etmek için, yüksek oranda sıkıştırma yapılabilmesi için yeni yöntemler geliştirilmesi amaçlanmıştır. Bu, üç farklı bakışla gerçekleşmektedir; (a) doğrudan sıkıştırma yöntemlerinin geliştirilmesi. Run-length, aritmetik ve Huffman yöntemlerinin farklı kullanım şekilleriyle optimize edilmesi veya AKD katsayılarının sıkıştırılmasına uygun yeni yöntemler geliştirilmesi. (b) Sıkıştırma işlemi için, verinin uygun bir hale getirilmesi. Farklı sıralama yöntemleri ve nicemleme değerlerinin seçiminde yeni yöntemler ile bu işlem gerçekleştirilmektedir. (c) Nicemleme yerine budama işleminin kullanılması.

Literatürdeki çalışmalarda görüntü kalitesinin yükseltilmesi amaçlanmıştır ve yöntem

olarak budama veya nicemleme yöntemlerinin uygulanması sırasında, kalite için önemli olan verilerin korunabilmesi uygulanmaktadır.

Literatürde, sıkıştırmanın etkisini artırabilmek veya zaman ve işlemden kazanç sağlamak amacıyla yapılmış araştırmalar bulunmaktadır.

Bir araştırmada, 3B AKD katsayılarının nicemlenmesi için, nicemleme değerlerinin üretilmesinde yeni bir yöntem geliştirilmiştir. Katsayıları belirten küp matris, iki bölgeye bölünmektedir. Anlamlı katsayıların yer aldığı bölge, yüksek frekansların yer aldığı bölgedir. Nicemleme değerleri, üssel fonksiyon ile hesaplanmaktadır. Bu fonksiyon aynı zamanda tarama sırasında kullanılacak sıralamanın belirlenmesinde kullanılmaktadır ve bu sıralama, klasik sıralamaya göre daha iyidir. Böylelikle daha fazla sıkıştırma için uygun bir veri deseni ortaya çıkarılması amaçlanmıştır [1].

Bir başka makalede ise nicemleme yerine budama işleminin, 2 boyutlu görüntü sıkıştırma uygulamalarında, sıkıştırma oranına ve sıkıştırma performansına etkisi incelenmiştir. Araştırmada paralel işleme uygulanmış ve 8x8'lik matrisler paralel olarak hesaplanmıştır [2].

Samsung Elektronik'ten Kim ve Shin'in yayınladıkları bir makalede, 2B AKD üzerinde 8x8'lik blok matrisler üzerinde budama uygulanırken, budama genişliği adaptif olarak, 8x8'lik bloğun uzaysal düzlemdeki sinyalinden enerji hesaplaması yapılmakta ve hesaplanan enerji değerine göre, sol üst köşeden 1x1, 3x3 veya 4x4'lük AKD katsayıları olarak kabul edilmekte, diğerleri budanmaktadır. Budama işlemi yapılarak sıkıştırma ve geri açma sırasındaki toplama ve çıkarma işlemleri azaltılması sağlanırken, adaptif budama genişliği sayesinde görüntü kalitesinin bozulmaması amaçlanmaktadır [3].

1991 senesinde yapılan bir araştırmada, nicemleme ve sıralama sonrasında ortaya çıkan veri yapısına ve de 0 değerlerinin sayısına göre, Run-length veya Huffman sıkıştırma yöntemlerinden biri uygulanarak daha yüksek oranda bir sıkıştırma gerçekleştirmek amaçlanmıştır [4].

Boussakta ve Alshibami, 3B AKD hesaplamasının daha hızlı yapılabilmesi için vektör tabanlı bir hesaplama yöntemi geliştirmiştir. Bu yöntem ile 3B AKD hesaplaması sırasındaki çarpma işlemi sayısı azaltılarak, işlem zamanının azaltılması sağlanmıştır [5].

Bu çalışmada, Matlab ile 3B AKD sıkıştırma yöntemini kullanan bir uygulama hazırlanmış ve budama katsayısının, çerçeve sayısının ve görüntüdeki değişimin sıkıştırmaya etkisi incelenmiştir.

Ayrıca, harekete dayalı sıkıştırma algoritmalarında kullanılabilmesi amacıyla hareket tanıma yöntemleri incelenmiştir. Özellikle sıkıştırılmış görüntü dizilerinin üzerinde işlem yapabilmek için, frekans düzleminde hareket tanıma konusu incelenmiştir. Bu sayede görüntü dizilerinin geri açılmasına gerek kalmadan, yüksek hızda daha az işlem ile hareket tanıma yapılabilir. Sözde faz hesaplanarak hareket tanıma yönteminin, üzerinde yapılan iyileştirmelerle gürültülü ve nesnenin ayırt edilebilirliğinin düşük olduğu ortamlarda başarılı bir şekilde uygulanabilmesi sağlanmıştır.

Çalışmanın 2. bölümünde AKD açıklanmakta ve matematiksel tanımları yapılmaktadır. Yine bu bölümde AKD'nin özellikleri anlatılmakta ve bu özelliklerin görüntü sıkıştırma uygulamalarındaki etkisi açıklanmaktadır. 3. bölümde 3B AKD ile sıkıştırma yöntemi alt işlemleriyle beraber anlatılmaktadır. 4. bölümde hareket tanıma anlatılmaktadır. Literatürdeki çalışmalar ve video görüntüleri üzerindeki uygulamalar anlatılmıştır. 5. bölümde uygulanan 3B AKD ile sıkıştırma yöntemi sonuçlarıyla beraber açıklanmıştır. Sonuç olarak 6. bölümde hazırlanan bu çalışmanın ve sonuçları öneriler sunulmaktadır.

2. AYRIK KOSİNÜS DÖNÜŞÜMÜ (AKD)

AKD, sinyalin kendisini oluşturan temel kosinüs fonksiyonları şeklinde gösterilerek, frekans düzlemine çevrilmesidir. AFD, karmaşık sayı düzleminde tanımlı bir sinyali, karmaşık spektrumuna çevirir. Gerçek sayı düzleminde tanımlı bir sinyalin, karmaşık kısmı sıfırdır ve AFD alanında bu sinyalin gerçek kısmı simetriktir. Bu da frekans düzleminde veri tekrarı demektir. AKD’de ise sadece gerçek katsayılar spektruma aktarılmaktadır. Böylece AFD’ye göre daha az bellek kullanılarak aynı verinin tanımlanması sağlanır [6].

Bir sinyalin frekans düzlemindeki gösterimi, sinyalin, içerdiği değişimler olarak ifade edilmesidir. Sayısal görüntü bilgisinde sadece gerçek sayı düzleminde veriler olduğu için, ve de görüntünün ayırt edilebilirliği, içeriğindeki değişimlere bağlı olmasından dolayı görüntü işlemede çoğunlukla AKD kullanılmaktadır. Bununla birlikte AKD, sinyalin enerjisini diğer dönüşüm yöntemlerine göre daha küçük bir alana sıkıştırarak, sinyalin daha az sayıda veriyle ifade edilebilmesine olanak sağlar. Bu yüzden sıkıştırma uygulamaları için elverişli bir veri ortaya çıkarır.

2.1. AKD Hesaplama Yöntemleri

AKD, hesaplama yöntemleri birbirlerinden biraz farklı olmak üzere farklı tiplere sahiptir.

1. tip AKD matematiksel olarak Eş.2.1’deki gibi hesaplanmaktadır. 1. Tip AKD’nin ters dönüşümü yine kendisiyle hesaplanmaktadır.

$$X(k) = \frac{1}{2} \left(x(0) + (-1)^k x(N-1) + \sum_{n=1}^{N-2} x(n) \cos \left[\frac{\pi nk}{N-1} \right] \right) \quad (2.1)$$

$$k = 0, 1, 2, \dots, N-1$$

x_0 ve x_{N-1} terimleri $\sqrt{2}$ ile çarpılarak, AKD matrisi ortogonal yapılabilmektedir.

2. tip AKD matematiksel olarak Eş. 2.2'deki gibi hesaplanmaktadır. Uygulamalarda en çok kullanılan yöntem, 2. tip AKD'dir. Bu yüzden genel olarak AKD söylemi bu tipe işaret etmektedir.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)\pi k}{2N} \right] \quad (2.2)$$

$(k = 0, 1, 2, \dots, N-1)$

3. Tip AKD matematiksel olarak Eş. 2.3'teki gibi hesaplanmaktadır. 3. Tip AKD aynı zamanda 2. Tip AKD için ters dönüşüm fonksiyonudur.

$$X(k) = \frac{1}{2} x(0) + \sum_{n=1}^{N-1} x(n) \cos \left[\frac{(2k+1)\pi n}{2N} \right] \quad (2.3)$$

$(k = 0, 1, 2, \dots, N-1)$

x_0 terimi 2. tip AKD hesaplamasında $1/\sqrt{2}$ ile, 3. tip AKD hesaplamasında $\sqrt{2}$ ile çarpılarak AKD matrisleri ortogonal yapılabilir.

Literatürde en çok kullanılan bu 3 tip hesaplamının yanı sıra 4., 5., 6., 7. ve 8. tip hesaplama yöntemleri de geliştirilmiştir.

AKD çok boyutlu matrisler üzerinde de uygulanabilir. En çok bilinen yöntem ile çok boyutlu matrislerde önce 1. boyutta satır satır AKD katsayıları hesaplanır. Daha sonra bu satırların oluşturduğu çok boyutlu matriste, 2. boyut üzerinde sütun sütun hesaplanır ve her bir boyut için bu işlem gerçekleştirilir. Örnek olarak 3B AKD hesaplaması Eş. 2.4'te görüldüğü gibidir.

$$S(u, v, w) = \sqrt{\frac{8}{N_F N_R N_C}} k(u) k(v) k(w) \sum_{x=0}^{N_F-1} \sum_{y=0}^{N_R-1} \sum_{z=0}^{N_C-1} s(x, y, z) \cos \left[\frac{(2u+1)\pi x}{2N_C} \right] \cos \left[\frac{(2v+1)\pi y}{2N_R} \right] \cos \left[\frac{(2w+1)\pi z}{2N_F} \right] \quad (2.4)$$

$$k(n) = \frac{1}{\sqrt{2}}, n = 0$$

$$k(n) = 1, n \neq 0$$

Ters 3B AKD hesaplaması Eş. 2.5'te gösterilmektedir.

$$s(x, y, z) = \sum_{u=0}^{N_F-1} \sum_{v=0}^{N_R-1} \sum_{w=0}^{N_C-1} \sqrt{\frac{8}{N_F N_R N_C}} k(u)k(v)k(w)S(u, v, w) \cos\left[\frac{(2x+1)\pi u}{2N_C}\right] \cos\left[\frac{(2y+1)\pi v}{2N_R}\right] \cos\left[\frac{(2z+1)\pi w}{2N_F}\right] \quad (2.5)$$

$$k(n) = \sqrt{2}, n = 0$$

$$k(n) = 1, n \neq 0$$

2.2 AKD Matrisi

Ayrık sinyallere kosinüs dönüşümü uygulamasının formüller hesaplamaları Bölüm 2.1'de gösterildiği şekilde yapılabilmektedir. Bunun yanısıra sinyaller, tanımlanmış AKD matrisleri ile çarpılarak AKD katsayıları hesaplanabilmektedir.

B; n x n 'lik ayrık sinyal, A; n x n lik AKD matrisidir. AB matris çarpımı, B'nin kolonlarının AKD katsayılarını hesaplamaktadır. 2 boyutlu (2B) AKD hesaplaması ise ABA^{-1} matris çarpımıyla yapılabilmektedir. $A^{-1}B$ matris çarpımı ile B'nin kolonlarının AKD ters dönüşümü katsayıları hesaplanabilmektedir.

$$\begin{bmatrix} 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 \\ 0,4904 & 0,4157 & 0,2778 & 0,0975 & -0,0975 & -0,2778 & -0,4157 & -0,4904 \\ 0,4619 & 0,1913 & -0,1913 & -0,4619 & -0,4619 & -0,1913 & 0,1913 & 0,4619 \\ 0,4157 & -0,0975 & -0,4904 & -0,2778 & 0,2778 & 0,4904 & 0,0975 & -0,4157 \\ 0,3536 & -0,3536 & -0,3536 & 0,3536 & 0,3536 & -0,3536 & -0,3536 & 0,3536 \\ 0,2778 & -0,4904 & 0,0975 & 0,4157 & -0,4157 & -0,0975 & 0,4904 & -0,2778 \\ 0,1913 & -0,4619 & 0,4619 & -0,1913 & -0,1913 & 0,4619 & -0,4619 & 0,1913 \\ 0,0975 & -0,2778 & 0,4157 & -0,4904 & 0,4904 & -0,4157 & 0,2778 & -0,0975 \end{bmatrix}$$

Şekil 2.1. 8x8'lik AKD matrisi

Şekil 2.1'de 8x8'lik AKD matrisi gösterilmiştir. Formüller hesaplamalardaki, sinyalin indis numaralarına bağlı olan kosinüs hesaplamaları yerine AKD matrisi içerisinde önceden tanımlanmış değerler verilmektedir. Sinyalin her bir satırındaki değerlerin, kosinüs katsayılarıyla çarpılarak seri halinde toplanması işlemi ise matris çarpımıyla

uygulanmaktadır.

AKD matrisi, görüntü işleme uygulamalarında, sıkıştırılmış görüntülerin uzaysal düzleme çevrilmesi veya uzaysal düzlem verilerinin AKD katsayılarının hesaplamalarında kolaylık oluşturmaktadır. Donanımsal görüntü işleme uygulamalarında, entegre devrelerle AKD hesaplamasında yaygın olarak kullanılmaktadır. Aynı zamanda, en yaygın olarak kullanılan Joint Photographic Experts Group (JPEG) sıkıştırma yöntemi içerisinde görüntü 8x8'lik bloklara ayrıştırılarak işlem yapıldığından, JPEG uygulamalarında hesaplama kolaylığı gerektiren noktalarda AKD matrisleri kullanılmaktadır.

Bir çok araştırmacı tarafından AKD katsayılarının hesaplanmasında daha az işlem gerçekleştirilmesi ve daha hızlı hesaplanabilmesi amacıyla yeni yöntemler geliştirilmiştir. Bu yöntemler ağırlıklı olarak, geleneksel hesaplama yöntemlerinin matematiksel olarak farklı formülasyonlarla ifade edilmesine dayanmaktadır. Bu sayede görüntü işleme uygulamalarında daha hızlı işlem gerçekleştirilmesi amaçlanmaktadır.

Kou ve Fjällbrant (1991), artarda gelen blokların AKD katsayılarını beraber hesaplayarak, çarpma ve toplama işlemlerinin sayılarını düşürmüşlerdir. Bu yönteme "Doğrudan Algoritma" ismini vermişlerdir. Yöntemin temelinde AKD matrisi kullanılmaktadır. Ardarda gelen iki blok için ayrı ayrı AKD katsayıları hesaplandıktan sonra, tek bir blok gibi düşünülerek, diğer bloğun katsayılarının yeniden hesaplanmasında ikinci bloğun hesaplanmış AKD katsayıları kullanılmaktadır. Bu yüzden, artarda blokların açılması gerektiren görüntü işleme ve ses işleme uygulamaları için uygundur. Bu yönteme göre, ters dönüşüm hesaplanmasında da, artarda gelen blokların sırasının bozulmaması gerekmektedir. Sonuç olarak Çizelge 2.1'de toplama ve çarpma işlemi sayılarının azaldığı görülmektedir [7].

Çizelge 2.1. Doğrudan Algoritma'nın sonuçlarının Geleneksel Algoritma'yla karşılaştırılması [7]

N	Toplama Sayısı		Çarpma Sayısı	
	GA	DA	GA	DA
4	24	24	28	26
8	78	68	68	60
16	228	204	172	160
32	576	532	404	376
64	1 380	1 292	892	824
128	3 192	3 012	1 892	1 736

Pao ve Sun (1998), video kodlayıcıların daha verimli çalışabilmesi için, AKD katsayılarının yaklaşık değerlerinin hesaplanması yöntemini önermişlerdir. Hesaplanan AKD katsayılarının Peak Signal to Noise Ratio (PSNR) değeri, kabul edilebilir bir değer olmaktadır. Bu çalışmada Pao ve Sun, nicemleme sonrasında ortaya çıkan en son sıfır olmayan AKD katsayısı ile, nicemleme parametreleri arasındaki ilişkiyi inceleyerek, bu veriyi adaptif olarak 8 x 8'lik bloğun tamamındaki AKD katsayılarını veya bir kısmını hesaplamak için kullanılmaktalar. İkinci olarak, işlem sayısını düşürebilmek amacıyla, AKD katsayılarını klasik yöntemlerle hesaplamak yerine, yaklaşık değerlerinin bulunmasını önermektedirler. Sonuç olarak Çizelge 2.2'deki tabloda görüleceği üzere, video görüntüsünün PSNR değeri çok fazla değişmezken, işlem sayılarından önemli kazanç sağlanmıştır [8].

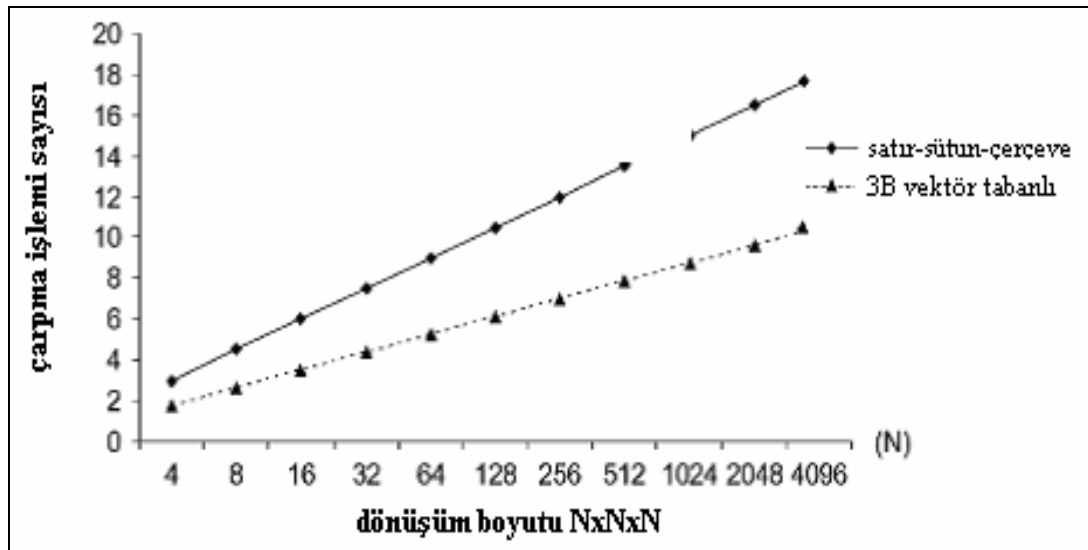
Çizelge 2.2. Pao ve Sun'un çalışmalarının sonuç tablosu [8]

	Video Gör. 1 (20 kb/s) Orijinal	Video Gör. 1 (20 kb/s) önerilen yöntem	Video Gör. 2 (20 kb/s) orijinal	Video Gör. 2 (20 kb/s) önerilen yöntem
8x8 AKD sayısı	570 834	429 000	89 100	39 798
4x4 AKD sayısı	0	141834	0	49 302
Toplamdaki kazanç	-	%18,85	-	%41,98
Çarpmadaki kazanç	-	%16,56	-	%236,89
PSNR (dB)	31,0757	31,0747	36,1095	36,0892
Bozulma PSNR (dB)	-	0,0010	-	0,0203

Çizelge 2.2. Devam Pao ve Sun'un çalışmalarının sonuç tablosu [8]

	Video Gör. 1 (40 kb/s) Orijinal	Video Gör. 1 (40 kb/s) önerilen yöntem	Video Gör. 2 (40 kb/s) orijinal	Video Gör. 2 (40 kb/s) önerilen yöntem
8x8 AKD sayısı	570 834	493 080	89 100	81 972
4x4 AKD sayısı	0	77 754	0	7 128
toplamdaki kazanç	-	%10,33	-	%6,07
çarpmadaki kazanç	-	%9,08	-	%5,33
PSNR (dB)	33,1631	33,1589	38,4874	38,5069
Bozulma PSNR (dB)	-	0,0042	-	0,0195

Boussakta ve Alshibami (2004) ise, 3 boyutlu AKD hesaplamalarında klasik satır sütun-çerçeve yöntemiyle AKD hesaplamalarının geliştirilerek daha hızlı yöntemleri önermek yerine, vektör tabanlı hesaplamayı önermektedirler [5]. Bu yönteme göre, AKD prosedürü içerisinde hesaplanan bazı değerler, sonraki katsayıların hesaplanmasında da kullanılmaktadır. Böylelikle toplama ve çarpma işlem sayıları azalmaktadır. Şekil 2.2'de bu yöntemin, satır-sütun-çerçeve tabanlı hesaplama yöntemiyle karşılaştırması görülmektedir.

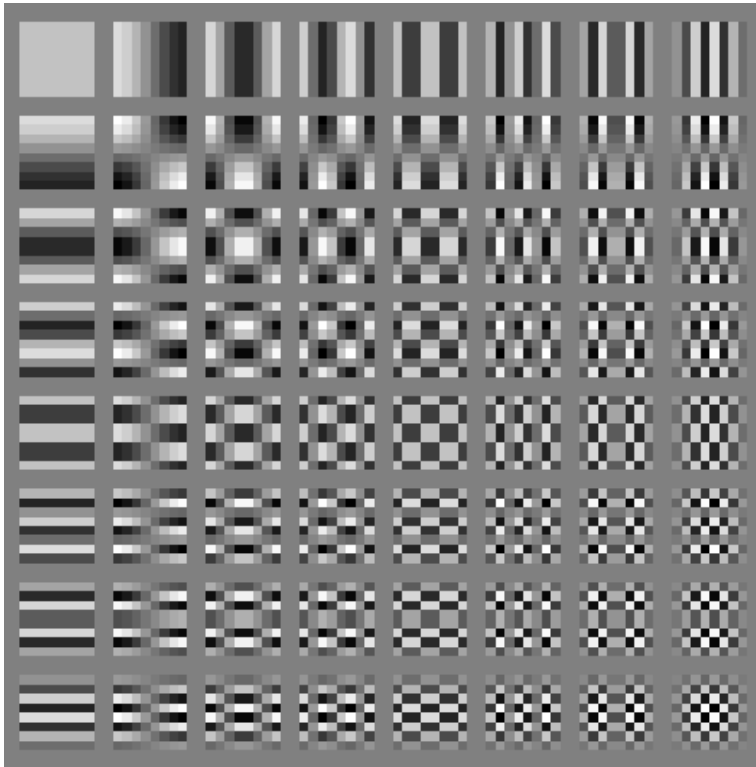


Şekil 2.2. Vektör tabanlı 3B AKD hesaplama sonuçları

2.3 Taban Görüntüler

Bir sinyalin AKD uzayına dönüşümü, o sinyali oluşturan kosinüs fonksiyonlarının toplamı şeklinde ifade edilmesidir. Farklı frekanslardaki kosinüs fonksiyonları toplanarak, diğer sinyaller oluşturulabilir. Sinyali oluşturan farklı frekanslardaki sinüzoidal fonksiyonlara temel fonksiyonlar denilmektedir.

Görüntü iki boyutlu sinyal olarak ele alınmaktadır. Bu durumda bir görüntü için, AKD dönüşümünü oluşturan temel sinüzoidal fonksiyonlardan oluşan görüntülere 'Taban Görüntüler' denilmektedir.



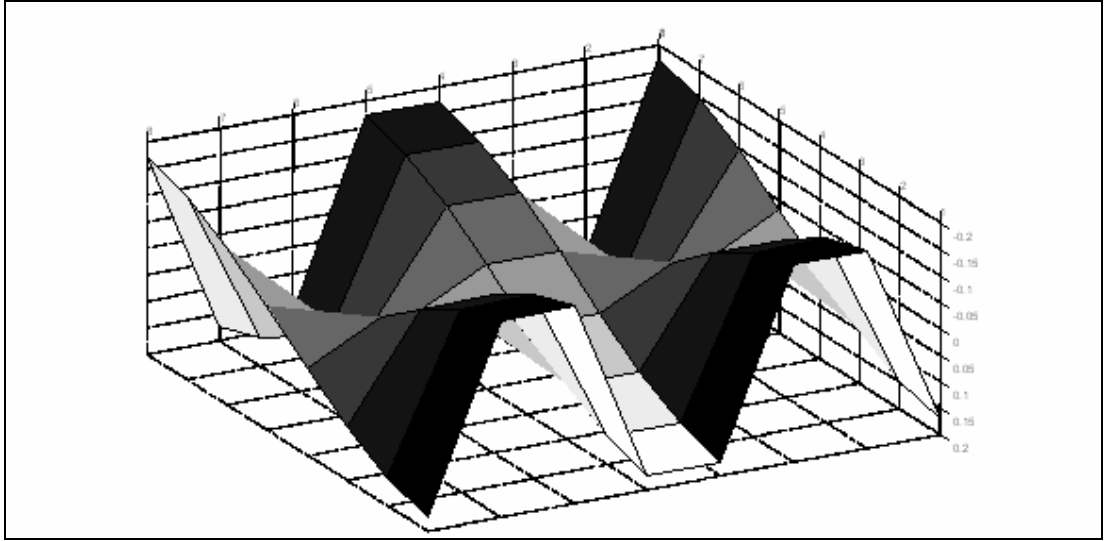
Şekil 2.3. Taban görüntüler

$N \times N$ 'lik bir görüntü için, $N \times N$ kadar taban görüntü vardır. Taban görüntüler, tüm elemanları 0 olan, yalnız ifade ettiği eleman 1 olan matrislerin ters dönüşümü ile elde edilir. Örnek olarak 8×8 'lik bir görüntüde (5,2) bileşeni için taban görüntü Şekil 2.4'te olduğu gibidir.

$$AKD^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

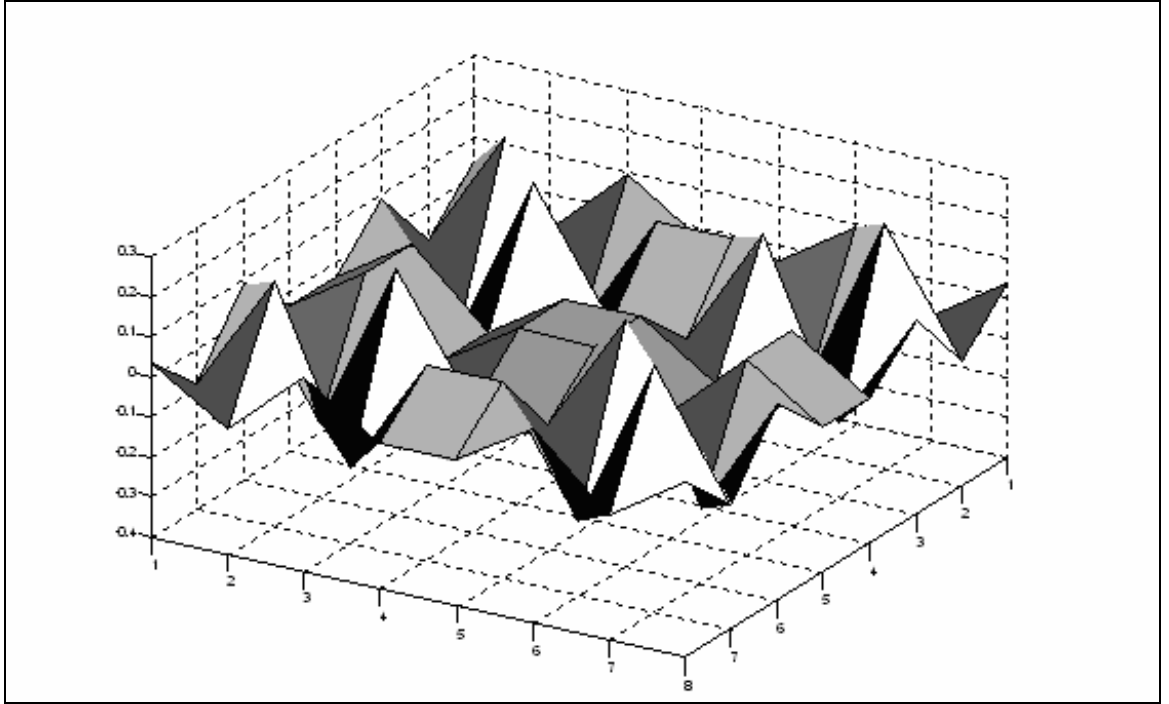
$$= \begin{bmatrix} 0,1734 & -0,1734 & -0,1734 & 0,1734 & 0,1734 & -0,1734 & -0,1734 & 0,1734 \\ 0,1470 & -0,1470 & -0,1470 & 0,1470 & 0,1470 & -0,1470 & -0,1470 & 0,1470 \\ 0,0982 & -0,0982 & -0,0982 & 0,0982 & 0,0982 & -0,0982 & -0,0982 & 0,0982 \\ 0,0345 & -0,0345 & -0,0345 & 0,0345 & 0,0345 & -0,0345 & -0,0345 & 0,0345 \\ -0,0345 & 0,0345 & 0,0345 & -0,0345 & -0,0345 & 0,0345 & 0,0345 & -0,0345 \\ -0,0982 & 0,0982 & 0,0982 & -0,0982 & -0,0982 & 0,0982 & 0,0982 & -0,0982 \\ -0,1470 & 0,1470 & 0,1470 & -0,1470 & -0,1470 & 0,1470 & 0,1470 & -0,1470 \\ -0,1734 & 0,1734 & 0,1734 & -0,1734 & -0,1734 & 0,1734 & 0,1734 & -0,1734 \end{bmatrix}$$

Şekil 2.4. 8x8'lik görüntüde taban görüntü matrisi



Şekil 2.5. 8x8'lik görüntüde (5,2) elemanının taban görüntü grafiği

Şekil 2.5'te 8x8'lik bir görüntüde (5,2) indisindeki taban görüntünün grafiği görülmektedir.



Şekil 2.6. 8x8'lik görüntüde (7,7) elemanının taban görüntü grafiği

Şekil 2.6'da ise 8x8'lik bir görüntüde (7,7) indisindeki taban görüntünün grafiği görülmektedir. Şekil 2.6 ve Şekil 2.5'te grafikleri gösterilen taban görüntüler arasındaki fark frekans farkıdır, her ikisi de görüntüyü oluşturan iki farklı frekans bileşenini taşımaktadır.

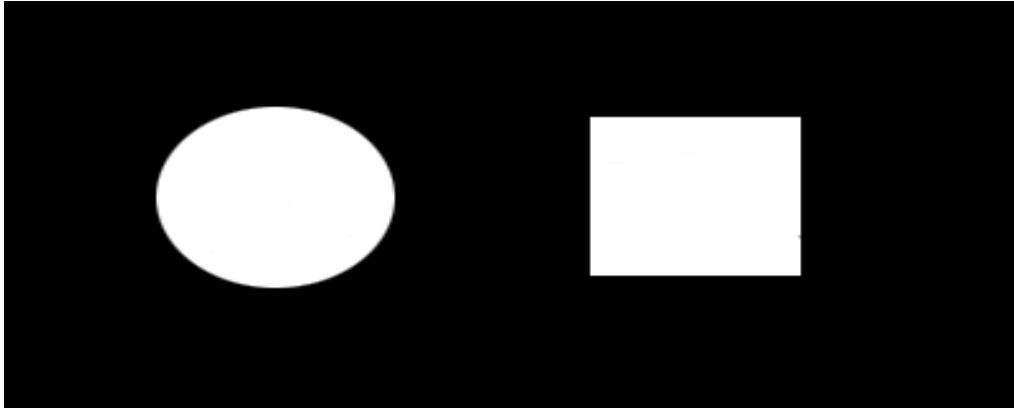
Taban görüntüler kullanılarak AKD katsayıları hesaplanabilmektedir. Örnek olarak $N \times N$ 'lik bir görüntünün (u,v) indisindeki AKD katsayısını hesaplamak için Eş.2.6'daki formül kullanılmaktadır.

$$X(u, v) = \sum_{m=1}^N \sum_{n=1}^N x(m, n) \text{temel_görüntü}_{uv}(m, n) \quad (2.6)$$

Bu sayede AKD matrisi kullanımına benzer şekilde, bir görüntünün AKD hesaplaması taban görüntü matrisleri kullanılarak kolaylıkla yapılabilmektedir.

2.4 Görüntü İşleme Uygulamalarında AKD

Görüntü iki boyutlu bir sinyal olarak ele alınmaktadır. Video görüntüleri ise, üçüncü boyutu zaman olan görüntü sinyalleri olarak ifade edilebilir. Görüntü sinyalinin en önemli özelliği, insanlar tarafından anlaşılabilirliğinin, sinyalin çözünürlüğüne bağlı olmasıdır. Uzaysal ve gri seviyesi düzlemindeki çözünürlüklerin yüksek olması, görüntünün insanlar tarafından ayırt edilebilirliğini artırır. Görüntüyü oluşturan şey, görüntüdeki en küçük nokta olarak tanımlanabilecek olan piksellerin çevresindeki diğer piksellerle ilişkisidir.



Şekil 2.7. Farklı iki görüntü

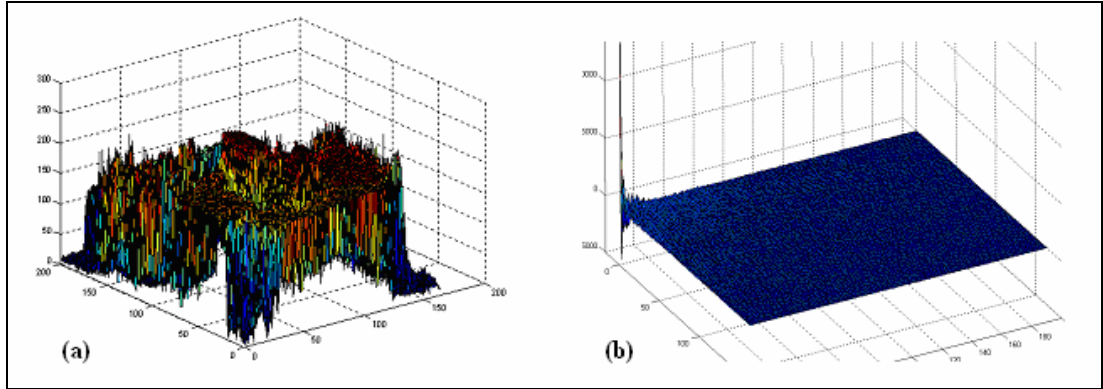
Şekil 2.7’de birbirinden farklı iki adet görüntü bulunmaktadır. Öncelikle bu şekil içerisinde görüntü olduğu insan beyni tarafından, çevresindeki renk ile görüntünün renginin farklı oluşundan algılanabilmektedir. Bu iki görüntünün birbirinden farklı görüntüler olduğu ise, piksellerin birbirleriyle olan yer ilişkisinden dolayı algılanabilmektedir. Dolayısıyla görüntü, uzay ve renk düzlemindeki farklılıklardan oluşmaktadır.

Görüntünün kendisi bir farklılık algılaması olduğundan, sayısal olarak görüntüyü ifade edebilmek için de bu farklılıklar kullanılmaktadır. Bir görüntü sinyali frekans düzlemine dönüştürülerek, içerdiği değişimlerle ifade edildiğinde, matematiksel olarak anlamaya ve işlem yapmaya uygun bir şekilde ifade edilmiş olur.

Görüntü sinyali içerisinde gerçek sayı düzleminden sayılar içermektedir. Bu yüzden görüntü işleme uygulamaları içerisinde en yaygın olarak AKD kullanılmaktadır. Çünkü AKD, sinyali içerdiği kosinüs dalgaları şeklinde ifade etmektedir. Kosinüs katsayıları sinyalin gerçek kısmına aittir. Fourier dönüşümü ve diğer bazı dönüşüm yöntemleri sinyali gerçek ve sanal kısımlarıyla ifade ettiklerinden, görüntü için uygulandığında simetrik olacak ve fazla bilgi olarak tanımlanacaktır. Bu yüzden AKD kullanımı, sinyalin ifade edilmesinde gerekli bellek miktarını azaltmaktadır.

En yaygın kullanılan görüntü sıkıştırma yöntemi olan JPEG ve en yaygın video sıkıştırma yöntemi Moving Pictures Expert Group (MPEG), görüntüyü AKD katsayılarıyla tanımlayarak işlem yapmaktadır. Günlük hayattaki ve doğal görüntülerde değişim düşüktür. Hızlı geçişler yoktur. Bu yüzden Bölüm 3.2’de anlatılan AKD özelliklerinden “Enerji Sıkıştırma” özelliği, görüntü sinyalinin sahip olduğu enerjiyi bir bölgeye sıkıştırabilmektedir. Bu da görüntü işleme uygulamalarının üzerinde çalıştığı görüntülerin birçoğu için daha az veri ile işlem yapabilmesini sağlamaktadır.

Dönüşüm kodlama, çağdaş görüntü ve video işleme tekniklerinin bütünleştirici bileşenini oluşturmaktadır. Dönüşüm kodlama, görüntü içerisindeki bir pikselin, komşu pikseller ile ilişki içerisinde olduğu önermesine dayanmaktadır. Benzer şekilde, video aktarımında, bir piksel, kendisinden sonra gelen kareler içerisinde aynı koordinatlardaki pikseller ile yakın ilişki içerisinde. Sonuç olarak, pikseller arasındaki bu ilişkiler kullanılarak bir pikselin değeri, ilişkili olduğu diğer piksellerden yola çıkılarak tahmin edilebilir. Bu halde, dönüşüm, uzaysal düzlemdeki bu ilişkili verilerin, ilişkisiz katsayılara çevrilmesidir. Dönüştürülen bu ilişkisiz katsayılardan, uzaysal düzlemdeki değişim ortaya çıkartılır, bu sayede bir pikselin değeri, değişim katsayılarına göre tahmin edilebilir [7].



Şekil 2.8. (a) Görüntünün sinyal gösterimi (b) a görüntüsünün AKD katsayılarının dağılımı

Şekil 2.8’de (a) görüntüsü sinyal olarak ele alınıp grafik düzlemine aktarıldığında, her bir satır ve sütunlar grafiğin x ve y eksenlerini, gri seviyesi değerleri ise z eksenini oluşturmaktadır (b)’de ise (a)’daki görüntünün AKD katsayıları görülmektedir. Görüntü uzay düzleminde renk değerleriyle ifade edilirken sinyalin enerjisi tüm uzaysal düzleme yayılmışken, aynı görüntüyü frekans düzlemine çevirerek AKD katsayıları hesaplandığında, AKD katsayılarının oluşturduğu sinyalin (0,0) en düşük frekans noktasına yakın bölgede yoğunlaştığı görülmektedir.

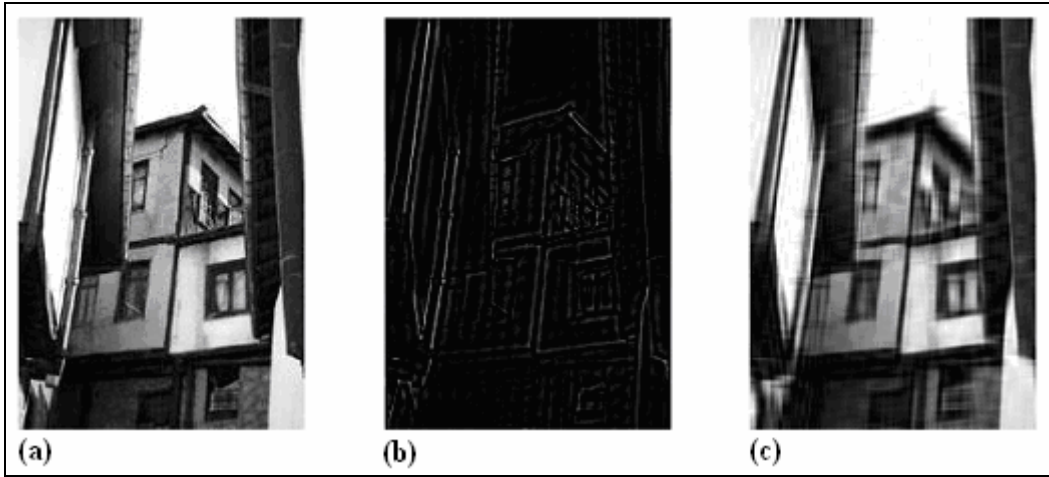
2.5 AKD Düzleminde Görüntü İşleme

Bir sinyalin AKD’si, sinyalin sahip olduğu kadar örnekleme sayısına sahiptir. 2 ve daha fazla boyutlu sinyaller içinde aynısı geçerlidir. $N \times N$ ’lik bir görüntüye AKD uygulanması sonucu yine $N \times N$ ’lik bir sinyal ortaya çıkar. N boyutlu bir sinyalin AKD’si üzerinde 0. indis sinyalin içerisindeki en düşük frekansın sahip olduğu yoğunluğu, N . indis ise en yüksek frekansın yoğunluğunu göstermektedir. N değeri ne olursa olsun ilk ve son frekansların değerleri aynı olduğundan, N (örnekleme sayısı) uzaysal düzlemde çözünürlüğü belirttiği gibi, frekans düzleminde de, sinyali ifade etmek için kullanılacak frekansların çözünürlüğünü belirlemektedir.

Bir görüntünün AKD sinyali üzerinde yapılacak değişiklik ve işlemler, görüntünün üzerinde de değişikliğe sebep olur. Görüntünün renksel ve uzaysal verileri,

görüntüyle ilgili bazı bilgilerin okunması için elverişli değildir. Ancak frekans düzlemindeki veriler, görüntüyü değişimleriyle tanımladığından dolayı bazı bilgiler bu düzlemde daha kolay okunabilir. Bu yüzden AKD üzerinde görüntü işleme yöntemleri geliştirilmekte ve uygulanmaktadır.

AKD katsayılarından bazılarının sıfır yapılması, o katsayıların ifade ettiği frekansların görüntü üzerinde yok edilmesi anlamına gelmektedir. Görüntü üzerindeki şekillerin kenarları, bir gri seviyesinden diğerine hızlı bir geçiş sağladığı için yüksek frekans anlamına gelmektedir. Bir şeklin iç dolgusundaki yakın gri seviyesine sahip renkler ise düşük frekansları oluşturmaktadır. AKD matrisinde satır ve sütunların en son kısımları yüksek frekansları ifade etmektedir. Bu bölgedeki katsayılar 0 yapıldığında, görüntü içindeki kenarlar yumuşar ve bozular. (0,0) indisine yakın bölgedeki katsayılar 0 yapıldığında, görüntü içerisindeki kenarlar belirginleşir, düz kısımlar silinir. Bu durum Resim 2.1’de görülmektedir.

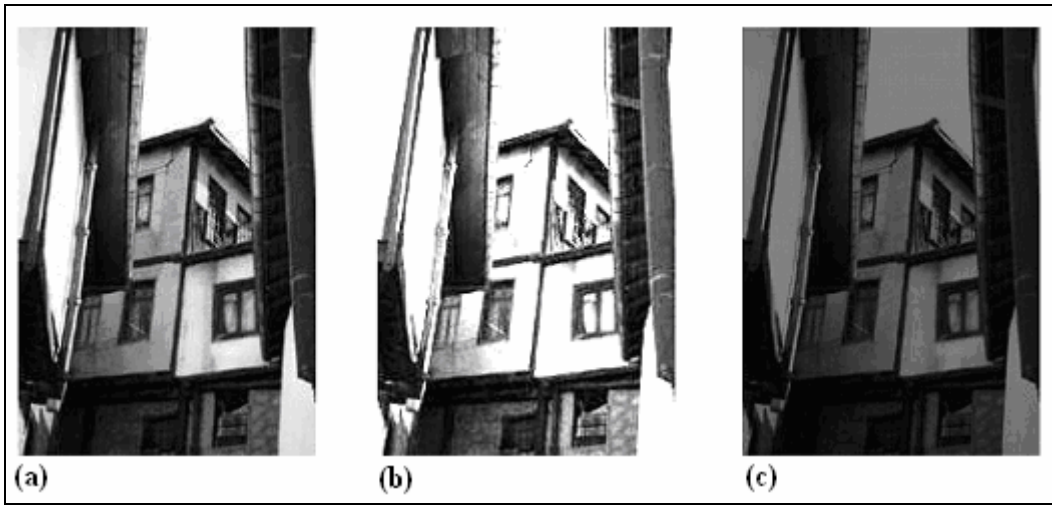


Resim 2.1. (a) orijinal görüntü (b) düşük frekansların silinmesi sonucu ortaya çıkan görüntü (c) yüksek frekansların silinmesi sonucu ortaya çıkan görüntü

AKD katsayılarının belirli bir katsayıyla çarpılması, tüm frekansların yoğunluk değerlerinin değiştirilmesi anlamına gelmektedir. Frekans yoğunlukları arasındaki fark değiştiği için, uzaysal düzlemde gri seviyeleri arasındaki fark değişmektedir. Çarpımda kullanılan katsayı 1’den büyük olması durumunda fark artmakta ve

kontrast yükselmektedir. Katsayı 0 ile 1 arasında olması durumunda, frekansların arasındaki değer farkı azalmakta, dolayısıyla kontrast düşmektedir. Resim 2.2’de bu durum görülmektedir.

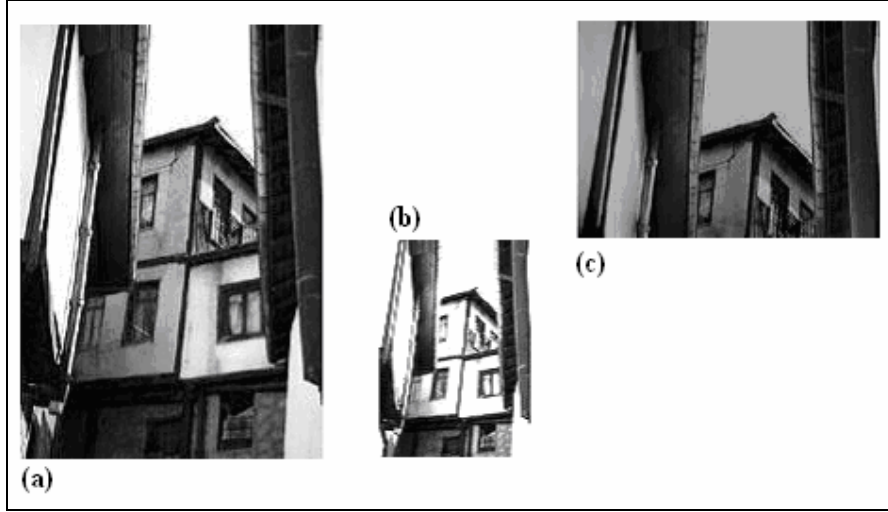
G ; görüntü, $AKD()$; AKD katsayıları hesaplama fonksiyonu, $AKD^{-1}()$; ters AKD hesaplama fonksiyonu, K ; katsayı olarak ele alındığında; *yeni görüntü* = $AKD^{-1}(AKD(G) * K)$ işlemi uygulanarak görüntünün kontrastı değiştirilebilir.



Resim 2.2. (a) Orijinal görüntü (b) AKD katsayıları 2 ile çarpılmış durum (c) AKD katsayıları 0,5 ile çarpılmış durum

AKD katsayılarının belirli bir bölümünün ters dönüştürülerek yeni bir görüntü oluşturulması, orijinal görüntünün boyutunu değiştirmektedir. AKD katsayılarının bir bölümü ters dönüşümde kullanılmazken, belirli bir frekans bölgesi göz ardı edilmiş olur. Bu da görüntünün gri seviyelerinin, yani parlaklığının da değişmesi anlamına gelir. Resim 2.3’de orijinal (a) görüntüsünün AKD katsayıları hesaplandıktan sonra, AKD katsayılarının satır ve sütunlarından yarısı ters AKD fonksiyonu ile geri dönüştürülerek (b) görüntüsü elde edilmiştir. Dolayısıyla ortaya çıkan sinyalin büyüklüğü de, AKD katsayıları kadardır. Bu da görüntünün boyutunun satır ve sütunlarda yarıya inmesi demektir. Yüksek frekans bölgelerindeki değerler yerine, orta frekanslardaki değerler yüksek frekans olarak ele alınmıştır. Bu da yüksek frekans değerlerinin artması anlamına gelmektedir. Bu yüzden görüntünün parlaklığı

artmıştır. AKD katsayılarının satırları yeniden örneklenerek yarıya düşürüldüğünde ise, görüntünün yarısı elde edilmektedir. (c) görüntüsü bu işlemin sonucudur.

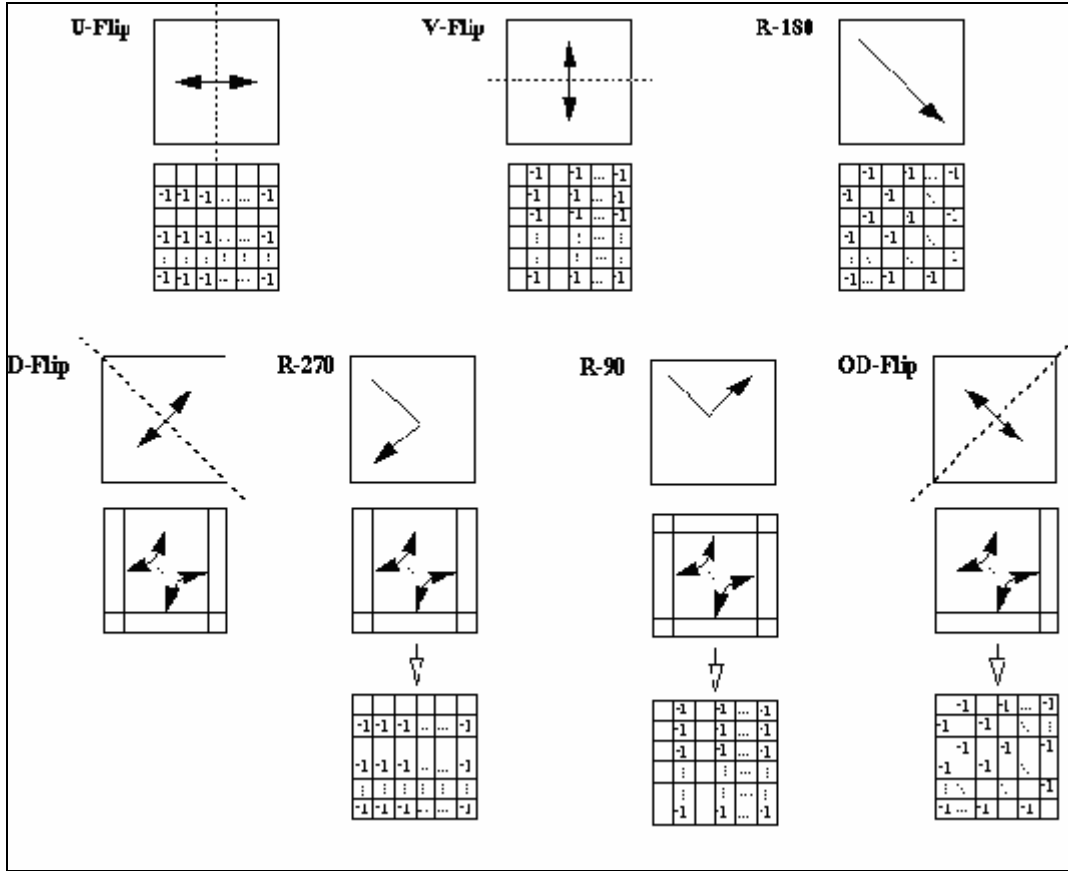


Resim 2.3. (a) Orijinal görüntü (b) AKD katsayılarının satır ve sütunlarından ilk yarısı ters dönüşümde kullanılarak elde edilmiş görüntü (c) AKD katsayılarının satırların ilk yarısının yeniden örneklenerek ters dönüşümde kullanılarak elde edilmiş görüntü

Shen ve Sethi (1995), çalışmalarında AKD katsayıları üzerinde yaptıkları değişikliklerle, görüntünün nasıl döndürüleceğini anlatmaktalar [9]. Çizelge 2.3'te yapılan işlemler tablosu görülmektedir. Şekil 2.9'da ise bu işlemlerin görüntü üzerindeki etkileri gösterilmiştir.

Çizelge 2.3. Shen ve Sethi'nin AKD işlem tablosu [9]

Uzaysal düzlemdeki işlem	Matematiksel açıklaması	AKD katsayıları değişimi
Değişim yok	$G(u,v) = F(u,v)$	İşlem yapılmaz
Kolonların yer değişimi (U-Flip)	$G(u,v) = \cos(u\pi)F(u,v)$	Katsayıların bir satır aralıklarla işaretlerinin değiştirilmesi
Satırların yer değişimi (V-Flip)	$G(u,v) = \cos(v\pi)F(u,v)$	Katsayıların bir kolon aralıklarla işaretlerinin değiştirilmesi
180° döndürülmesi	$G(u,v) = \cos(u\pi)\cos(v\pi)F(u,v)$	Katsayıların bir eleman aralıklarla işaretlerinin değiştirilmesi
Diagonal döndürme (D-Flip)	$G(u,v) = F(v,u)$	Katsayıların satırların karşılığı kolonla yer değiştirilmesi
270° döndürülmesi	$G(u,v) = \cos(u\pi)F(v,u)$	Satır ve sütunların yer değiştirilmesi ve bir satır aralıkla işaretlerin değiştirilmesi
90° döndürülmesi	$G(u,v) = \cos(v\pi)F(u,v)$	Satır ve sütunların yer değiştirilmesi ve bir kolon aralıkla işaretlerin değiştirilmesi
Ters diagonal döndürme (OD-Flip)	$G(u,v) = \cos(u\pi)\cos(v\pi)F(u,v)$	Satır ve sütunların yer değiştirilmesi ve bir eleman aralıkla işaretlerin değiştirilmesi

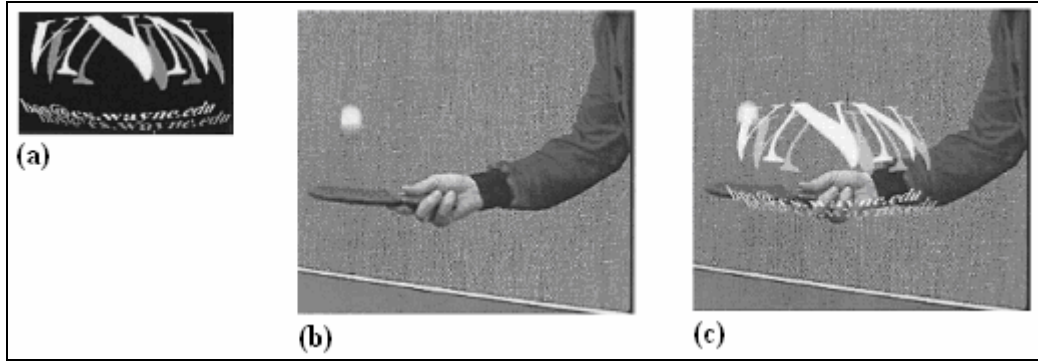


Şekil 2.9. AKD boyutunda görüntü döndürme

Hu ve Panchanathan (1998), resim ve video görüntülerinin AKD katsayıları üzerinden işlem yaparak genişletilmesi ve küçültülmesini incelemiştir [10]. Böylelikle, JPEG, MPEG ve H.261 gibi standart kodlayıcıları kullanan cihazlar, AKD katsayılarını geri döndürmeye gerek duymadan görüntüyü küçültüp büyütebilecekler ve daha efektif bir görüntü aktarımı sağlamış olacaklardır.

Shen ve arkadaşları (1998), AKD düzleminde, AKD'nin simetriklik ve ortogonalite özelliklerini kullanarak konvolüsyon teoremi geliştirmişlerdir [11]. Bu çalışmada uzaysal düzlemde iki görüntünün piksel piksel çarpımlarının, AKD düzleminde AKD konvolüsyon çarpımına denk olduğu gösterilmiştir. Bu denkleğe bağlı olarak Alfa Karıştırma yöntemi, AKD düzleminde konvolüsyon çarpımı ile uygulanmış ve iki görüntü karıştırılmıştır. Aynı şekilde MPEG biçimindeki bir video görüntüsü için, ara çerçeveye uygulanarak video görüntüsü içerisine yazı yazılmıştır (Resim 2.4). Bu

işlemlerin yapılması sırasında AKD katsayılarının geri dönüştürülmesi uygulanmadığından, uzaysal düzlemde işlem yapan görüntü ve video editörlerine göre daha yüksek performans elde edilmiştir.



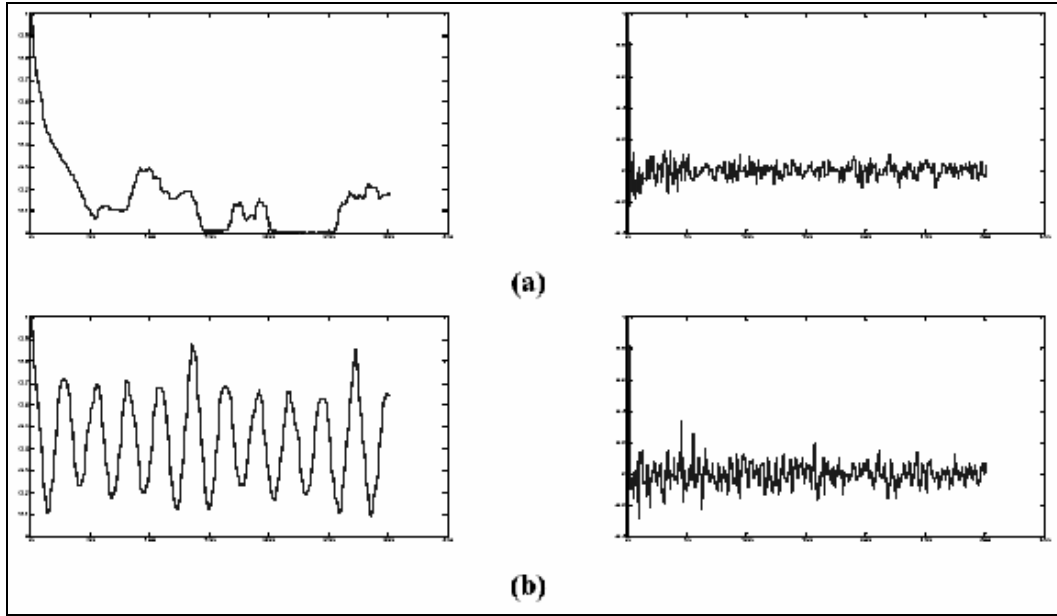
Resim 2.4. AKD konvolüsyon ile görüntü karıştırma (a) logo görüntüsü (b) video görüntüsü (c) birleştirilmiş görüntü [11]

2.6. AKD Özellikleri

Bu bölümde AKD'nin matematiksel özellikleri ve bu özelliklerin görüntü üzerindeki etkileri incelenmektedir [12].

2.6.1. Dekorelasyon (İlişkilerin azaltılması)

Görüntü işleme uygulamalarında dönüşümün amacı, yakın pikseller arasındaki benzerliklerin ve aynılıkların ortadan kaldırılması için uygun bir veri deseni ortaya çıkarabilmektir. Böylelikle dönüşüm katsayıları bağımsız olarak kodlanabilecektir. Aşağıdaki grafiklerde, dönüşüm uygulanmış ve uygulanmamış olan sinyallerin, otokorelasyon sonuçları görülmektedir.



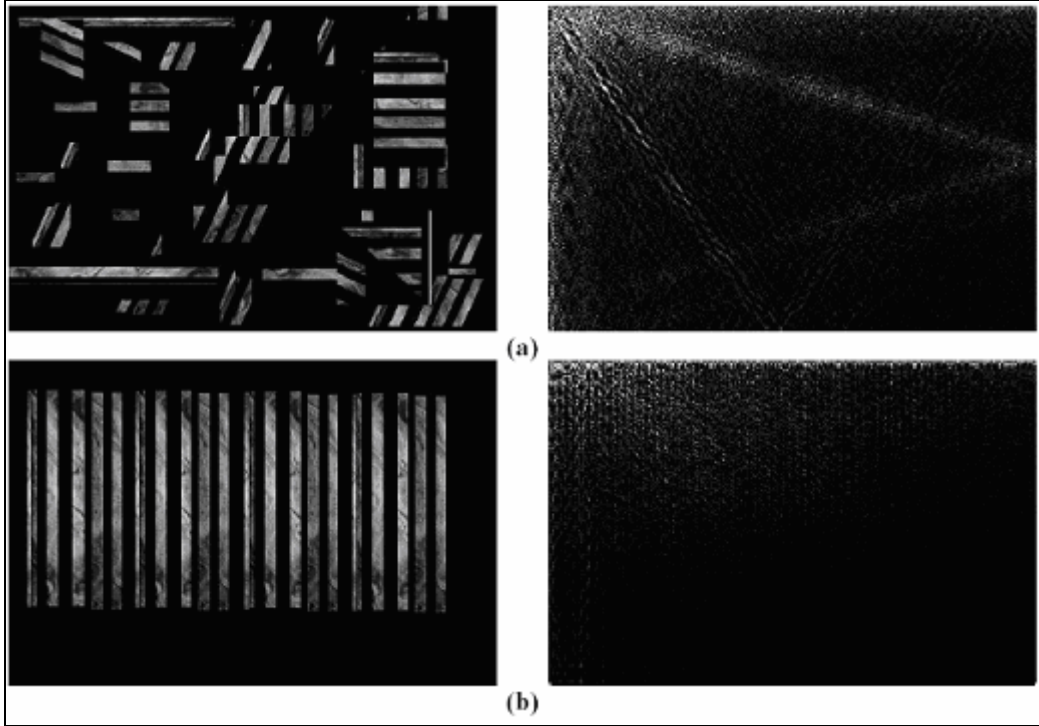
Şekil 2.10. (a) İçeriği ilişkisiz bir görüntünün, uzay düzleminde ve AKD sinyalinin normalize edilmiş otokorelasyon sonucu (b) İçeriği ilişkili bir görüntünün, uzay düzleminde ve AKD sinyalinin normalize edilmiş otokorelasyon sonucu [12]

Şekil 2.10'a göre, AKD katsayılarında, görüntünün uzaysal düzlemdeki değerlerine göre benzerliklerin büyük ölçüde ortadan kalktığı görülmektedir.

2.6.2. Enerji sıkıştırma

Sıkıştırma işlemi için bir dönüşümün etkisi, girdi sinyalini en iyi şekilde ve mümkün olduğunca az katsayı ile ifade edecek şekilde biçimlendirmesine bağlıdır. Bu sayede sıkıştırma fonksiyonları tarafından, dönüştürülmüş sinyalin çok küçük katsayıları ihmal edilerek, görüntü üzerinde insan algısının dikkatini çekmeyecek farklılıkları ortadan kaldırılarak, daha fazla sıkıştırma uygulanabilecektir. AKD bir görüntüyü, ağırlıklı olarak düşük frekans katsayılarıyla ifade edebilir. Görüntünün, dönüşüm sonrasında hangi aralıkta daha fazla ifade edileceği ise görüntü içeriğindeki değişimlere bağlıdır.

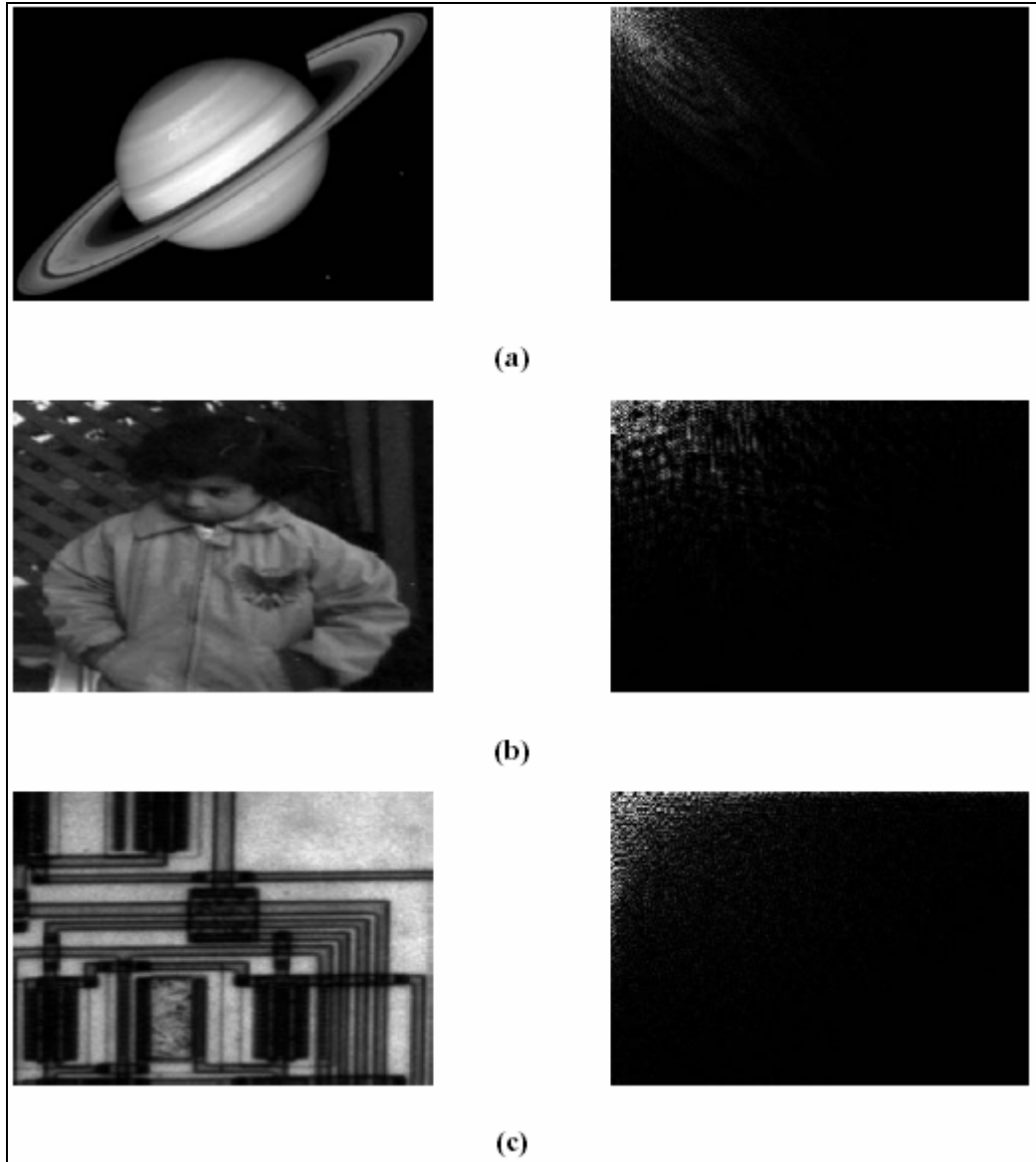
Şekil 2.11'deki grafikte iki görüntünün AKD sonuçlarının, görüntü içeriğindeki değişim ve periyodikliğe bağlı olarak nasıl farklılaştığı görülmektedir.



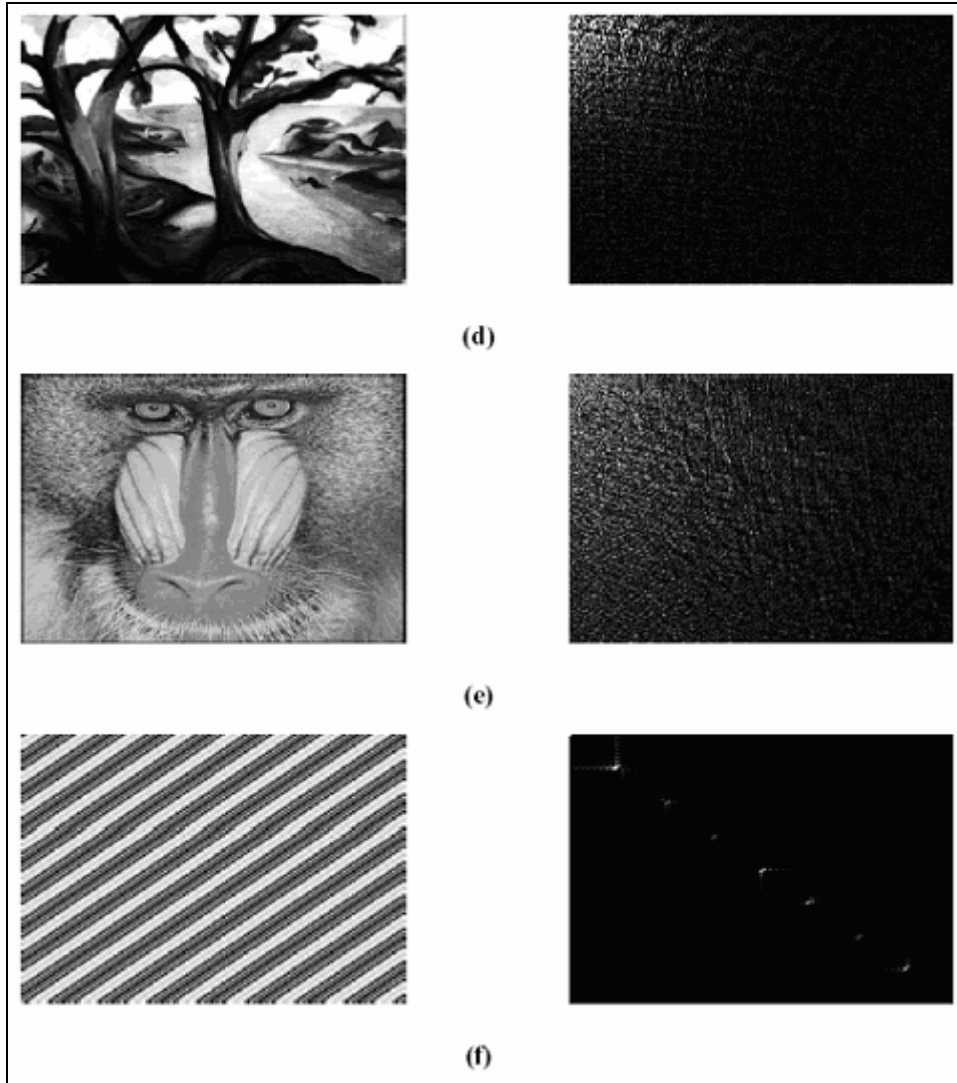
Şekil 2.11. (a) İçeriği ilişkisiz görüntünün AKD sinyali, (b) içeriği ilişkili görüntünün AKD sinyali [12]

Şekil 2.11'e göre, düzenli ve daha periyodik olan (b) görüntüsünün enerjisi düşük frekans bölgesinde yoğunlaşmıştır. Enerji yoğunluğu yaşanan bölge dışında kalan veriler kullanılsa dahi, geri dönüşümde görüntü kalitesi, insan gözünün algılayabileceği kadar değişmemiş olacaktır.

Resim 2.5'deki grafiklerde enerjinin nasıl sıkıştırıldığı, ya da sıkıştırılmadığı görülmektedir.



Resim 2.5. (a) Satürn ve AKD'si (b) Çocuk ve AKD'si (c) Devre ve AKD'si (d) Ağaç ve AKD'si (e) Maymun ve AKD'si (f) Bir sinüs dalgası ve AKD'si [12]



Resim 2.5. (Devam) (a) Satürn ve AKD'si (b) Çocuk ve AKD'si (c) Devre ve AKD'si (d) Ağaç ve AKD'si (e) Maymun ve AKD'si (f) Bir sinüs dalgası ve AKD'si [12]

Resim 2.5'deki örnekleri yakından inceleyecek olursak; (a) ve (b) resimlerinde geniş alanda yavaş değişimler mevcuttur. Bu yüzden düşük frekans bölgesine iyi bir enerji sıkıştırma söz konusudur. (c) resminde geçiş basamakları fazla ancak içerik ayındır. Bu sebeple benzerlik yüksektir ve AKD bu benzerliği yok etmiştir. (d) ve (e) resimlerinde fazla sayıda farklı içerik ve yüksek frekans bulunmaktadır. Bu yüzden enerji sıkıştırma düşüktür ve yüksek frekanslara yayılmıştır. (f) resminde periyodiklik, dolayısıyla fazla değişim vardır ve AKD düşük frekanstan yüksek frekansa doğru belirli bölgelerde tepiler oluşturmaktadır.

Çizelge 2.4. Yatay piksel sayısı 128 olan üç farklı görüntünün enerji dağılımı/AKD katsayısı çizelgesi

Enerjinin Yüzdesi	90	91	92	93	94	95	96	97	98	99
Lena	1,008	1,086	1,289	1,617	2,070	2,648	3,734	5,648	9,211	18,211
Evler	3,946	4,511	5,027	5,755	6,832	8,130	9,957	12,880	18,033	29,163
Maymun	1,792	1,845	1,994	2,143	2,363	2,643	3,030	3,833	5,405	10,161

Üç farklı görüntü alınarak, her bir satırı 1 boyutlu sinyal olarak düşünülmüştür. Tüm görüntüler için yatay piksel sayısı 128'dir, yani sinyallerin örnekleme sayısı 128'dir ve bu sinyallerin AKD uzayındaki katsayılarının kaç tanesi ile sinyalin enerjisinin ne kadarının tanımlanabildiği hesaplanmıştır (Çiz. 3.4). Tüm satırların ortalaması alınarak tablodaki değerler oluşturulmuştur. Örneğin Lena görüntüsünün enerjisinin %99'unu ifade edebilmek için, AKD katsayılarının %18,211'i yeterli olmaktadır. Bu çizelge AKD'nin enerjiyi nasıl sıkıştırabildiğini göstermektedir.

2.6.3. Ayrılabilirlik

2B AKD fonksiyonu aşağıdaki şekilde de tanımlanabilir:

$$C(u, v) = \sqrt{\frac{4}{NN}} k(u)k(v) \sum_{x=0}^{N-1} \cos\left[\frac{(2x+1)\pi u}{2N}\right] \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2y+1)\pi v}{2N}\right] \quad (3.1)$$

$$u, v = 0, 1, 2, \dots, N-1$$

Bu sayede, öncelikle y ekseninde, sonra x ekseninde ayrı ayrı 1B AKD işlemi uygulanarak 2B AKD elde edilir. Aynı işlem daha fazla boyutlu sinyaller üzerinde de uygulanabilir. Bu özellik, kolay hesaplama, ve gerektiğinde hesaplamanın basitleştirilmesine olanak sağlamaktadır. Görüntü işleme uygulamaları açısından, matematiksel hesaplamaları kolaylaştırarak performans sağlamaktadır.

2.6.4. Simetri

Ayrılabilirlik özelliğinde tanımlanan fonksiyon (Eş. 3.1) incelendiğinde, satır ve

sütunlar için uygulanan işlemin fonksiyonel olarak aynı olduğu görülmektedir. Bu tip dönüşümlere “simetrik dönüşüm” denilir. Bu sayede, dönüşüm önce satırlar, sonra sütunlar üzerinde yapılabileceği gibi, önce sütunlar, sonra satırlar üzerinde de yapılabilir.

Ayrılabilir ve simetrik bir dönüşüm, $T=AfA'$ şeklinde gösterilebilir. A , elemanları $a(i,j)$ olan ve Eş. 3.2'de gösterilen dönüşüm matrisidir. f ise görüntü matrisidir.

$$a(i, j) = \sqrt{\frac{2}{N}} k(j) \sum_{j=0}^{N-1} \cos \left[\frac{(2j+1)\pi i}{2N} \right] \quad (3.2)$$

Dönüşüm matrisinin, görüntü matrisinden bağımsız olması, özellikle gerçek zamanlı uygulamalarda, önceden hesaplanarak, anlık işlem miktarının azaltılması ve performansın yükseltilmesi anlamına gelmektedir.

2.6.5. Ortogonallik

$T=AfA$ eşitliğine göre ters dönüşüm fonksiyonu, $f=A^{-1}TA^{-1}$ şeklinde yazılabilir. AKD'yi oluşturan temel fonksiyonlar birbirlerine ortogondur, bu halde dönüşüm matrisinin transpozesi, tersine eşittir; $A^{-1}=A^T$

Bu özellik sayesinde, simetri özelliğinde anlatıldığı gibi ön hesaplama yapan uygulamalarda hesaplama kolaylığı sağlanmaktadır.

3. 3B AKD İLE GÖRÜNTÜ SIKIŞTIRMA

Sayısal görüntüde görüntünün boyutu uzaysal çözünürlüğünü, her bir pikselin alabileceği değer aralığı ise gri seviyesi çözünürlüğünü belirler. Böylelikle 500 piksel genişliğinde 500 piksel yüksekliğinde, ve 256 farklı renk değeriyle renklendirilen bir resmin, hafıza üzerinde kapladığı alan $500 \times 500 \times \log_2(256)$ bit olacaktır.

Hem hafızada görüntü tutmak için, hem de görüntünün kısıtlı kapasiteye sahip ağlar üzerinden gerçek zamanlı aktarımı için görüntü fazla yer kaplamaktadır. Bu sebeple görüntü sıkıştırma uygulanmaktadır.

Genel olarak veri sıkıştırma yöntemlerinin çalışması; verinin içerisindeki desenlerin kodlanması, veya artarda aynı verinin tekrarı bilgilerine dayanmaktadır. Böylece veri desenleri, desenin kapladığı alandan daha az yer kaplayan bir kod ile tanımlanır ve veri daha az yer kaplayacak şekilde ifade edilir.

Görüntü ve video görüntüleri, veri sıkıştırmak için uygun yapıya sahiptir. Çünkü resim içerisinde bir bölgede birbirine yakın ve aynı renk değerleri bulunur. Video görüntülerinde ise, artarda gelen iki kare içerisinde sadece hareket olan alanlardaki piksel değerlerinde değişiklik olur. Hareket olmayan veya az hareket olan bölgelerdeki piksel değerleri yakın veya aynı olur.

Görüntü bu özelliğiyle, kayıpsız sıkıştırma yöntemleri için de uygun bir veri yapısına sahipken, daha büyük oranda sıkıştırma uygulayabilmek amacıyla AKD kullanılır, ve sıkıştırma görüntünün frekans düzlemindeki gösterimi üzerinde uygulanır.

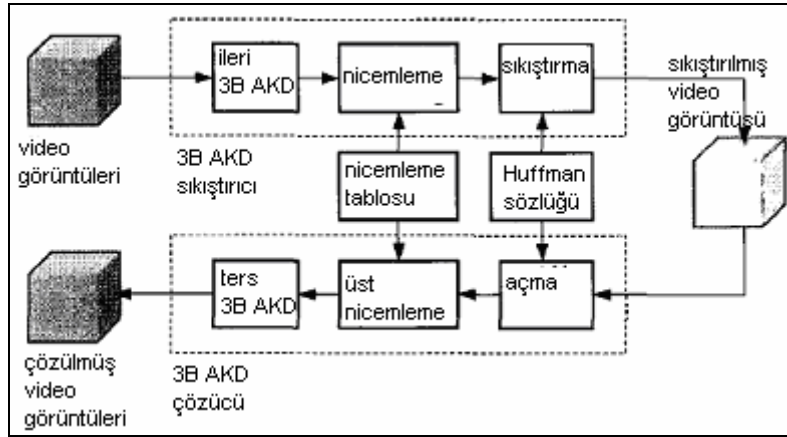
AKD ile sıkıştırma uygulamalarının en çok bilineni JPEG dir. JPEG sıkıştırma dört adımdan oluşmaktadır:

1. 2B AKD uygulanması
2. AKD üzerinde nicemleme yapılması
3. 2 boyutlu AKD katsayılarının, düzenli sıralama ile tek boyutlu hale

dönüştürülmesi

4. Tek boyutlu dizinin sıkıştırılması

Aynı yöntemler, görüntülerin artarda eklenerek, 3 boyutlu video görüntüsü üzerine de uygulanabilmektedir.



Şekil 3.1. 3B AKD sıkıştırma ve açma gösterimi [13]

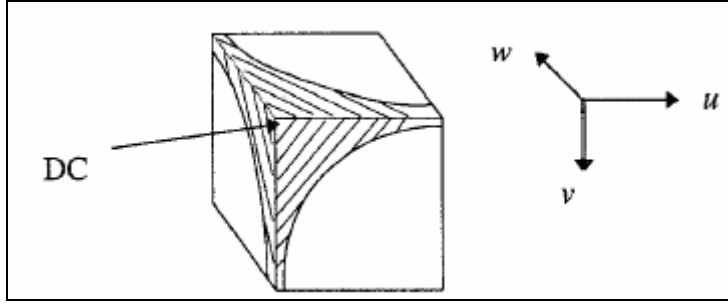
3.1. Orijinal Video Görüntüsü

Video görüntüsü artarda gelen karelerden oluşmaktadır ve her bir kare kendi başına 2 boyutlu bir görüntüdür.

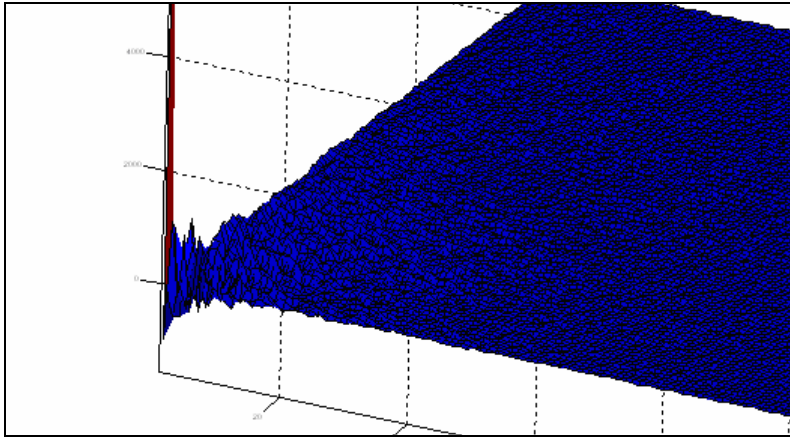
Orijinal görüntü YÜKSEKLİKxGENİŞLİKxZAMAN boyutunda, 3 boyutlu matris olarak ifade edilir.

3.2. Orijinal Video Matrisi Üzerinde 3B AKD Hesaplanması

Günlük yaşamdan bir sahnenin yer aldığı normal bir video görüntüsü içerisinde, geniş alana yayılmış yavaş renk değişimleri, ve normal hızda hareketler yer alır. Bu sebeple elde edilen 3B AKD fonksiyonunda enerji (0,0,0) noktasına yakın bölgelerde yoğunlaşır.



Şekil 3.2. AKD fonksiyonu enerjisinin yoğunlaşma bölgesi [1]

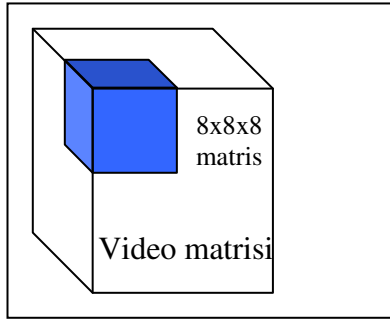


Şekil 3.3. Bir resmin 2B AKD grafiği

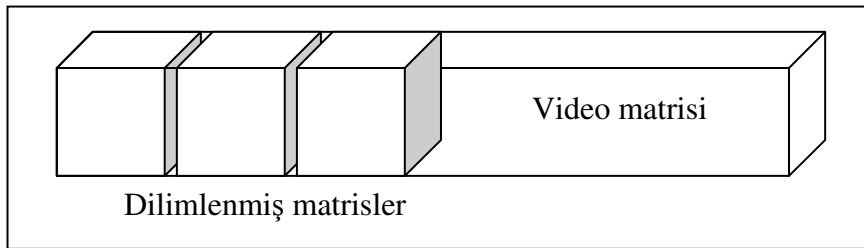
Şekil 3.2’de 3B AKD işlemi sonucu enerjinin yoğunlaştığı bölge gösterilmekte iken, Şekil 3.3’te gerçek bir resmin 2B AKD sonuç grafiğinden enerjinin (0,0) noktasında yoğunlaştığı görülmektedir.

3B AKD alma işlemi iki farklı şekilde yapılır:

Video görüntüsü matrisi Şekil 3.4’teki gibi $8 \times 8 \times 8$ ’lik küçük küp parçalarına bölünerek, küçük parçalar üzerinde işlem yapılır.



Şekil 3.4. Video görüntüsünün küçük bloklara bölünmesi



Şekil 3.5. Video görüntüsünde çerçevelerin gruplanması

Video matrisi, görüntü enine ve boyuna kesilmeyecek şekilde, zaman düzleminde N kadar çerçeve alınarak Şekil 3.5'teki gibi matrisler seçilir ve bu küçük parçalar üzerinde işlem yapılır.

3.3. 3B AKD Katsayılarının Nicemlenmesi

Aynı boyutta bir nicemleme matrisi kullanılarak, AKD fonksiyonunun değerleri, nicemleme matrisinde aynı indisle gösterilen elemanın değerine bölünür, sonucun tam sayı kısmı alınır. Geri dönüştürülmesi sırasında, çarpma işlemi yapılarak orijinal piksel değerine yakın bir değer elde edilir. Böylelikle birbirlerine yakın değerler yuvarlanarak aynı değere dönüştürülür. Örnek; nicemleme değeri=8, AKD katsayı değerleri=29,30,31,32,33,34,35,36, nicemlenmiş değerler=3.625, 3.75, 3.875, 4, 4.125, 4.25, 4.375, 4.5, tüm bu değerler 4 olarak alınır. Böylece 8 farklı birbirine yakın katsayı değeri aynı kabul edilir. Bu yöntemle daha fazla sıkıştırma uygulanabilecek bir veri deseni ortaya çıkmaktadır. Ancak geri dönüşüm sırasında tüm değerler $4 \times 8 = 32$ olarak dönüştürülecektir. Bu da veri kaybına yol açmaktadır.

Bu sebeple nicemleme, enerjinin yoğun olduğu bölgelerde nicemleme vektörü değerleri küçük alınarak daha az uygulanmaktadır. Enerjinin yoğun olmadığı bölgelerde yüksek nicemleme değerleri ile, AKD katsayı değerleri birbirlerine yakınlaştırılır veya sıfırlanır.

Nicemlemenin amacı, enerjinin yoğun olduğu bölgeleri fazla değiştirmeden, enerjinin düşük olduğu bölgelerdeki katsayı değerlerini aynılarmaktır. Nicemleme vektörü ters orantı içerisindeki sıkıştırma oranı ve görüntü kalitesi değerlerini belirleyen parametredir.

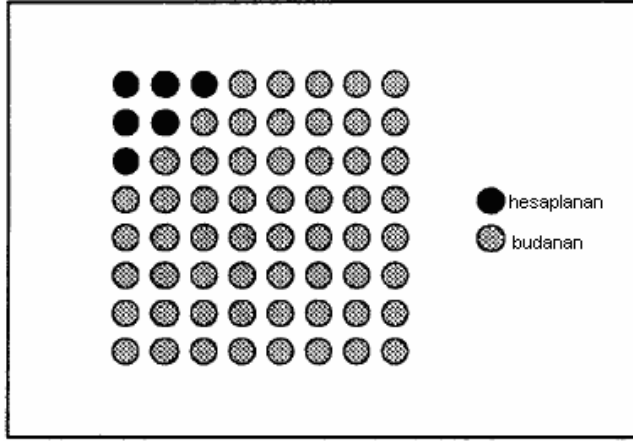
$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 47 & 49 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

Şekil 3.6. MPEG için 8x8'lik nicemleme matrisi [14]

Bu değerlere bakıldığında, (0,0) indisinden (8,8) indisine doğru gidildikçe her iki yönde değerlerin büyüdüğü görülmektedir. Günlük yaşamdan alınan video görüntüleri için, AKD fonksiyonu üzerinde, enerjinin yoğunlaştığı bölgeler (0,0) (düşük frekans bölgesi) olduğu için, bu bölgelerde nicemleme az yapılarak, verinin korunması sağlanmaktadır. (8,8) indisine yakın bölgede ise (yüksek frekans bölgesi) enerji az olduğundan bu kısımlar daha fazla nicemlenerek, değerlerin aynılanması sağlanır.

Nicemleme benzeri bir başka yöntem de, budamadır. Bir fonksiyon ile 3B AKD üzerinde enerjinin yoğun olduğu bölgelerdeki katsayıları değiştirmeden alınarak, geri kalanlar 0 yapılır. Böylelikle enerjinin düşük olduğu yüksek frekanslar silinecektir. Bu işlem gözün algılayamayacağı kadar düşük olursa kaliteyi de etkilemeyecektir. Kullanılacak bu fonksiyon $u+v+w < \text{SABIT}$ şeklindedir. Böylelikle her üç düzlem

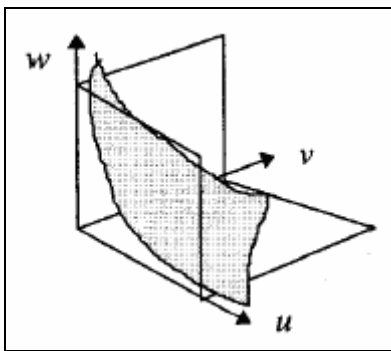
boyunca yüksek enerji bölgelerini kapsayacaktır.



Şekil 3.7. 8x8'lik matrisin budama değeri=3 olarak budanması [2]

Çizelge 3.1. Budama işlemi sonuçları [2]

	1	2	3	4	5	6	7	8
kazanılan zaman %	48,0	40,0	34,0	27,6	19,7	14,8	9,60	2,18
RMS hata değeri	38,5	29,2	24,1	20,9	18,6	17,7	15,5	13,6
sıkıştırma oranı artışı	6,90	3,50	2,28	1,68	1,36	1,16	1,06	1,00



Şekil 3.8. 3B AKD fonksiyonu üzerinde $u+v+w=SABIT$ fonksiyonu yüzeyi [1]

Nicemleme işleminde, JPEG ve MPEG standartları belirlenen görüntü kalitesine göre tanımlanmış tablolar kullanılmaktadır. Farklı görüntüler için kullanılan bu tablolar tüm görüntülerde en iyi sonucu vermemektedirler. Bu yüzden literatürde, tüm

görüntülerin kendilerine en uygun tabloları kullanabilmeleri için geliştirilmiş yöntemler mevcuttur. Çoğunlukla adaptif yöntemler kullanılarak, en uygun tablo oluşturulmaya çalışılmıştır.

Chan ve Lee (1997), yaptıkları bir çalışmada, belirli bir tablo oluşturmak yerine nicemleme tablosu için uygun bir fonksiyon önermişlerdir [1]. Farklı görüntüler için AKD katsayıları incelenerek, AC ve DC katsayılarının dinamik aralıkları çıkarılmış ve AC katsayılarının dağılımının Laplace fonksiyonuna benzediği görülmüştür. Sonuç olarak Eş. 3.1'deki gibi üssel bir fonksiyon kullanılması önerilmiştir.

$$q(u, v, w) = \begin{cases} A_i \left(1 - \frac{e^{-\beta_i((u+1)(v+1)(w+1))}}{e^{-\beta_i}} \right) + 1 & f(u, v, w) \leq C \\ A_0 \left(1 - e^{-\beta_0((u+1)(v+1)(w+1))} \right) & f(u, v, w) > C \end{cases} \quad (3.1)$$

A, β ve C parametreleri nicemleme değerlerini etkilemektedirler ve farklı görüntülerde, görüntü kalitesini belirlemek için kullanılmaktadırlar.

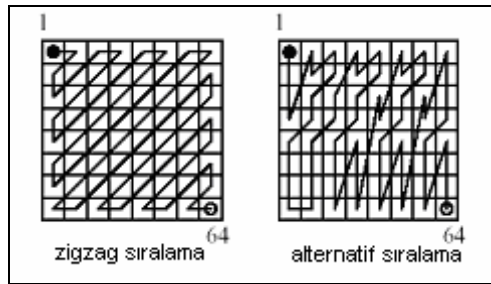
Westwater ve Furht (1996), çalışmalarında, her görüntünün içeriğine ait verilere göre istatistiksel ölçümler yaparak, bu ölçümlere dayalı olarak adaptif nicemleme tablolarını önermişlerdir [15].

Bhaskaran ve ark. (1997), taranmış metinlerin ve görüntülerin keskinleştirilmesi için JPEG boyutunda bir çözüm önermektedirler [16]. Bu yöntemin temelinde standart bir nicemleme tablosuyla nicemlenen görüntünün geri açılmasında, aynı nicemleme tablosu yerine belirli bir katsayı tablosuyla çarpılarak üretilmiş bir nicemleme tablosu kullanılmaktadır. Bhaskaran ve ark. geri açmada kullanılacak nicemleme tablosunu bulabilmek için, sentetik karakterle yazılmış yazıların AKD boyutundaki frekans yoğunluklarını, taranmış yazıların AKD boyutundaki frekans yoğunluklarıyla karşılaştırmış ve standart nicemleme tablolarını, bu iki verinin oranıyla çarpılarak elde etmişlerdir. Sonuç olarak bu yöntemle keskinleştirilmiş metin içerikli görüntülerin AKD katsayılarından yapılan frekans analizleri, sentetik yazıların frekans analizlerine daha çok benzediği görülmüştür.

3.4. 3B AKD Katsayılarının Sıralanması

Nicemleme yapılarak, katsayılar içerisinde aynı desenlerden ortaya çıkması sağlanmıştır. Böylelikle, desen kodlama ile sıkıştırma yöntemi (Huffman, aritmetik vb.) için uygun bir veri yapısı ortaya çıkarılmıştır. Sıralama yapılarak, ardarda gelen değerleri işleyerek sıkıştırma yapan yöntemler (Run-length vb.) için uygun bir yapı ortaya çıkarılmaya çalışılmaktadır.

Sıralama 3B AKD katsayılarının belirli bir sırayla dizilerek, bir vektör haline getirilmesidir. Bu sıralama için, sıra numarası belirten matrisler kullanılmaktadır.



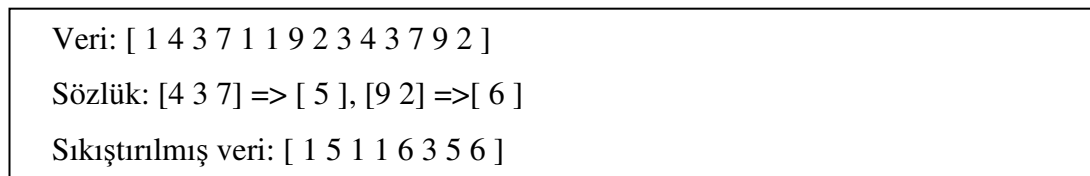
Şekil 3.9. MPEG’de kullanılan iki farklı sıralama matrisi [14]

Chan ve Lee (1997), nicemleme tablosunun adaptif oluşturulması için önerdikleri yöntemde kullandıkları üssel fonksiyonu (Eş. 3.1), sıralama için de kullanmayı önermişlerdir. Sıralama yönteminde önemli olan, nicemlenen katsayıların birbirine yakın ve aynı olan değerlerinin ardarda gelecek şekilde bir vektör haline dönüştürülmesidir. Chan ve Lee’nin Eş. 4.1’de önerdikleri fonksiyonda $A_i < A_0$ alındığında büyük AC katsayıları C yüzeyinin iç kısmında kalmakta, nicemleme sonrası 0 olabilecek daha küçük katsayılar ise C yüzeyinin dışında kalmaktadırlar. Böylelikle, sıralama amacıyla aynı fonksiyon kullanılması konumunda 0 olan katsayılar, kodlamanın daha verimli olacağı bir desen oluşturabilmesi için sıralamada ardarda gelmektedirler. Bu da, sıkıştırmanın etkisini artırmaktadır [1].

3.5. Sıkıştırma

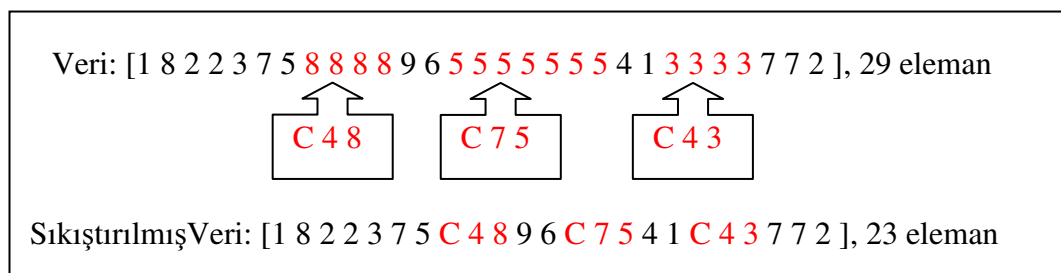
Son olarak elde edilen vektör, nicemleme ve sıralama sonrasında, aynı desene sahip ve artarda aynı değerlerden oluşan bir vektördür.

Huffman sıkıştırma yöntemi, en çok kullanılan veri desenlerini, ifade ettiklerinden daha küçük yer kaplayacak şekilde yeni kodlarla ifade edecek bir sözlük oluşturarak, desenleri bu kodlarla değiştirmek suretiyle, aynı veriyi daha küçük bir alanda ifade eder. Şekil 3.10'daki örnekte gösterilen 14 elemanlık dizi kodlanarak 8 elemanlık bir dizi olarak gösterilebilmektedir.



Şekil 3.10. Huffman kodlama

Run-length sıkıştırma yöntemi ise Şekil 3.11'de gösterildiği gibi, artarda gelen aynı değerleri, tekrarlanma sayısı ve değeri ile birlikte yazarak kodlar.



Şekil 3.11. Run-length sıkıştırma

Bu iki yöntem kullanılarak sıkıştırma yapılır, ve sonuç olarak daha az yer kaplayan bir vektör elde edilir.

Lakhani (2004), adaptif Huffman kodlama önermektedir [17]. Çalışmasının temelinde, Run-length kodlama sonucunda ortaya çıkan terimlerin Huffman

tablolarıyla kodlanırken, terimlerin her AC pozisyonunda ortaya çıkma ihtimalindeki farklılıkların kullanılması bulunmaktadır. Run-length kodlama sonucu ortaya çıkan terimlerin, AC pozisyonuna göre ortaya çıkma istatistikleri değerlendirilerek, bu istatistiki bilgiye göre 63 adet farklı kodlama tablosu oluşturulmaktadır. Böylece her kodlama tablosunda, terimlerin bir AC pozisyonunda ortaya çıkma ihtimali ne kadar yüksekse, o tabloda o terim için en düşük bit sayısı ile ifade edilen bir dizin kullanılmaktadır. JPEG yönteminde kullanılan Run-length kodlama gösterimine ufak bir değişiklik yaparak, (Sıfır adedi, Katsayı) gösterimi (Katsayı, Sıfır adedi) şekline dönüştürülmüştür. Bu sayede çözücü, hangi kodlama tablosunu kullanmasını gerektiğini bilmektedir. Tüm adaptif yöntemlerde karşılaşılan, oluşturulan tabloların veri ile birlikte sunulması gerekliliği ve bunun toplam dosya boyutunu artırması problemine çözüm olarak da, bu tabloların fark kodlama yöntemiyle tanımlanması önerilmiştir. Bu öneriye göre, tablolarda farklı değerler sol üst köşede toplanmaktadır ve tablolar birbirlerine benzemektedirler. Bu yüzden sadece ilk tablo tamamiyle ifade edilirken, diğer tablolardaki değerler bu tabloya göre farklarıyla ifade edilmektedir. Böylelikle çoğunluğu sıfır olan ve farklı değerleri küçük sayılarla ifade edilen tablolar elde edilmektedir. Bu tablo deseni de yine kolonlardaki sıfır sayılarına göre kodlanarak, az bir veriyle ifade edilmektedir. Sonuç olarak Çizelge 3.2’deki değerler elde edilmiş, ve Lakhani’nin önerdiği yöntemin, standart JPEG kodlama tablolarına ve dinamik Huffman tablolarına göre daha başarılı sonuçlar elde ettiği gösterilmiştir.

Çizelge 3.2. Lakhani’nin çalışmasının sonuçları [17]

1	2	3	4	5	6	7	8
Görüntü	JPEG K.5 Tablo kodlama	Jpeg Özel kodlama	Dinamik Huffmann kodlama	Aritmetik kodlama	Önerilen yöntem kodlaması	Kod tablosu gereksinimi	kazanç
	bit	%	%	%	bit	Bit	%
Yelkenli	213 173	0,29	0,14	0,81	184 787	6 278	10,97
Tiffany	150 839	3,04	2,28	3,49	127 740	7 404	11,25
Biber	162 930	1,77	1,26	2,63	140 232	6 543	10,70
Lena	154 984	1,15	0,69	2,12	134 406	5 977	10,25
Baboon	335 662	0,73	0,53	0,66	281 520	7 101	14,38
Uçak	161 470	0,63	0,31	1,43	141 893	5 901	9,27
Kız	249953	0,71	0,46	1,46	210357	5410	14,17

Çizelge 3.2. (Devam) Lakhani'nin çalışmasının sonuçları [17]

Zelda	193689	3,46	2,79	1,43	162538	4684	14,30
Barbara	333950	1,15	1,00	1,33	286465	8201	12,14
Ortalama	226045	1,38	1,01	1,93	192991	6376	12,02

Arazaki ve ark. (1991) Run-length kodlamada adaptif bir yöntem geliştirmişlerdir [4]. Bu çalışmada, AKD katsayıları, içerisinde sıfır taşıyan bölgeler ve sıfır olmayan bölgeler olarak ikiye ayrılmaktadır. Ve üç farklı Huffman tablosu kullanılmaktadır. 1. tablo sıfır taşımayan vektörlerin kodlanmasında, 2. tablo sıfır taşıyan bölgelerin Run-length kodlama sonrasında kodlanmasında ve üçüncü tablo da Run-length kodlanmış verinin kodlanmasında kullanılmıştır.

4. HAREKET TANIMA

Hareket, en az iki cismin birbirlerine göre uzaysal düzlemde pozisyonlarının zaman içerisinde değişmesidir. Hareket tanıma, hareket halindeki bir cisimi gösteren video görüntüsü veya yer değiştiren bir cismin belirli bir süre arayla alınmış iki görüntüsü kullanılarak, hareketin tanımlanmasıdır. Hareket sadece “var / yok” şeklinde tanımlanabileceği gibi, iki cisim arasında mesafeyi değiştiren yönde bir hareket söz konusu ise, yön ve hız olarak da tanımlanabilmektedir.

Görüntü işleme uygulamalarında genellikle, dünya üzerinde hareket eden bir cismin ardışık görüntülerinden, hareketin varlığının veya yönünün saptanması hesaplanmaktadır. Otomasyon sistemlerinde, alarm sistemlerinde, robotlarda ve takip sistemlerinde hareket tanıma kullanılmaktadır. Örnek olarak harekete duyarlı kameralar ile, güvenlik kameralarında belirli bir hareket başladıktan sonra kayıt alma işlemi uygulanarak, depolama ünitelerinden tasarruf sağlanmaktadır. Plaka tanıma ve yüz tanıma sistemlerinde kullanılan kameralar, harekete duyarlı hale getirilerek, sadece hareket anında fotoğraf çekerek, işlenen ve taşınan veri miktarını azaltmaktadırlar. Trafikte bir yaya veya aracın hareketinin takibi otomatik hale getirilerek, güvenlik birimlerine yardımcı bilgiler sağlayan yazılım ve donanımlar geliştirilmiştir. Aynı şekilde gelişen robot teknolojisi içerisinde, robotların karar verme mekanizmalarına, çevrelerindeki görüntü parametre olarak aktarılarak, insan hareketine daha yakın robotlar geliştirilmektedir. Aynı zamanda, video kodlayıcılar hareket tanıma algoritmalarını kullanarak, video görüntülerindeki aynılıkların sıkıştırılması için farklı yöntemler uygulamaktadırlar.

İnsan algılamasına göre düşünüldüğünde, en basit hareket tanımlama yöntemi, iki farklı görüntüyü, nesnelerin birbirlerine göre durumlarını karşılaştırarak elde edilmesidir. Görüntü işleme uygulamalarında, uzaysal düzlemde yapılan çalışmalarda en yaygın yöntem, bu bakış açısı üzerine kurulmuştur ve matematiksel olarak modellenmiştir. Bununla birlikte, görüntüler, sıkıştırılmış olarak saklama birimlerinde tutulmaktadır. Bu sebeple literatürde sıkıştırılmış görüntüler üzerinde hareket tanımaya yönelik olarak çalışmalar bulunmaktadır. Bu sayede, sıkıştırılmış

verilerin geri açılmasına gerek kalmadan, daha az performans gereksinimi ile hareket tanıma yapılabilmektedir.

4.1. MPEG

3B AKD sıkıştırma yöntemi günlük kullanım ihtiyaçlarını tam olarak karşılamamaktadır. Günlük kullanımda video görüntüsü daha çok CD, DVD üzerinde tutulmaktadır. Aynı şekilde sınırlı kapasiteli ağlar üzerinden görüntü aktarımı da oldukça yaygındır. Her iki yöntem için de, veri aktarımı düşük hızlarda yapılabilmektedir, ayrıca veri aktarımı sırasında hata oluşma ihtimali yüksektir.

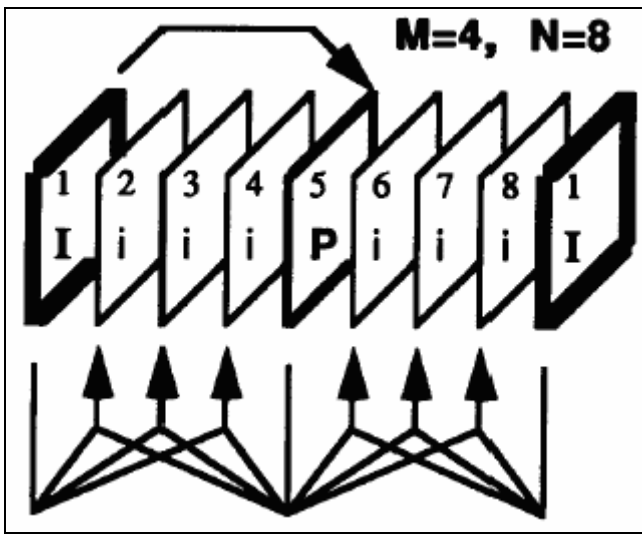
Bunun yanında günlük kullanımda önemli bir ihtiyaç ta, video görüntüsünü baştan sona kadar izlemek yerine, belirli bir noktadan izlemeye başlayabilmektir. Ayrıca aradaki bazı kareler üzerinde değişiklik yapabilmek, ya da yeni kareler ekleyebilmekte gerekmektedir.

Tüm bu ihtiyaçlardan dolayı, farklı algoritmalar geliştirilmiştir ve video standardı olarak kullanılmaktadır. Bunlardan en bilineni MPEG'dir (Motion Picture Experts Group).

MPEG video sıkıştırma algoritması, temel olarak iki basit tekniğe dayanmaktadır; artarda gelen karelerdeki benzerliğin sıkıştırılması için blok tabanlı hareket tamamlama, uzaysal düzlemdeki benzerliklerin sıkıştırılması için AKD tabanlı sıkıştırma. Hareket tamamlama teknikleri interpolasyon görüntüleri ile uygulanır ve hareketin geri kalanına dair kalan hata bilgisi uzaysal düzlem bilgisi ile birlikte AKD sıkıştırma içerisinde barındırılır. Harekete dair bilgi 16x16 blok üzerinde tutulur ve uzaysal düzlem bilgisi ile birlikte aktarılır. Hareket bilgisi maksimum verim alabilmek için değişken uzunluk kodlama ile kodlanır.

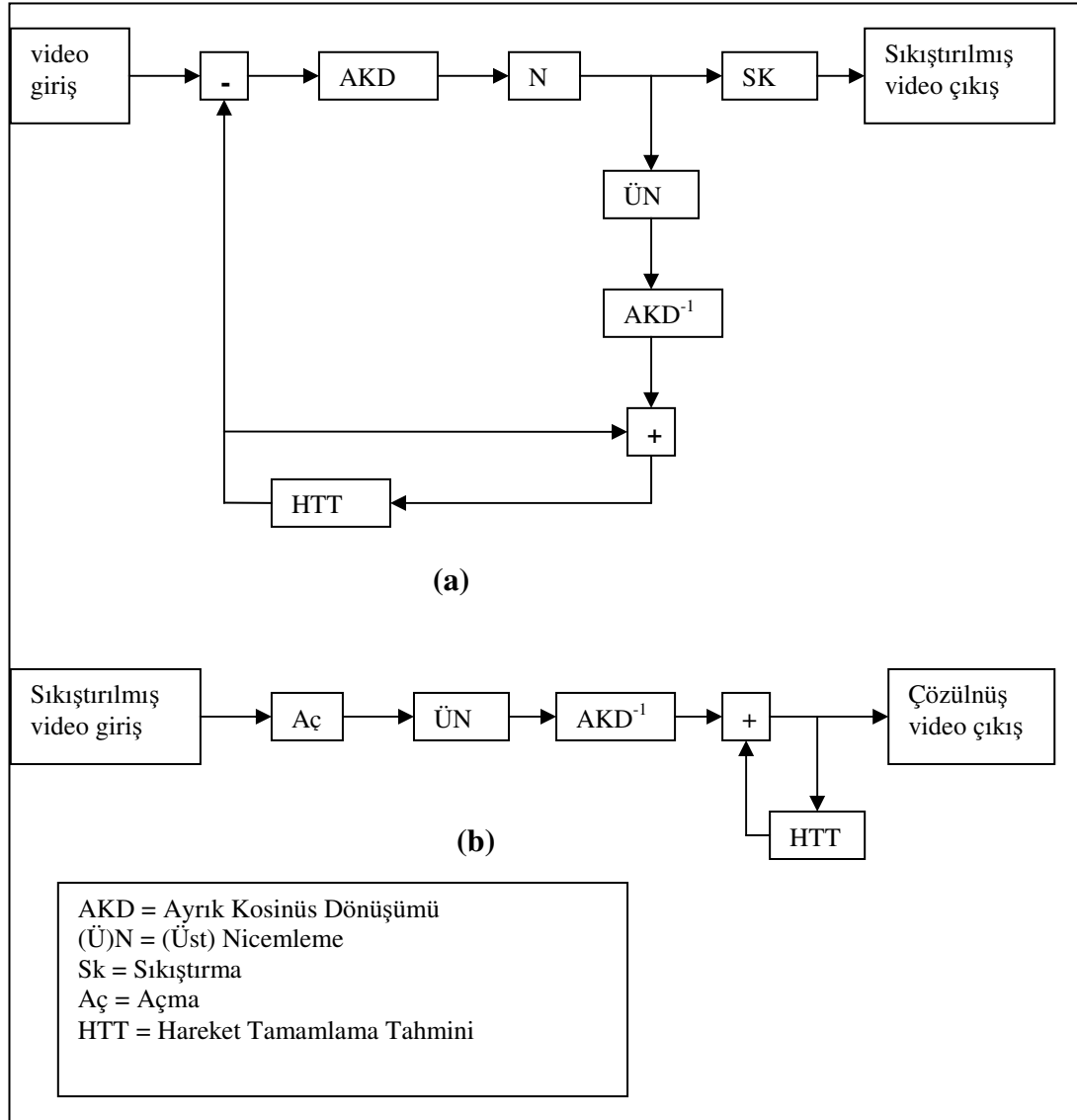
Saklanan video üzerinde rasgele erişim ve hareket tamamlama interpolasyonu tarafından uygulanan veri aktarımındaki azaltmanın öneminden dolayı, MPEG'de üç tip görüntü kullanılmaktadır; Intra picture, predicted picture, interpolated picture.

Intra picture; rasgele erişimde kullanılan erişim noktasını belirler sadece sıkıştırmayı yönetir. Interpolated picture, en yüksek sıkıştırmaya sahip görüntülerdir, geri oluşturulabilmesi için önceki ve sonraki karelere ihtiyaç duyar. Predicted picture ise geri oluşturulabilmesi için sadece kendisinden önce gelen karelere ihtiyaç duyar. Üç görüntü tipi arasındaki ilişki aşağıdaki şekilde gösterilmektedir. N ve M değerleri uygulamanın amacına özel olarak, rasgele erişim ve sıkıştırma oranı gereksinimlerine bağlı olarak parametriktrir.



Şekil 4.1. MPEG’de görüntü tiplerinin dizilimi [18]

Intra picture tipindeki görüntülerin uzaysal sinyali AKD ile sıkıştırılır. Uzaysal sinyal 8x8’lik kareler şeklinde frekans düzlemine AKD ile aktarılır. Frekans düzlemindeki sinyal adaptif kriter ile nicemlenir. Nicemlenmiş dönüşüm katsayıları, 2 boyutlu matristen tek boyutlu bir vektöre çevrilerek, Run-length ve Huffman sıkıştırma yöntemleriyle sıkıştırılır [18].



Şekil 4.2. MPEG (a) sıkıştırma ve (b) çözme fonksiyonları akış şeması [19]

4.2. Blok Eşleştirme Algoritması (BEA)

Bu algorithmada, iki görüntü arasında piksel değerlerine göre, karşılaştırılan alanın aynı olup olmadığına bakılır. Arama yapılan alanın seçilmesine göre matematiksel işlemi azaltmaya ve performansı yükseltmeye yönelik olarak farklı yöntemler geliştirilmiştir.

4.2.1 Tüm arama algoritması

Uzaysal düzlemde, hareket tanıma yapan bu algorithmada, referans görüntüdeki $N \times N$ 'lik bir pencere içerisindeki alan, arama yapılan görüntü üzerinde $N \times N$ 'lik alanlarla, piksel değerlerine göre karşılaştırılarak aynı görüntü bloğu aranır (Şekil 4.3).

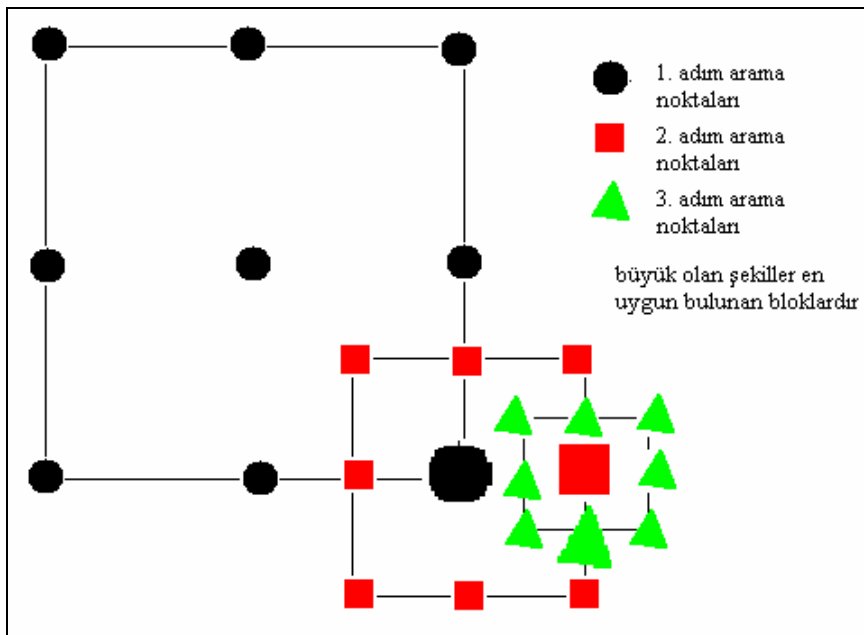


Şekil 4.3. BEA tüm arama uzayı

Arama penceresi, arama yapılan görüntü üzerinde tarama yapar. Bu yüzden çok sayıda matematiksel işlem gerektirir. Bunun yanı sıra nesnenin hareketi, görüntüleri alan kameraya göre, yatay veya dikey düzlemde hareket etmesi durumunda iyi sonuç elde edilir. Görüntü orijini etrafında dönmesi durumunda veya kameraya yaklaşması durumunda, arama penceresine göre açısı değişmiş veya büyümüş olacağından, referans görüntü üzerindeki alan ile aynı sayıda piksel değerlerine sahip olmaz, bu da algoritmayı yanıltır. Bu yüzden arama uzayını, açıya bağlı olarak ve z eksenini boyunca da genişletmek gerekir. Bu da karşılaştırma sırasındaki matematiksel işlemleri daha da karmaşık hale getirmektedir. Dezavantajlarına rağmen BEA, en temel mantığa sahip ve yaygın kullanılan bir yöntemdir.

4.2.2 Logaritmik arama

Bu arama yönteminde, $N \times N$ 'lik blok $p \times p$ 'lik bir arama alanı içerisinde arama yaparken, ilk adımda arama alanının çevresi üzerinde 8 noktada arama yapılır. Bu 8 noktanın merkezdeki referans görüntü bloğuna uzaklığı $d_1 = 2^{k-1}$, $k = \log_2(p+1)$ dir. İkinci adımda ise, birinci adımda en uygun blok, merkezde olacak şekilde aynı yöntem uygulanır, ancak arama blokları arasındaki uzaklık $d_2 = d_1/2$ dir [20].



Şekil 4.4. Logaritmik arama noktaları

Blok Eşleştirme Algoritmasında, arama alanlarının karşılaştırması için farklı matematiksel yöntemler kullanılmaktadır. En basit yöntem piksel farkları toplamıdır. Arama alanındaki her bir piksel değeri, karşılaştırılan alandaki aynı pozisyondaki piksel değerinden çıkartılarak, tüm farkların toplamı hesaplanmaktadır (Eş. 4.1). Sonucu 0'a en yakın olan karşılaştırılan alan, en çok benzerlik taşımaktadır.

$$SAD = \sum_{x=1}^{N_1} \sum_{y=1}^M |f_{mevcut}(x, y) - f_{referans}(x, y)| \quad (4.1)$$

4.3 AKD Düzleminde Hareket Tanıma

Uzaysal düzlemdeki veriler ile hareket tanıma uygulayabilmek için, en çok kullanılan MPEG, H.26x gibi kodlayıcılarla kodlanmış video görüntüleri, geri açılarak piksel değerlerini elde etmek gerekmektedir. Yazılım ve donanımlarda performansı yükseltebilmek için ters AKD hesaplaması yapmadan, AKD katsayıları üzerinden hareket tanıma yapabilmek için farklı çalışmalar yapılmaktadır.

AKD, kendi içerisinde sinyalin bazı özelliklerini taşımaktadır. Hareket tanıma uygulamalarında, AKD'nin sinyal ile ilgili taşıdığı bu özellikler ayrıştırılarak, hareket belirlenmeye veya tahmin edilmeye çalışılmaktadır.

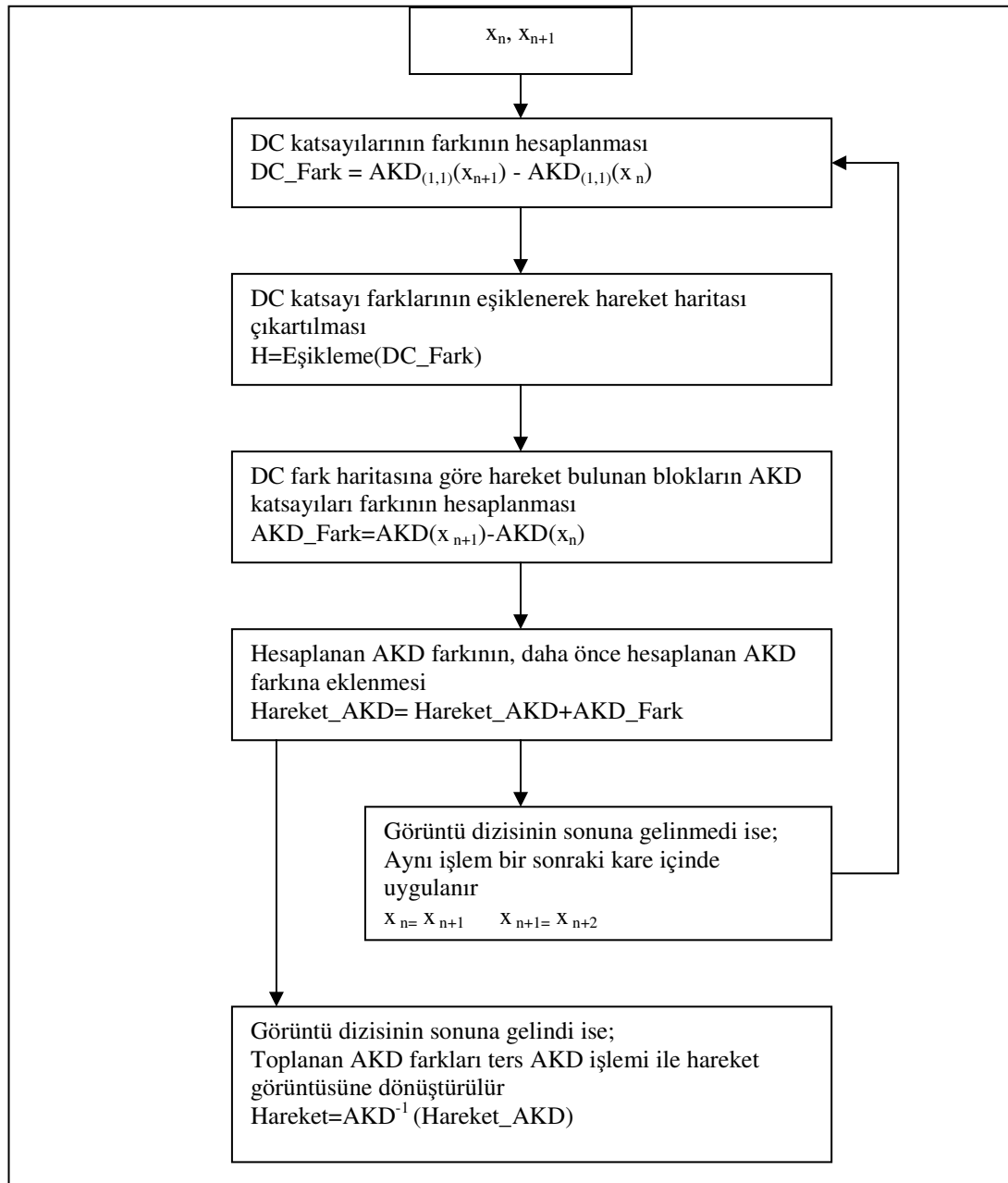
Peacock ve ark. (2001) AKD boyutunda toplama işleminin ayrılabilirlik özelliğini, uzaysal düzlemde fark alarak hareket tanıma yöntemiyle birlikte kullanarak, MPEG gibi kodlayıcılarda yüksek performans ile uygulanmasını önermişlerdir [21].

Fark alma yöntemine göre, hareket unsuru içeren bir video görüntüsünde ardarda gelen karelerin piksel değerlerinin farkları alınarak, hareketin kenarları belirlenebilir. Hareket eden cisim haricinde kalan bölgeler hemen hemen aynı piksel değerlerinden çıkartıldığında 0 değerine yaklaşmakta, hareket eden cismin hareket yönündeki kısımlarında ise, cisim ile arka plan arasındaki gri seviyesi farkı nedeniyle 0 dan çok farklı mutlak değerler elde edilmektedir. Tüm bu fark görüntüleri toplandığı zaman ise, sadece hareketin geçtiği bölgelerdeki piksel değerleri 0'dan farklı çıkmakta, ve kenarlar belirlenmiş olmaktadır. Peacock ve ark. ise Eş. 6.2'ye göre, MPEG içerisindeki ara çerçeve görüntülerinin AKD farkları alınarak, tek bir ters AKD işlemi uygulayarak bir serinin uzaysal düzlemdeki fark görüntülerinin elde edilmesini önermişlerdir.

$$AKD(A - B) = AKD(A) - AKD(B) \quad (4.2)$$

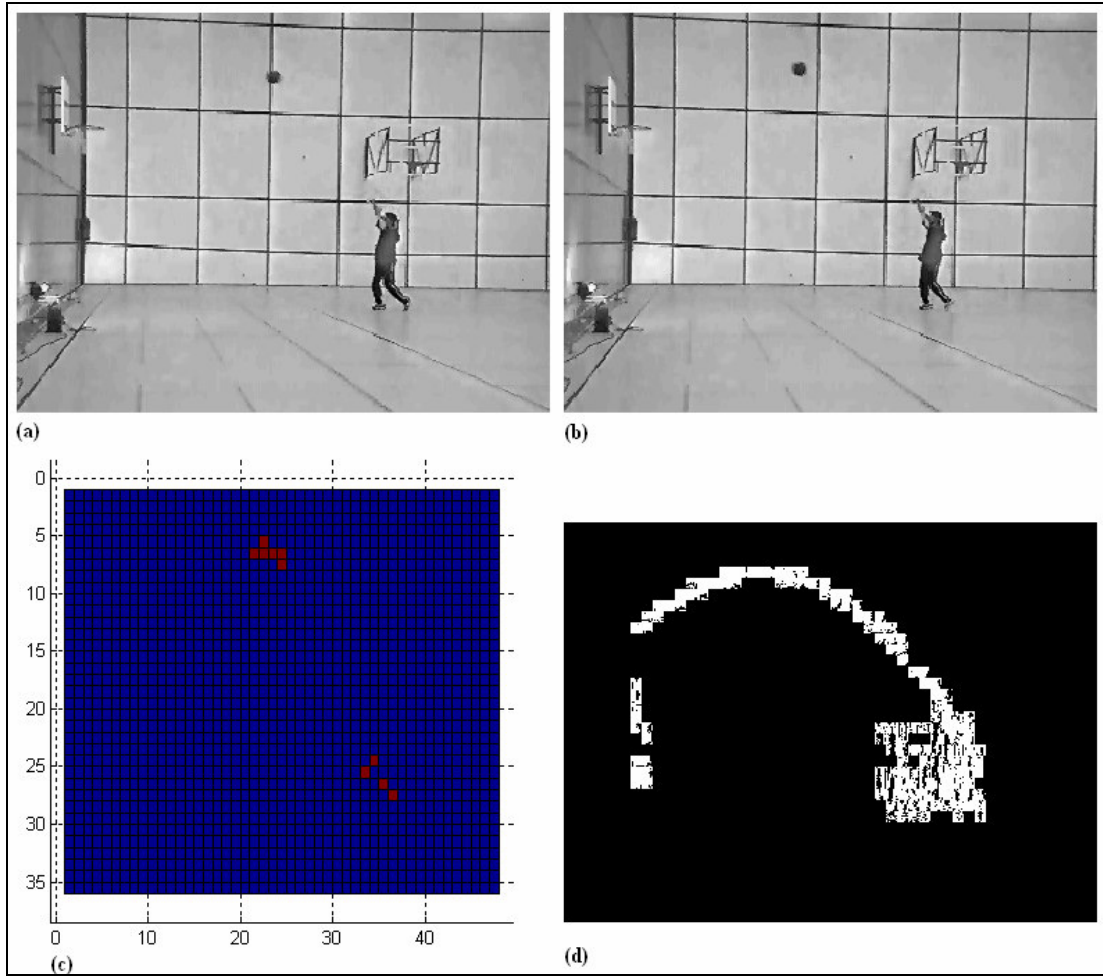
Bir AKD bloğu içerisinde DC katsayısı, bloğun luminans bilgisini vermektedir. Hareket içeren bir blokta hızlı bir değişim olacağından referans ve önceki

görüntülerin, aynı bloklarının DC katsayıları arasındaki yüksek orandaki değişim, o blok üzerinde bir hareket olup olmadığı bilgisini vermektedir. Bu sayede MPEG içerisindeki I görüntüleri ve JPEG içerisindeki görüntünün bir alanına ait 8x8'lik AKD bloklarının DC katsayıları karşılaştırılarak, o blok üzerinde hareket olup olmadığı saptanabilmektedir.



Şekil 4.5. AKD fark alma ile hareket tanıma akış şeması

Şekil 4.5'te AKD fark alma ile hareket tanıma akış şeması görülmektedir. JPEG ve MPEG kodlayıcılar görüntü karelerini 8x8'lik bloklara bölmektedirler. Ardışık iki görüntü karesine ait 8x8'lik blokların DC katsayılarının farkları hesaplanarak hareketin bulunduğu bloklar tespit edilir. Bir blokta değişim varsa DC katsayısı da yüksek oranda değişmektedir. Değişim ne kadar az ise DC katsayısındaki değişimde o oranda azdır. Hesaplanan DC farkları eşiklenerek, sadece yüksek oranda değişim yaşanan bloklar tespit edilmektedir. Bu sayede farklı hassasiyetlerde hareket tanıma yapılabilmektedir. Ayrıca kamera titremesi gibi durumlardan oluşan sabit arka planın titreterek oluşturduğu hareket etkisi göz ardı edilebilmektedir. Şekil 6.4 (c) görüntüsünde, (a) ve (b) görüntülerinden hesaplanmış, eşiklenmiş hareket haritası görülmektedir. Sadece hareket tespit edilen blokların AKD katsayıları, referans görüntüdeki aynı bloğa ait AKD katsayılarından çıkartılarak, uzaysal düzlemdeki fark alma işlemi, AKD ters dönüşüm hesaplanmasına gerek kalmadan yapılmaktadır (Bkz. Eş. 4.2). Hareket tespit edilmeyen bloklar için AKD katsayıları 0 kabul edilmektedir. Bu sayede elde edilen hareket görüntüsünde, sadece hareket bulunan alanlarda hareket izi bulunmakta, diğer alanlardaki titreşimler sonucu etkilememektedir. Aynı zamanda sadece hareket bulunan alanlarda AKD dönüşümü uygulandığından işlem sayısını azaltmaktadır. Aynı işlem sonraki görüntü kareleri içinde uygulanarak AKD farkları toplanır. Görüntü dizisi tamamlandığında elde edilen AKD farkları toplamının ters dönüşümü hesaplanarak, hareketin bulunduğu yol üzerinde siyah arka plandan farklı bir gri seviyesi değeriyle iz bulunan görüntü elde edilmektedir. Şekil 4.6 (d) görüntüsü, basketçi video görüntüsünün DC katsayılarından hareket tanıma ve AKD fark alma işlemi sonucunda elde edilen hareket görüntüsüdür. Basketbolcunun ileri geri yürüyüşü ile, topun hareketi açık bir şekilde görülmektedir. DC katsayıları karşılaştırmasında kullanılan eşiklemeden dolayı kameradaki titremeden dolayı meydana gelen arka plandaki hareket etkisi yok edilmiştir. Ancak eşikleme değerinin verdiği hassasiyetten dolayı, topun potaya girerek fileden çıkmasına kadar olan hareket yavaş olduğundan bu andaki hareket görülmemektedir.



Şekil 4.6. AKD fark alma ile hareket tanıma (a) önceki görüntü (b) referans görüntü (c) DC katsayı farkı haritası (d) hareket görüntüsü

Roma ve Sousa, Least Square Estimation yöntemini, AKD katsayıları ile kullanarak hareket tanıma uygulamışlardır [22]. Uzaysal düzlemde hareket tanıma için kullanılan LSE yönteminde kullanılan görüntünün türev parametrelerini, AKD katsayıları üzerinden hesaplayarak MPEG-x ve H26x kodlayıcıları için, sıkıştırılmış verinin açılmasına gerek kalmadan hareketi ve hareket vektörünü hesaplayabilmeyi sağlamaktadırlar.

Literatürdeki bazı hareket tanıma çalışmaları, doğrudan hareketin otomasyon sistemlerine aktarılması yerine, kodlayıcıların harekete bağlı olarak farklı sıkıştırma parametreleri ve algoritmaları kullanarak, sıkıştırma oranının yükseltilmesine yöneliktir.

Xue ve Fan (1996), Blok Eşleştirme Algoritmasına benzer şekilde hareketi, blokların aralarındaki hata değişimini hesaplayarak, hata değişimine bağlı olarak hareket vektörünü hesaplayan bir yöntem geliştirmişlerdir [23]. Bu yöntemde göre, referans görüntüdeki bloğun x ve y akisleri boyunca -15,+15 kaydırmayla mevcut görüntü üzerindeki diğer bloklarla arasındaki hata değeri hesaplanmakta ve uyumluluk karşılaştırması değerini belirleyen fonksiyon bu hata değişimi değerine bağlıdır. Böylece en uygun değeri veren vektör, hareket vektörü kabul edilmektedir. Yani, blok eşleştirme yöntemine benzer bir şekilde bloklar karşılaştırılmakta, ancak bloklar arasındaki benzerlik bilgisi AKD katsayılarından elde edilen hata değişiminden çıkarılmaktadır.

H.261 kodlamada, MPEG'de olduğu gibi ara çerçeveler kullanılmaktadır. Çerçevelerin tamamını sıkıştırmak yerine, yüksek orandaki çerçeve sayısı, aralarda belirlenen referans çerçevelerdeki görüntülerle aralarındaki farka göre tanımlanmaktadır. Xiu ve Fan'ın hareket tanıma yöntemi, benzer şekilde uygulandığından H.261 kodlayıcıların sıkıştırma performansında önemli bir etki sağlamaktadır.

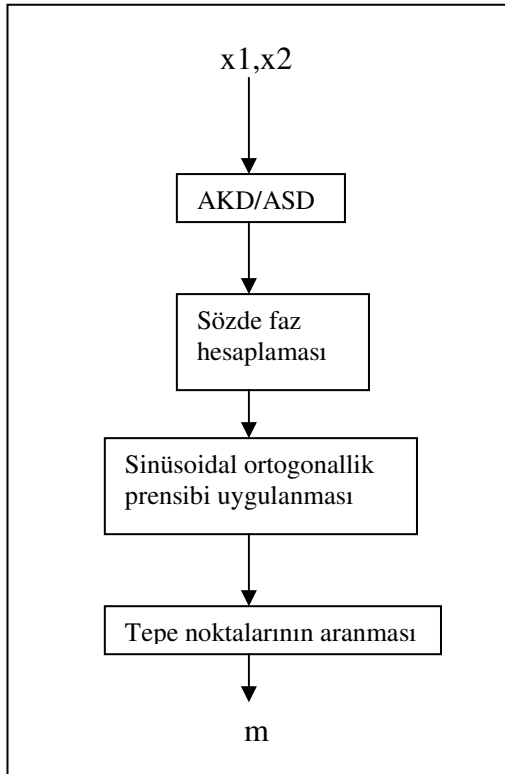
4.4 Sözde Faz Kullanarak Hareket Tanıma

Koc ve Liu, ilk olarak 1994 yılında yayınladıkları bir çalışmada [24], Ayrık Kosinüs ve Sinüs Dönüşümü kullanarak hareket tanıma yöntemlerini açıklamışlardır. 1998 yılında yayınladıkları çalışmada, yöntemlerini genişleterek yeni bir makaleyle anlatmışlardır [25].

Görüntü iki boyutlu sinyal olarak düşünüldüğünde, tek boyutlu sinyallerdeki ötelenme, görüntüdeki hareket unsuruna karşılık gelmektedir. Ötelenmiş bir sinyalin Fourier dönüşümü içerisinde ötelenme miktarı bilgisi faz bileşenine bağlı olarak taşınmaktadır. Ancak AKD veya ASD (Ayrık Sinüs Dönüşümü) içerisinde faz bileşeni yoktur. Koc ve Liu'nun çalışmasında, AKD ve ASD katsayılarından sözde faz bilgisi çıkartılmakta ve bu sayede ötelenme miktarı hesaplanabilmektedir. Sözde faz fonksiyonunu elde etmek için, sinüzoidal ortogonalite prensibi kullanılmaktadır.

4.4.1. 1B sinyaller için sözde faz hesaplama

AKD/ASD tabanlı hareket tanıma yöntemi, DXT-ME (Discrete Cosine/Sine Based Motion Estimation), öncelikle 1B sinyaller üzerine uygulanmıştır. Daha sonra 1 boyutlu üzerinde yapılan çalışma 2 boyutlu düzleme aktarılmıştır.



Şekil 4.7. Sözde faz ile hareket tanıma akış şeması

Şekil 4.7’de 1 boyutlu sinyaller için akış şeması görülmektedir. x_1 ; orijinal sinyal, x_2 ise x_1 ’in m kadar ötelenmiş halidir. Sözde faz hesaplaması Eş. 4.3’teki gibidir.

$$g_m^s(k) = \begin{cases} \frac{Z_1^C(k)X_2^S(k) - Z_1^S(k)X_2^C(k)}{[Z_1^C(k)]^2 + [Z_1^S(k)]^2}, & k \neq N \\ 1, & k = N \end{cases} \quad (4.3)$$

g_m^s ; sözde faz fonksiyonu, Z_1^C ; x_1 ’in 1. tip AKD’si, Z_1^S ; x_1 ’in 1. tip ASD’si, X_2^C ; x_2 ’nin AKD’si, X_2^S ; x_2 ’nin ASD’sidir.

Sözde faz fonksiyonu $\sin(k\pi/N)(m+0,5)$ 'e eşittir. Sonuç olarak sözde faz fonksiyonunun ters ASD'si hesaplanarak, gürültüsüz sinyaller için tek bir frekansta tanımlı sinüs fonksiyonunun ters dönüşümü olarak bir tepî fonksiyonu elde edilir. Gürültülü sinyaller üzerinde ise aynı tepî fonksiyonu gürültülü olarak elde edilir. Koc ve Liu çalışmalarında orijinal ve ötelenmiş sinyaller arasında PSNR=10dB gürültüde dahi ayırt edilebilir sonuç elde edildiğini göstermişlerdir.

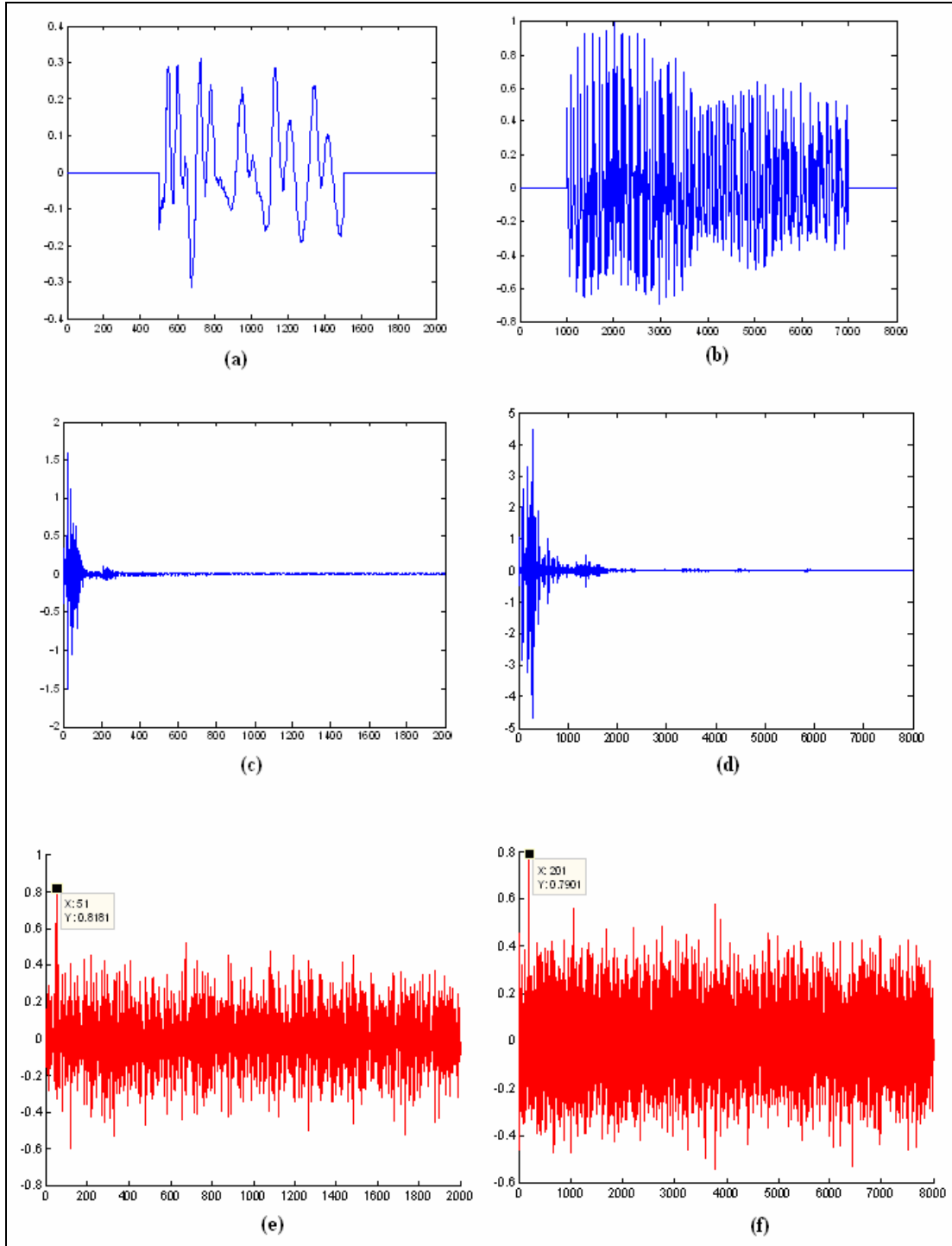
Ters dönüşüm sonucu elde edilen fonksiyonda zirve oluşan noktanın indisi öteleme miktarını vermektedir.

$$m = \begin{cases} i & ,d(i) > 0 \\ -(i+1) & ,d(i) < 0 \end{cases} \quad (6.4)$$

Eş. 6.4'e göre, $d(i)$ (tepinin değeri), pozitif yönde ise öteleme miktarı $+x$ yönünde i kadar, negatif yönde ise öteleme miktarı $-x$ yönünde $(i+1)$ kadardır.

Matlab'da sözde faz yöntemi uygulanarak, doğrulaması ve geliştirilmesi amaçlanmıştır. Gerçek bir ses dosyasından alınan örnek sinyaller ötelenerek ve üzerine gürültü eklenerek, DXT-ME yöntemi ile sinyaldeki öteleme bulunmaya çalışılmıştır. Orijinal sinyalin üzerine eklenen gürültü Eş. 6.5'teki formülle hesaplanmıştır.

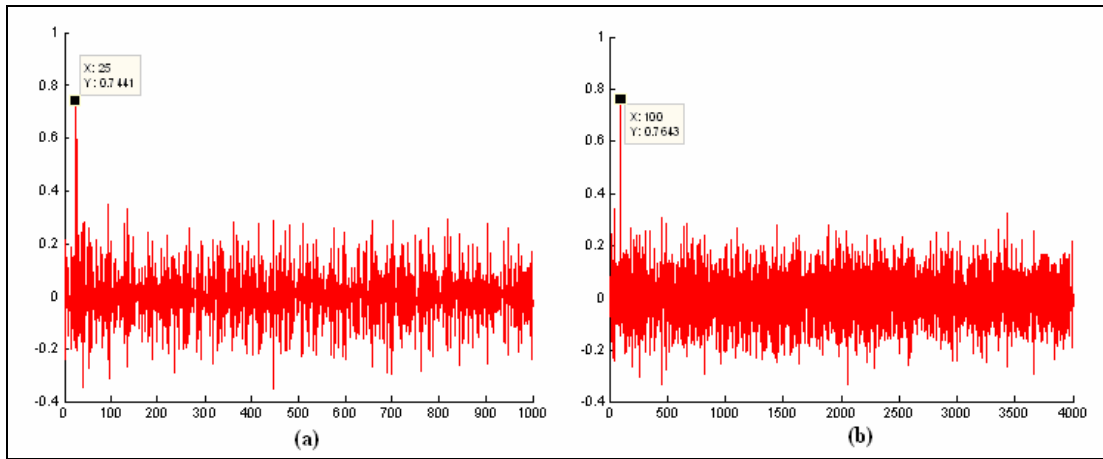
$$\text{PSNR hata} = 10 * \log_{10}(\text{sum}(\text{orijinal}.*\text{orijinal})/\text{sum}(\text{gurultu}.*\text{gurultu})) \quad (6.5)$$



Şekil 4.8. (a) x_1 sinyali, (b) x_2 sinyali, (c) x_1 sinyalinin AKD'si, (d) x_2 sinyalinin AKD'si, (e) x_1 sinyali üzerinde Koc ve Liu'nun yöntemiyle öteleme hesaplaması, (f) x_2 sinyali üzerinde Koc ve Liu'nun yöntemiyle öteleme hesaplaması

Şekil 4.8'de (a) ve (b) grafikleri sırasıyla x_1 ve x_2 sinyallerini göstermektedirler. (c)

ve (d) görüntüleri bunların AKD'lerini göstermektedir. Bu grafiklerden x2 sinyalinin yüksek frekanslarda x1 sinyaline göre daha fazla bileşen içerdiği görülmektedir. x1 sinyaline +50 öteleme uygulanmış ve eklenen gürültü sonucunda 13,8513 dB hata hesaplanmıştır. x2 sinyaline ise +200 öteleme uygulanmış ve eklenen gürültü sonucunda 11,4971 dB hata hesaplanmıştır. (e) grafiğinde x1 ve ötelenmiş sinyaline DXT-ME uygulanmıştır. 50 indisinde en yüksek genliğe sahip tepisi görülmektedir ve gürültüden ayırdedilebilmektedir. (f) grafiğinde x2 ve ötelenmiş sinyaline DXT-ME uygulanmış ve 200 indisinde ortaya çıkan tepinin genliği en yüksek tepisi olduğu görülmüştür. x2 sinyaline x1 sinyalinden daha fazla gürültü eklenmiş olmasına rağmen DXT-ME, x2 sinyali üzerinde x1 sinyaliyle benzer ayırdedilebilirlikte başarılı bir sonuç göstermektedir. x2 sinyalinin örnekleme sayısı daha fazla, ve daha yüksek frekansta bileşenler içermekte, yani sinyal daha fazla farklı bileşen içermektedir. DXT-ME yöntemi, iki sinyal arasındaki benzerliği frekans alanında incelemektedir. Benzerliklerin belirlenebileceği daha çok bilgi içermesinden dolayı x2 sinyali üzerinde daha başarılı sonuç vermektedir.



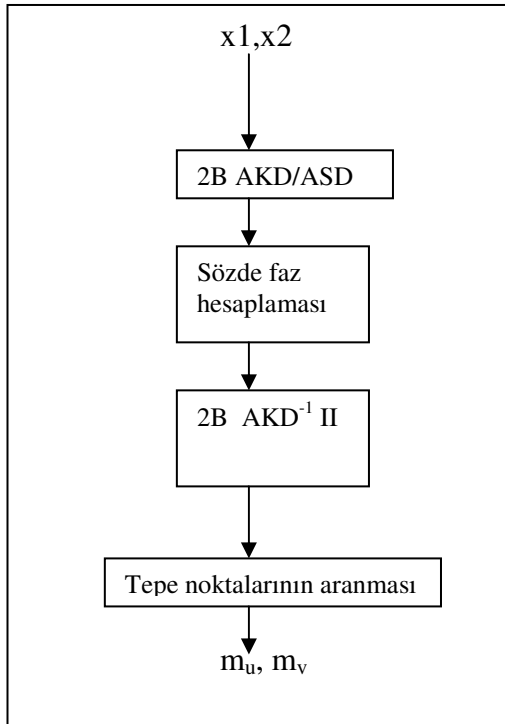
Şekil 4.9. (a) x1 sinyali üzerinde DXT-ME sonucu (b) x2 sinyali üzerinde DXT-ME sonucu

DXT-ME yönteminde, sözde faz fonksiyonu hesaplamasında kullanılan x1 ve x2 sinyallerinin AKD ve ASD katsayılarının, ilk yarısı hesaplama işlemine katılarak sonuçları araştırılmıştır. Frekans düzleminde çıkan katsayıların ilk yarısı alınarak, uzaysal düzlemde sinyalin örnekleme sayısı yarıya düşürülmüş, genliği artırılmış

olur. Bu sayede sinyalin içerdiği değişik frekanslardaki bileşen sayısı artmaktadır. Bu da sözde faz hesaplamasında, iki sinyal arasındaki benzerliğin tespiti için daha çok bilgi vermektedir. x_1 sinyali, +50 ötelenmiş ve üzerine eklenen gürültü ile PSNR hata oranı 13,7478 olan sinyal ile bu yöntem uygulanmıştır. Şekil 4.9 (a) grafiğinde sonucu görülmektedir. x_2 sinyali ise +200 ötelenmiş ve üzerine eklenen gürültü ile PSNR hata oranı 11,4867 olmuştur. Sonuç grafiği Şekil 4.9 (b) grafiğinde görülmektedir. sinyaller üzerindeki sıkıştırmadan dolayı öteleme miktarı yarıya indirilmiş olarak görülmektedir. x_1 için 25. indiste, x_2 için 100. indiste tepe oluştuğu görülmektedir. Bu da öteleme miktarındaki çözünürlüğü düşürmektedir. Örneğin +200 ve +201 öteleme miktarı için yöntem aynı sonuçları verecektir. Ancak tepe oluşan noktanın daha rahat okunabilir olduğu görülmektedir. Bu yüzden daha fazla gürültüde daha iyi sonuçlar elde etmek için kullanılabilmektedir.

4.4.2. 2B sinyaller için sözde faz hesaplama

İki boyutlu sinyaller için sözde faz hesaplaması akış şeması Şekil 4.10'daki gibidir.



Şekil 4.10. İki boyutlu sinyallere sözde faz uygulanması

İki boyutlu sinyaller üzerinde sözde faz hesaplaması yapılırken sinüs ve kosinüs dönüşümleri kullanılmaktadır. Bilinen hesaplama yöntemlerinden farklı olarak, birinci boyutta kosinüs, ikinci boyuta sinüs veya tam tersi kullanılarak dönüşüm hesaplanabilmektedir. x_1 sinyali t anındaki, x_2 sinyali ise $t+n$ anındaki görüntü ise, x_2 için Eş. 4.6 - Eş. 4.9'daki dönüşümler hesaplanmaktadır.

$$X_t^{CC}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cos\left[\frac{k\pi}{N}(m+0,5)\right] \cos\left[\frac{l\pi}{N}(n+0,5)\right] \quad (4.6)$$

$$k,l \in \{0,\dots,N-1\}$$

$$X_t^{CS}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cos\left[\frac{k\pi}{N}(m+0,5)\right] \sin\left[\frac{l\pi}{N}(n+0,5)\right] \quad (4.7)$$

$$k \in \{0,\dots,N-1\}; l \in \{1,\dots,N\}$$

$$X_t^{SC}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \sin\left[\frac{k\pi}{N}(m+0,5)\right] \cos\left[\frac{l\pi}{N}(n+0,5)\right] \quad (4.8)$$

$$k \in \{1,\dots,N\}; l \in \{0,\dots,N-1\}$$

$$X_t^{SS}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \sin\left[\frac{k\pi}{N}(m+0,5)\right] \sin\left[\frac{l\pi}{N}(n+0,5)\right] \quad (4.9)$$

$$k \in \{1,\dots,N\}; l \in \{1,\dots,N\}$$

x_1 için ise Eş. 4.10, Eş. 4.11, Eş. 4.12 ve Eş. 4.13'teki dönüşümler hesaplanmaktadır.

$$Z_{t-1}^{CC}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos\left[\left(\frac{k\pi}{N}\right)m\right] \cos\left[\left(\frac{l\pi}{N}\right)n\right] \quad (4.10)$$

$$k,l \in \{0,\dots,N\}$$

$$Z_{t-1}^{CS}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cos\left[\left(\frac{k\pi}{N}\right)m\right] \sin\left[\left(\frac{l\pi}{N}\right)n\right] \quad (4.11)$$

$$k \in \{0,\dots,N\}; l \in \{1,\dots,N-1\}$$

$$Z_{t-1}^{SC}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin\left[\left(\frac{k\pi}{N}\right)m\right] \cos\left[\left(\frac{l\pi}{N}\right)n\right] \quad (4.12)$$

$$k \in \{1, \dots, N-1\}; l \in \{0, \dots, N\}$$

$$Z_{t-1}^{SS}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \sin\left[\left(\frac{k\pi}{N}\right)m\right] \sin\left[\left(\frac{l\pi}{N}\right)n\right] \quad (4.13)$$

$$k, l \in \{1, \dots, N-1\}$$

x1 ve x2'ye ait dönüşüm hesaplamaları kullanılarak Eş. 4.14 ve Eş. 4.15'teki sözde faz fonksiyonları hesaplanmaktadır.

$$f_{mu,mv}(k,l) = \begin{cases} \hat{g}_{mu,mv}^{CS}(k,l) & k, l \in \{1, \dots, N-1\} \\ \frac{Z_{t-1}^{CC}(k,l)X_t^{CS}(k,l) - Z_{t-1}^{CS}(k,l)X_t^{CC}(k,l)}{Z_{t-1}^{CC}(k,l)^2 + Z_{t-1}^{CS}(k,l)^2} & k = 0, l \in \{1, \dots, N-1\} \\ \frac{Z_{t-1}^{CC}(k,l)X_t^{CS}(k,l) + Z_{t-1}^{SC}(k,l)X_t^{SS}(k,l)}{Z_{t-1}^{CC}(k,l)^2 + Z_{t-1}^{SC}(k,l)^2} & l = N, k \in \{1, \dots, N-1\} \\ \frac{X_t^{CS}(k,l)}{Z_{t-1}^{CC}(k,l)} & k = 0, l = N \end{cases} \quad (4.14)$$

$$g_{mu,mv}(k,l) = \begin{cases} \hat{g}_{mu,mv}^{SC}(k,l) & k, l \in \{1, \dots, N-1\} \\ \frac{Z_{t-1}^{CC}(k,l)X_t^{SC}(k,l) - Z_{t-1}^{SC}(k,l)X_t^{CC}(k,l)}{Z_{t-1}^{CC}(k,l)^2 + Z_{t-1}^{SC}(k,l)^2} & l = 0, k \in \{1, \dots, N-1\} \\ \frac{Z_{t-1}^{CC}(k,l)X_t^{SC}(k,l) + Z_{t-1}^{CS}(k,l)X_t^{SS}(k,l)}{Z_{t-1}^{CC}(k,l)^2 + Z_{t-1}^{CS}(k,l)^2} & k = N, l \in \{1, \dots, N-1\} \\ \frac{X_t^{SC}(k,l)}{Z_{t-1}^{CC}(k,l)} & k = N, l = 0 \end{cases} \quad (4.15)$$

$$\begin{bmatrix} Z_{t-1}^{CC} & -Z_{t-1}^{CS} & -Z_{t-1}^{SC} & Z_{t-1}^{SS} \\ Z_{t-1}^{CS} & Z_{t-1}^{CC} & -Z_{t-1}^{SS} & -Z_{t-1}^{SC} \\ Z_{t-1}^{SC} & -Z_{t-1}^{SS} & Z_{t-1}^{CC} & -Z_{t-1}^{CS} \\ Z_{t-1}^{SS} & Z_{t-1}^{SC} & Z_{t-1}^{CS} & Z_{t-1}^{CC} \end{bmatrix} \begin{bmatrix} \hat{g}^{CC}(k,l) \\ \hat{g}^{CS}(k,l) \\ \hat{g}^{SC}(k,l) \\ \hat{g}^{SS}(k,l) \end{bmatrix} = \begin{bmatrix} X_t^{CC} \\ X_t^{CS} \\ X_t^{SC} \\ X_t^{SS} \end{bmatrix} \quad (4.16)$$

Eş. 4.14 ve Eş. 4.15'te gösterilen sözde faz fonksiyonları matrisin kenar sütun ve

kolonlarının hesaplanmasında kullanılmaktadır. Buradaki \hat{g} ve \hat{f} fonksiyonları ise Eş. 4.16'da verilen eşitliğe göre hesaplanmaktadır.

Elde edilen \hat{f} ve \hat{g} fonksiyonlarının ters Ayrık Kosinüs Sinüs Dönüşümü (AKSD) ve Ayrık Sinüs Kosinüs Dönüşümü (ASKD) katsayıları Eş. 4.17 ve Eş. 4.18'deki gibi hesaplanarak elde edilen grafiğin tepsi noktasının işaret ve koordinat bilgisi Çizelge 4.1'e göre yorumlanarak öteleme miktarı ve yönü belirlenmektedir. Uygulamalar Matlab üzerinden yapıldığından dolayı, görüntü ve görüntü sinyali üzerinden elde edilen diğer sinyallerin grafikleri orijin noktası sol üst köşe olacak ve x eksen değeri sağa doğru büyümekte, y eksen değeri aşağı doğru büyümekte olduğu kabul edilmektedir.

$$AKSD^{-1} = \frac{4}{N^2} \sum_{k=0}^{N-1} \sum_{l=1}^N C^2(k) C^2(l) f_{mu,mv}(k,l) \cos(k\pi/N(m+0,5)) \sin(l\pi/N(n+0,5)) \quad (4.17)$$

$$ASKD^{-1} = \frac{4}{N^2} \sum_{k=1}^N \sum_{l=0}^{N-1} C^2(k) C^2(l) g_{mu,mv}(k,l) \sin(k\pi/N(m+0,5)) \cos(l\pi/N(n+0,5)) \quad (4.18)$$

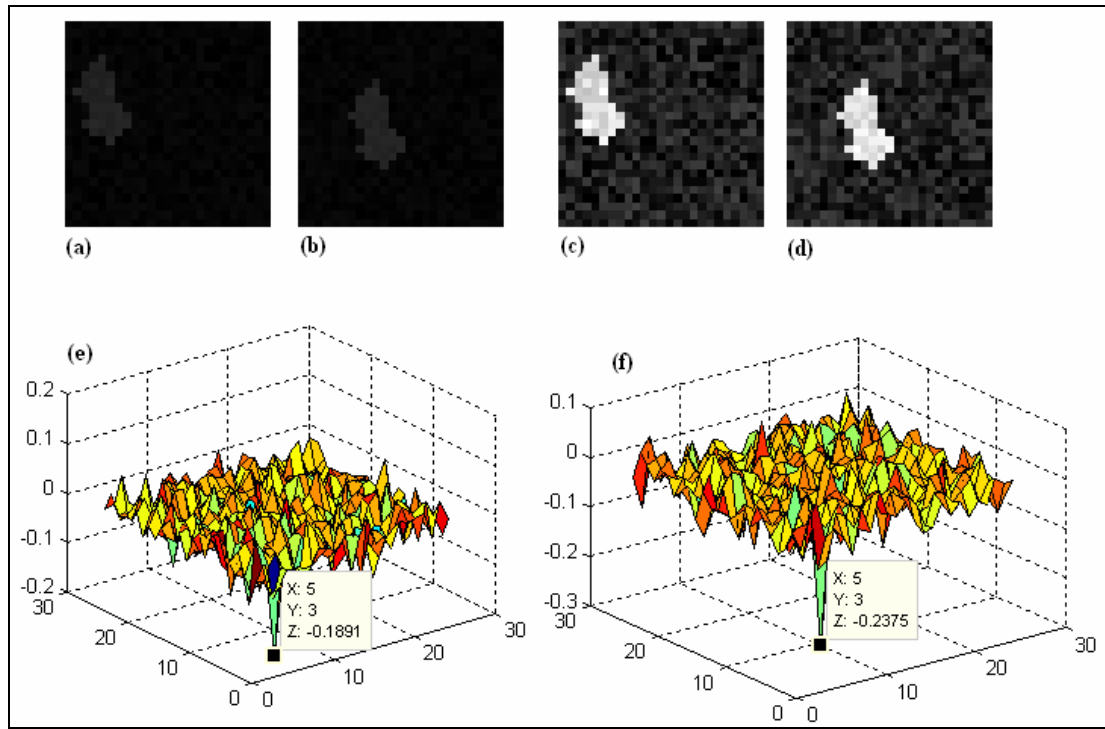
Çizelge 4.1. Sözde faz ile hareket belirleme çizelgesi

ASKD ⁻¹ işareti	AKSD ⁻¹ işareti	Öteleme (x,y)
+	-	(m _u , -(m _v -1))
-	-	(m _u , m _v)
+	+	(-(m _u -1), -(m _v -1))
-	+	(-(m _u -1), m _v)

Anlatılan yöntemin, başarısının doğrulanması amacıyla oluşturulan örnek veriler üzerinde uygulanmıştır. Arka plana ve nesneye gürültü eklenerek Eş. 4.19'da belirtilen hesaplama ile hata oranı hesaplanarak belirli bir hata oranı altında da sözde faz hesaplaması yapılarak hareket tanıma yapılabildiği incelenmiştir.

$$MSE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (x_2(i, j) - x_1(i, j))^2 \quad (4.19)$$

Arka plandaki hata değeri 17,1488, nesne üzerindeki hata değeri 12,9 olarak x2 görüntüsünde x1 görüntüsüne göre, x ekseninde +5 y ekseninde +3 birim öteleme uygulanarak, x1 ve x2 görüntüleri üzerinde sözde faz hesaplaması uygulanarak ötelenme hesaplanmıştır. Şekil 4.11’de sonuçları gösterilmektedir.



Şekil 4.11. (a) x1 görüntüsü (b) x2 görüntüsü (c) x1’in histogram eşitleme yapılmış hali, (d) x2’nin histogram eşitleme yapılmış hali, (e) $AKSD^{-1}(f_{\mu, mv}(k,l))$ grafiği, (f) $ASKD^{-1}(g_{\mu, mv}(k,l))$

4.4.3. Yapılan iyileştirmeler

Sözde faz bilgisi, bir sinyal ve ötelenmiş halinin AKD ve ASD katsayıları üzerinden hesaplanmaktadır. AKD ve ASD katsayıları tek başlarına faz bilgisi taşımamaktadır. Ancak her iki sinyal üzerinden yola çıkılarak faz bilgisine erişilebilmektedir. Alternatif olarak iki farklı yöntem daha incelenmiştir. Birincisi Fourier dönüşümü

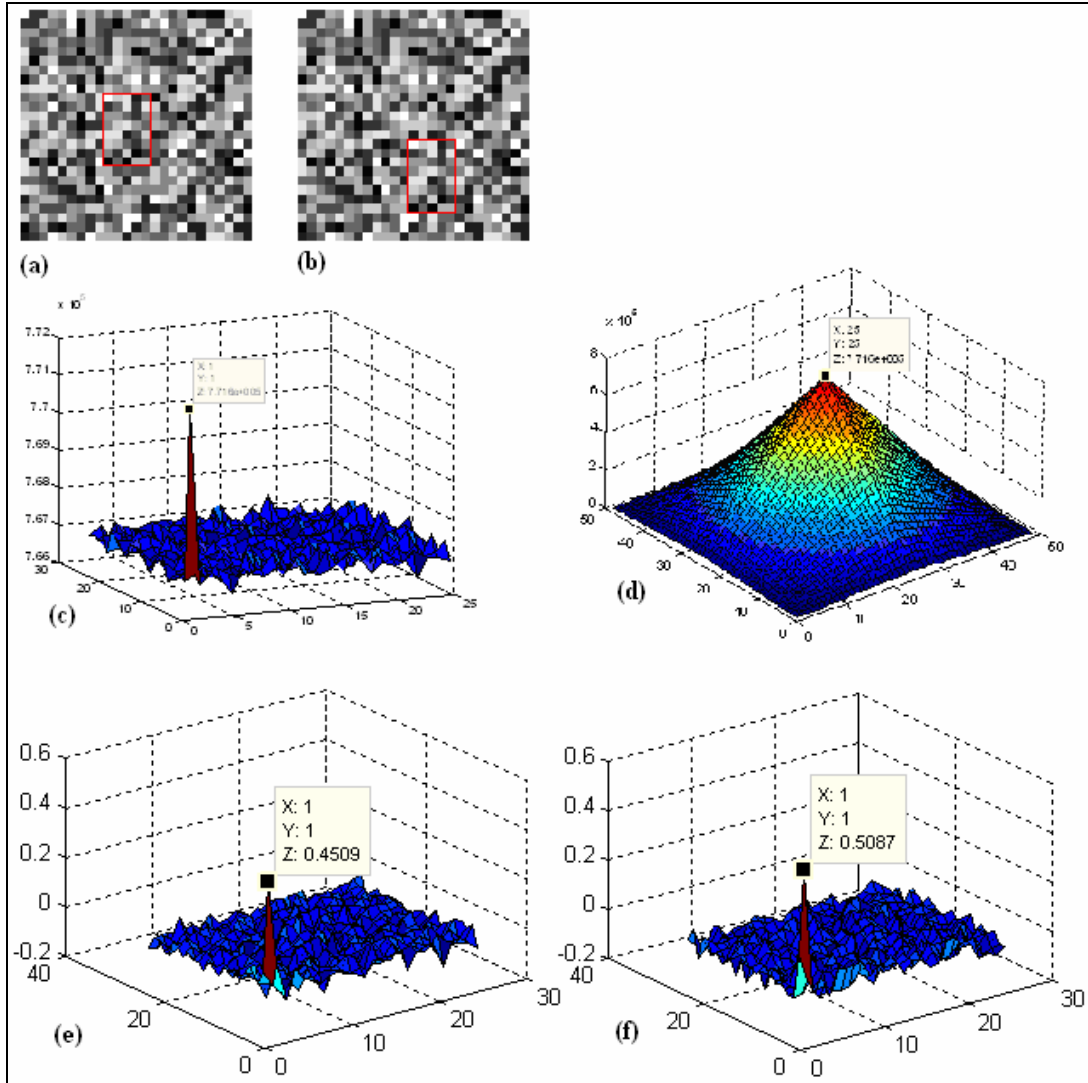
katsayıları üzerinden faz bilgisinin çıkartılmasıdır. Eş. 4.20’de Fourier dönüşüm ve faz arasındaki ilişki görülebilmektedir.

$$F(x(t-T)) = F(x(t))e^{-j\omega T} \quad (4.20)$$

Fourier dönüşüm katsayıları faz bilgisini taşımaktadır ve Eş. 4.21’de gösterildiği şekilde faz hesaplanabilmektedir.

$$F^{-1}(F(t)\text{conj}(F(t-T))) \quad (4.21)$$

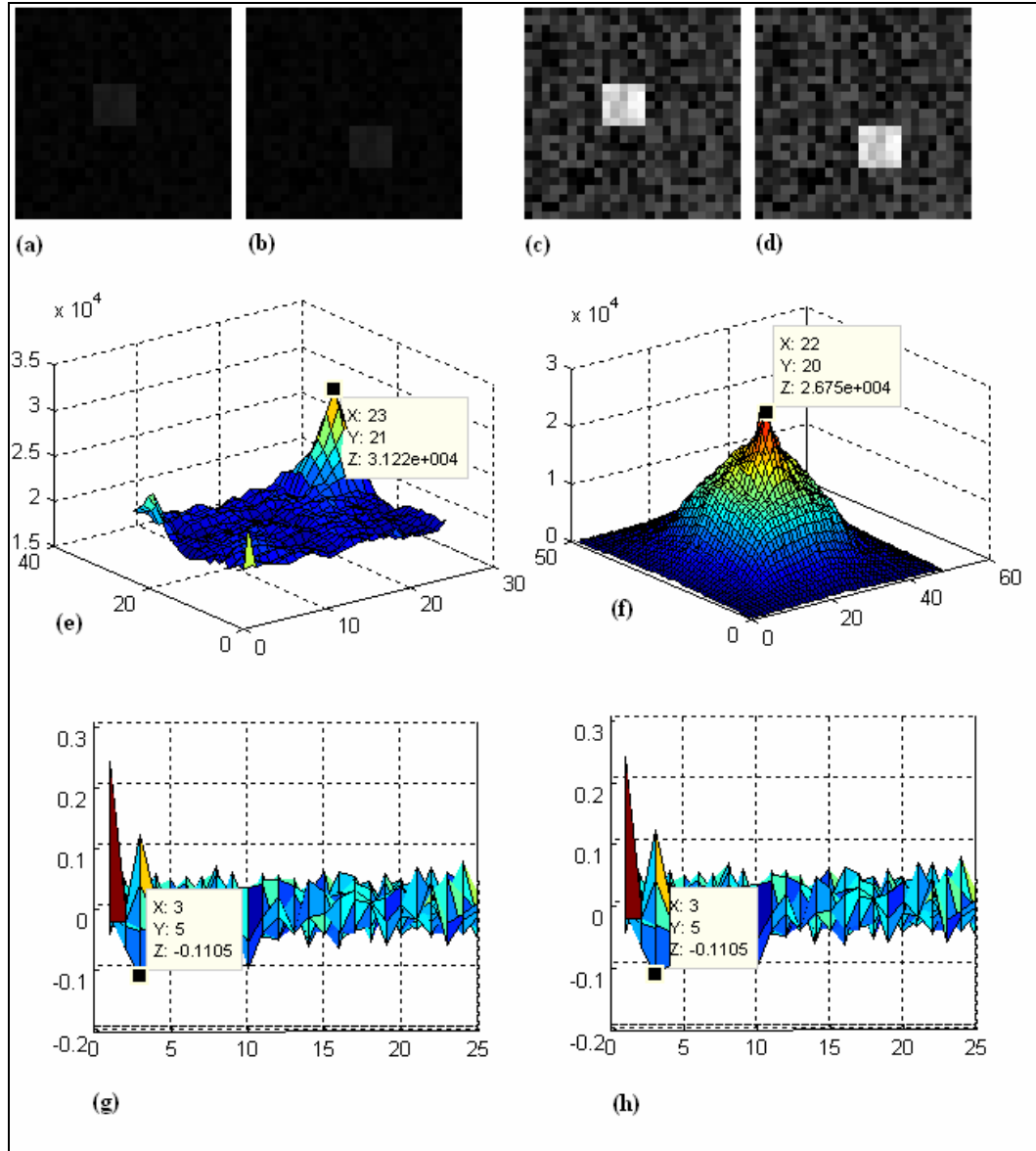
Öteleme hesaplanmasında ikinci yöntem, uzaysal düzlemde uygulanan korelasyondur. Korelasyon, Fourier katsayıları çarpımının işlemsel olarak uzaysal düzlemdeki karşılığıdır.



Şekil 4.12. (a) x1 görüntüsü (b) x2 görüntüsü, x'in ötelenmiş hali (c) Fourier faz hesaplama sonucu (d) korelasyon sonucu (e) sözde faz hesaplama $AKSD^{-1}$ sonucu (f) sözde faz hesaplama $ASKD^{-1}$ sonucu

Şekil 4.12'de, (a) görüntüsünde arka plan ve nesne aynı gri seviyesi arasında rastgele değerlerden oluşturulmuştur. (b) görüntüsünde kırmızı kare ile işaretlenmiş olan nesne aynı arka plan üzerinde ötelenmiştir. Nesnenin arka plan üzerindeki ayırt edilebilirliğinin incelendiği bu görüntüler üzerinde üç farklı faz hesaplama yöntemi uygulanmıştır. (c) grafiği Fourier faz hesaplama yöntemi sonucudur ve öteleme görülememektedir. (d) grafiği korelasyon sonucunu göstermektedir ve bu sonuca göre öteleme bulunamamıştır. (e) ve (f) grafikleri ise sözde faz hesaplama yönteminin $AKSD^{-1}$ ve $ASKD^{-1}$ sonuç grafikleridir. Aynı şekilde faz bilgisi bu

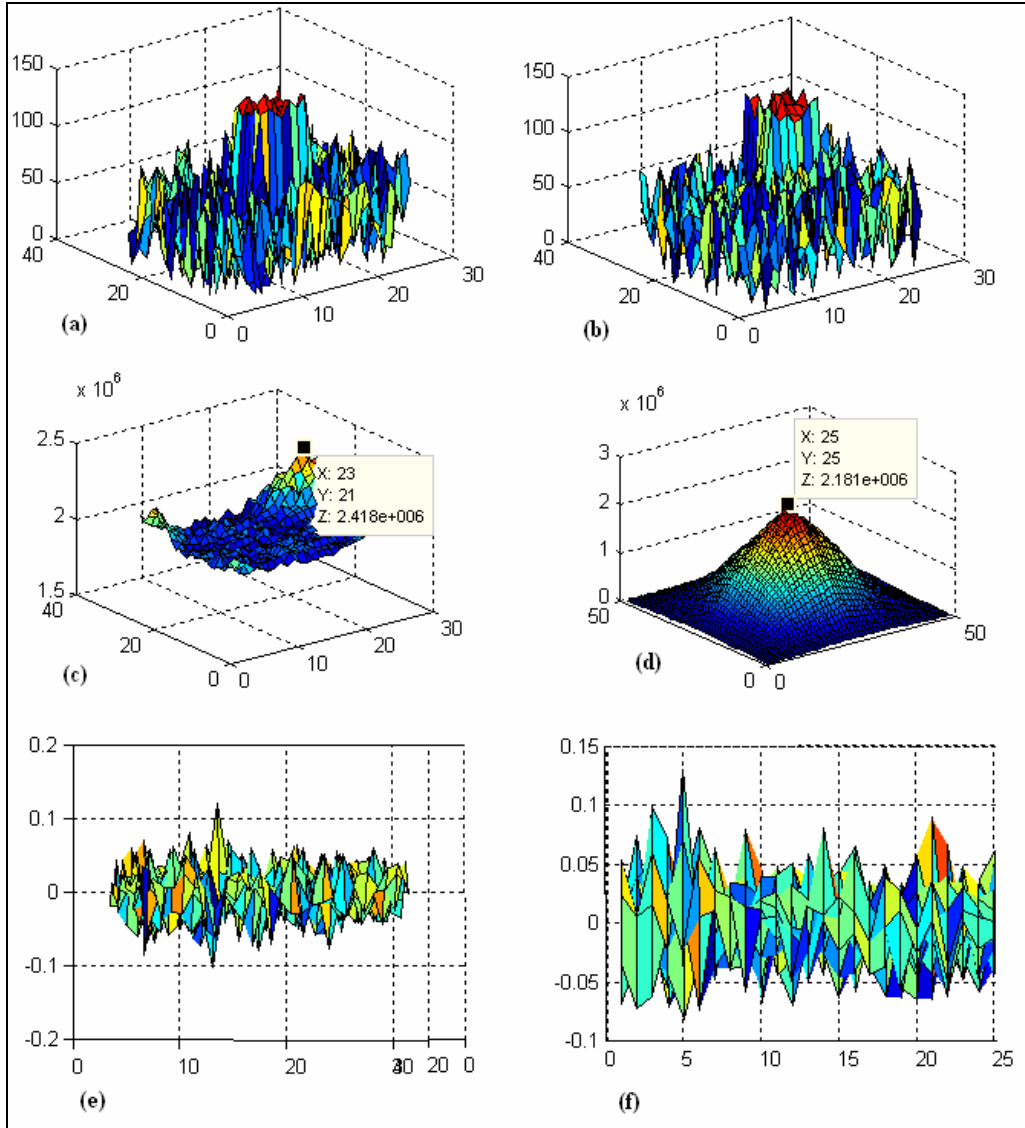
grafiklerden de elde edilememektedir. Sonuç olarak nesne ayırt edilebilir olmadığı durumda her üç yöntemde başarılı olmamaktadır.



Şekil 4.13. (a) x1 görüntüsü, arka plan 0-10 arası, nesne 20-30 arası rastgele değerlerden üretilmiştir (b) x2 görüntüsü, x1 görüntüsündeki nesnenin ötelenmiş hali (c) x1'in histogram eşitleme yapılmış hali (d) x2'nin histogram eşitleme yapılmış hali (e) Fourier faz hesaplama yöntemi sonucu (f) korelasyon sonucu (g) sözde faz hesaplama yöntemi AKSD⁻¹ sonucu (h) sözde faz hesaplama yöntemi ASKD⁻¹ sonucu

Şekil 4.13'de ise arka plana göre ayırt edilebilirliği düşük gri seviyesi değerlerinden

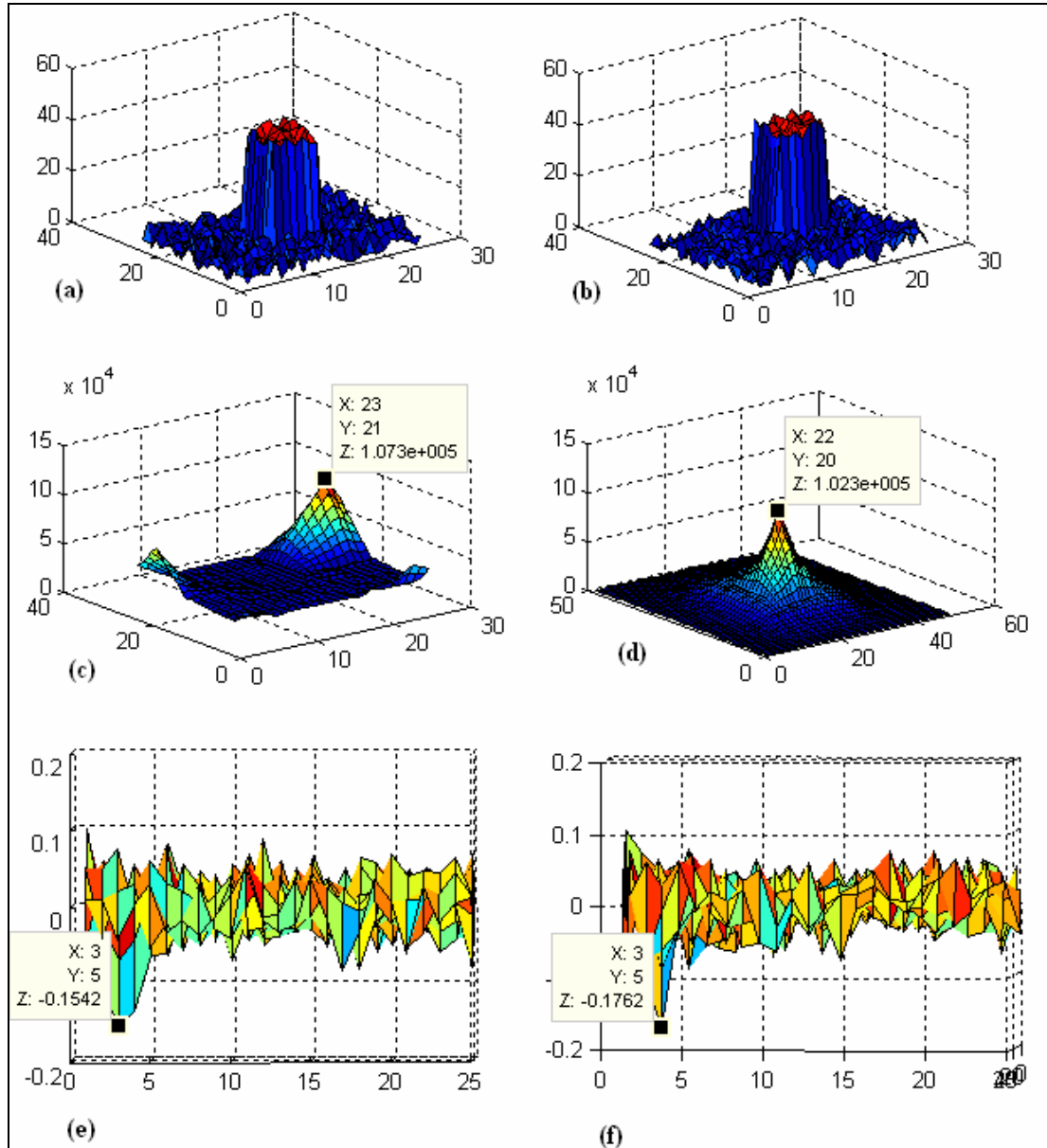
oluşmuş bir nesnenin görüntüsüne, öteleme uygulanarak her üç yöntem ile hareket tanıma yapılarak sonuçları incelenmiştir. (e) ve (f) grafiklerinden Fourier faz hesaplama ile korelasyonun başarıyla hareketi tanıyabildiği görülmektedir. (g) ve (h) grafiklerinde, sözde faz hesaplama yöntemi sonuçları görülmektedir. Ortaya çıkan sonuç grafiklerinin $x=0$ ve $y=0$ doğruları boyunca en yüksek değere sahip tep noktalarının oluştuğu görülmektedir. Ancak sonuç grafiğinin iç bölgesinde oluşan en yüksek tep noktaları doğru sonucu vermektedir. Nesne ile arka plan arasında ayırt edilebilirlik düşük olan görüntüler üzerinde sözde faz yöntemi, sonuç grafiklerinde $x=0$ ve $y=0$ doğruları üzerinde yanıltıcı sonuç verdiği görülmüştür.



Şekil 4.14. (a) x1 görüntü grafiği (b) x2 görüntü grafiği, x1'in üzerine gürültü eklenerek ötelenmiş hal (c) Fourier faz hesaplama yöntemi (d) korelasyon sonucu (e) sözde faz hesaplama $AKSD^{-1}$ sonucu (f) sözde faz hesaplama sonucu $ASKD^{-1}$ sonucu

Şekil 4.14'de üzerinde hareket tanıma yapılacak görüntünün arka planına ve nesneye gürültü eklenmiş ve nesne ötelenmiştir. Sonuçta ortaya çıkan x2 görüntüsü ile x1 görüntüsünün arka planları arasında MSE ile 1610,1 hata hesaplanmıştır. Nesne üzerinde ise MSE ile 132,1837 hata hesaplanmıştır. (a) görüntüsü x1'in, (b) görüntüsü ise x2'nin grafikleridir. (c) grafiği Fourier faz hesaplama yöntemi sonucudur ve bu grafiğe göre başarıyla öteleme miktarı bulunmaktadır. (d) grafiği korelasyon sonucunu göstermektedir ancak bu yöntem doğru sonucu

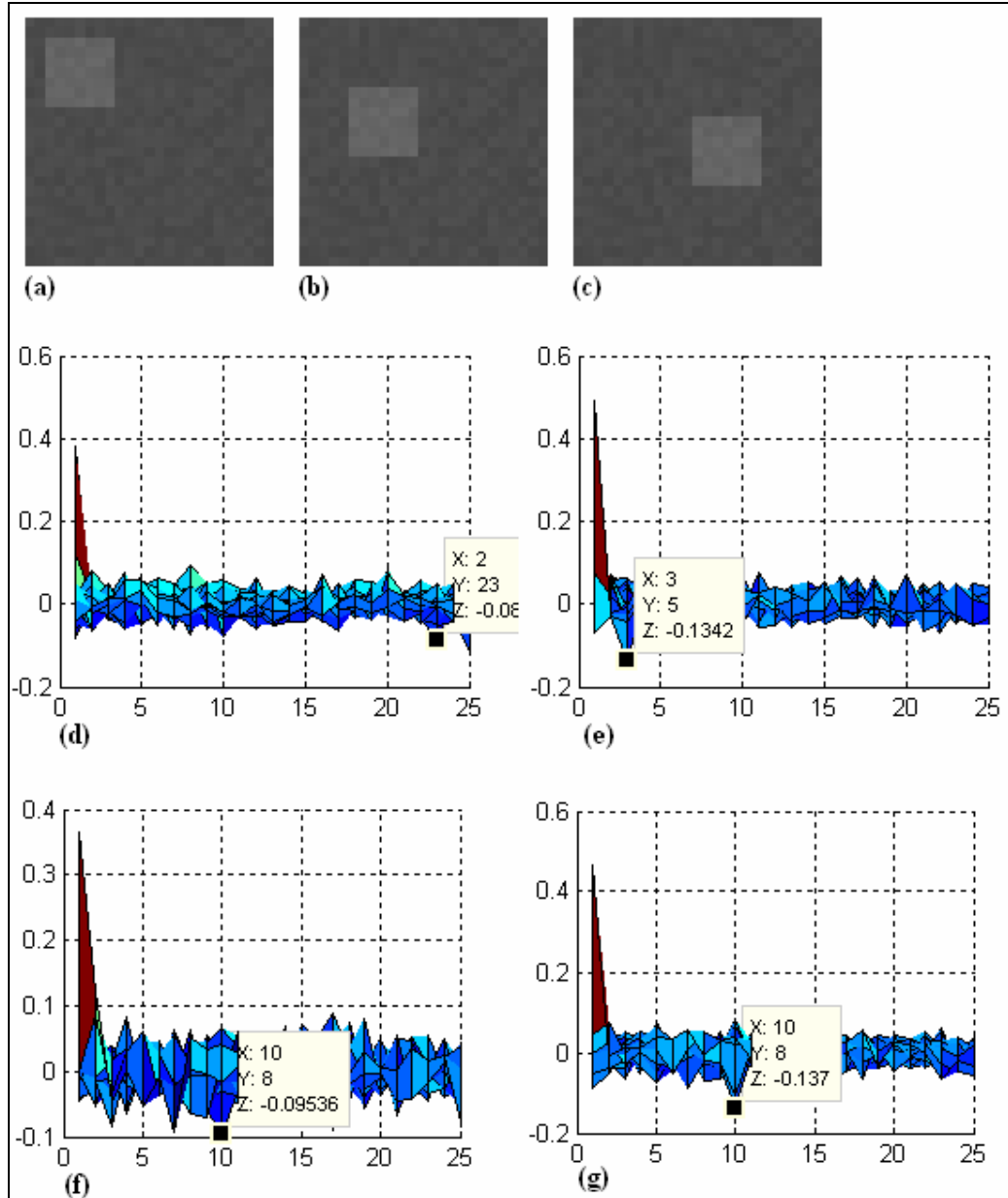
bulamamaktadır. (e) ve (f) grafikleri ise sözde faz hesaplama yöntemi sonuçlarıdır ve görüntü üzerindeki yüksek gürültü, sonuçları üzerinde de gürültü oluşturarak doğru sonucun bulunmasını engellemektedir.



Şekil 4.15. (a) x1 görüntü grafiği (b) x2 görüntü grafiği, x1'in üzerine gürültü eklenerek ötelenmiş hali (c) Fourier faz hesaplama yöntemi (d) korelasyon sonucu (e) sözde faz hesaplama AKSD⁻¹ sonucu (f) sözde faz hesaplama sonucu ASKD⁻¹ sonucu

Şekil 4.15'de x1 görüntüsünün arka planına MSE hesaplama ile 17,4880, nesneye ise

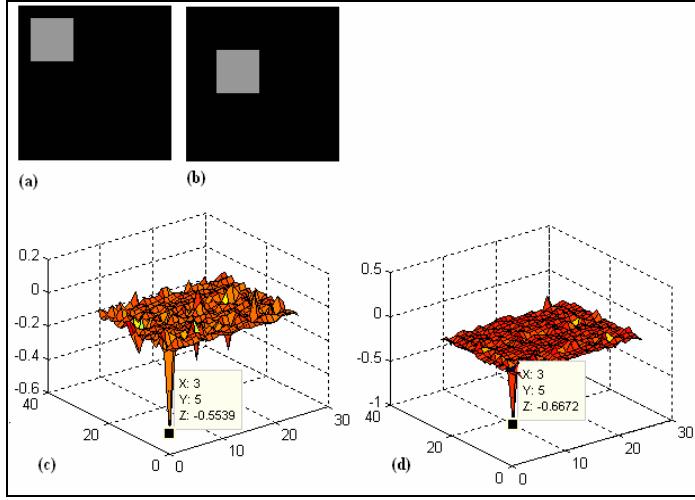
6,2245 hata eklenmiş ve öteleme uygulanarak x2 görüntüsü elde edilmiştir. Her üç hareket tanıma yöntemi uygulandığında, üçünün de başarılı sonuç verdiği görülmüştür. Sonuç olarak sözde faz hesaplama yönteminin başarısının iki görüntü arasındaki hatadan etkilendiği görülmüştür.



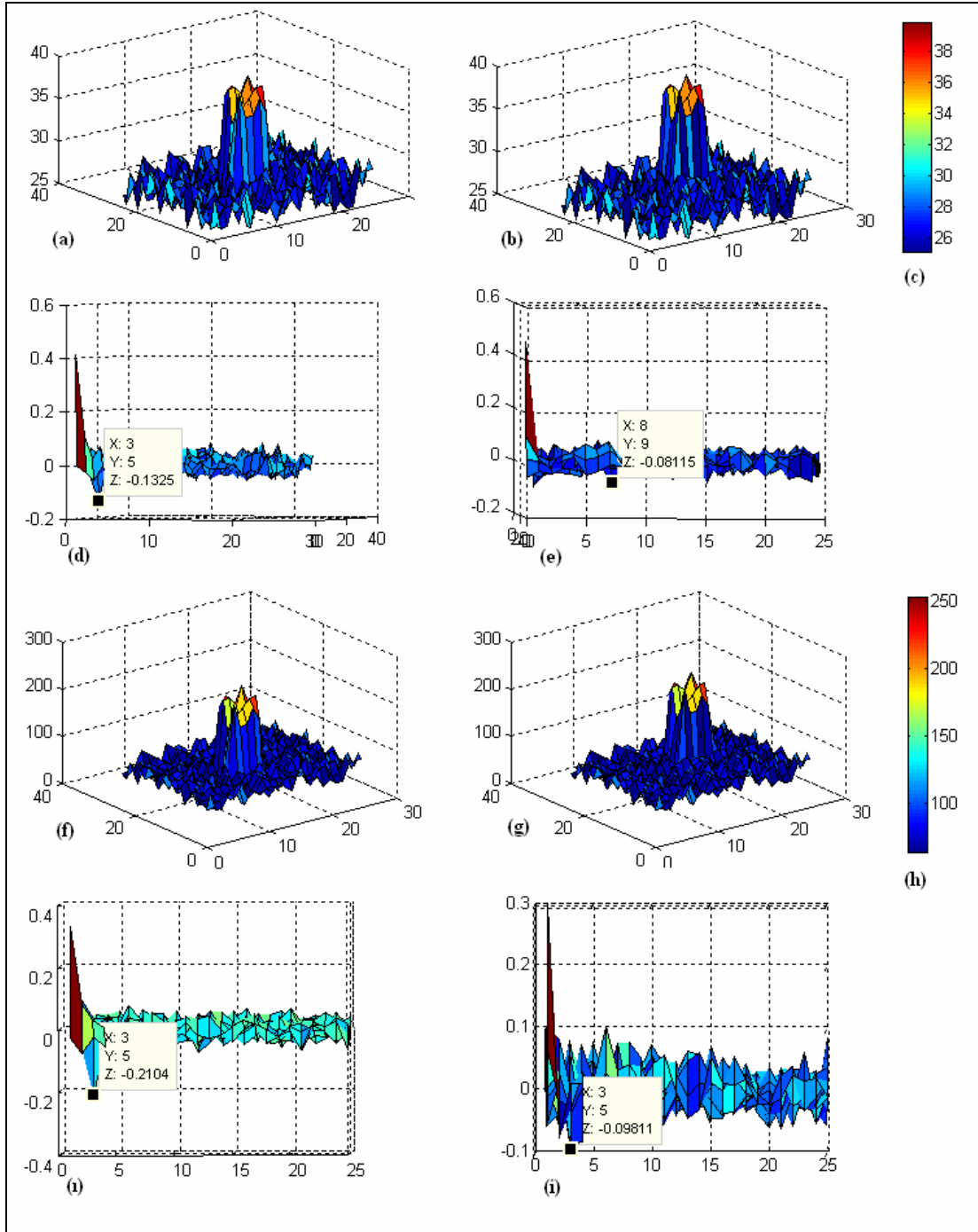
Şekil 4.16. (a) x1 görüntüsü (b) x2 görüntüsü, x1'in +3,+5 ötelenmiş hali (c) x3 görüntüsü, x1'in +10,+8 ötelenmiş hali (d,e) x1 ve x2 üzerinde sözde faz hesaplama yöntemi AKSD⁻¹ ve ASKD⁻¹ sonuçları (f,g) x1 ve x3 üzerinde sözde faz hesaplama yöntemi AKSD⁻¹ ve ASKD⁻¹ sonuçları

Şekil 4.16 (a)'da gösterilen x_1 görüntüsünün nesne ile arka planı arasındaki ayırt edilebilirlik düşüktür. (b)'deki x_2 görüntüsü, x_1 'in +3,+5 ötelenmiş halidir. Bu iki görüntü üzerinde sözde faz hesaplaması yapıldığında (d) ve (e) grafiklerinde görülen sonuçlar elde edilmiştir. Buna göre başarılı sonuç elde edilemediği görülmektedir. (c)'deki x_3 görüntüsü ise x_1 'in +10,+8 ötelenmiş halidir. x_1 ve x_3 üzerinde sözde faz hesaplaması yapıldığında (f) ve (g) grafiklerindeki sonuçlar elde edilmiştir ve başarılı olduğu görülmüştür. Buna göre, sözde faz hesaplamasının başarısı görüntü üzerindeki öteleme miktarına bağlı olduğu görülmektedir. Sözde faz hesaplaması görüntünün tamamına ait AKD ve ASD katsayıları ile yapılmaktadır. Bu yüzden gürültülü görüntüler ve nesnenin ayırt edilebilirliğinin düşük olduğu görüntülerde, sonuç üzerinde ortaya çıkan gürültü küçük değişimlerin hesaplanabilmesine engel olmaktadır. Bu problemi çözmek için bu çalışmada eşikleme yönteminin kullanılması önerilmiştir.

Şekil 6.16'daki x_1 ve x_2 görüntülerine Otsu'nun eşikleme yöntemi ile eşikleme uygulanarak nesne ve arka plan arasındaki ayırt edilebilirlik artırılmış ve Şekil 4.17'deki (a) ve (b) görüntüleri elde edilmiştir. Bu görüntüler üzerinde sözde faz hesaplaması yapıldığında, Şekil 4.17. (c) ve (d) grafiklerinde görülen $AKSD^{-1}$ ve $ASKD^{-1}$ sonuçları elde edilmiştir. Böylelikle nesnenin ayırt edilebilirliği düşük olan görüntülerde eşikleme uygulanarak sözde faz hesaplamasının başarısı artırılmıştır.

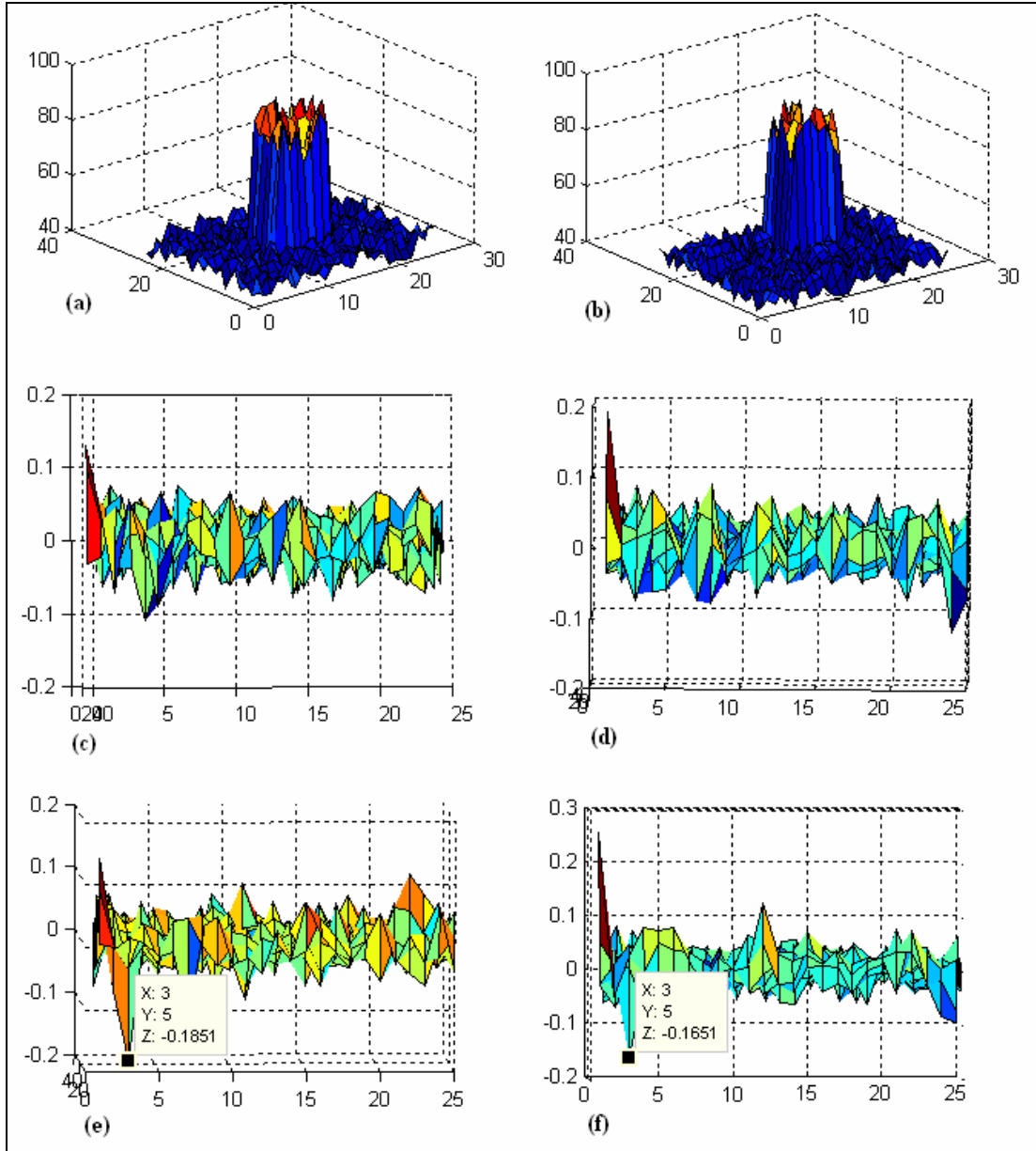


Şekil 4.17. (a) x1 görüntüsünün eşiklenmiş hali (b) x2 görüntüsünün eşiklenmiş hali (c,d) a ve b görüntülerinin sözde faz hesaplaması sonucu ortaya çıkan $AKSD^{-1}$ ve $ASKD^{-1}$ grafikleri



Şekil 4.18. (a) x_1 görüntüsü (b) x_2 görüntüsü, x_1 'in ötelenmiş hali (c) x_1 ve x_2 'nin gri seviyesi aralığı (d,e) x_1 ve x_2 üzerinde sözde faz hesaplaması $AKSD^{-1}$ ve $ASKD^{-1}$ sonuçları (f) y_1 , x_1 'in üssel fonksiyon ile gri seviyesi dönüşümü uygulanması sonucu (g) y_2 , x_2 'in üssel fonksiyon ile gri seviyesi dönüşümü uygulanması sonucu (h) y_1 ve y_2 'nin gri seviyesi aralığı (ı,i) y_1 ve y_2 üzerinde sözde faz hesaplaması $AKSD^{-1}$ ve $ASKD^{-1}$ sonuçları

Şekil 4.18'de (a) ve (b) görüntülerinde x_1 ve x_2 'nin grafiklerinde nesnenin ayırt edilebilirliğinin arka plana göre düşük olduğu görülmektedir. (c) şeklinde x_1 ve x_2 için gri seviyesi aralığı görülmektedir. Buna göre tüm görüntü 25-40 arasında değerlerden oluşmaktadır. x_1 ve x_2 üzerinde sözde faz hesaplaması yapıldığında (d) ve (e) grafiklerindeki sonuçlar elde edilmektedir. $AKDS^{-1}$ sonucu başarılı olmasına rağmen $ASKD^{-1}$ sonucu başarılı olamadığından dolayı kesin bir sonuca varılamaz. x_1 ve x_2 görüntülerine üssel fonksiyon kullanılarak gri seviyesi dönüşümü uygulandığında, (f) ve (g) şekillerinde görülen y_1 ve y_2 görüntüleri elde edilmiştir. (h) şeklinde ise y_1 ve y_2 'nin gri seviyesi aralığı verilmektedir. Buna göre y_1 ve y_2 görüntülerinin gri seviyeleri 50-250 arasında değerlerden oluşmaktadır. Böylece nesne ve arka plan arasındaki ayırt edilebilirlik artırılmıştır. y_1 ve y_2 görüntüleri üzerinde sözde faz hesaplaması yapılarak (ı) ve (i) sonuçları elde edilmiştir. Buna göre $AKSD^{-1}$ ve $ASKD^{-1}$ sonuçları $x=0$ ve $y=0$ doğruları göz ardı edildiğinde doğru sonucu vermektedir. Sonuç olarak nesnenin ayırt edilebilirliğinin düşük olduğu görüntüler üzerinde üssel fonksiyon ile gri seviyesi dönüşümü uygulanarak sözde faz hesaplamasının başarımı artırılabilir.



Şekil 4.19. (a) x1 görüntüsü, (b) x2 görüntüsü, x1'in gürültü eklenmiş ve ötelenmiş hali (c), (d) x1 ve x2'ye sözde faz hesaplaması uygulamasının $AKSD^{-1}$ ve $ASKD^{-1}$ sonucu (e), (f) x1 ve x2'ye medyan filtre uygulandıktan sonra sözde faz hesaplamasının $AKSD^{-1}$ ve $ASKD^{-1}$ sonucu

Şekil 4.19'da (b) x2 görüntüsü, (a) x1 görüntüsünde arka plana MSE hesaplamasına göre 16,3392, nesneye ise 85,5102 hata eklenmiş ve x ekseninde 3 birim y ekseninde 5 birim ötelenmiş halidir. Sözde faz hesaplaması doğrudan x1 ve x2 görüntüsü üzerine uygulandığında, sonuç grafikleri (c) ve (d) şekillerinde görüldüğü üzere başarılı olamamaktadır. x1 ve x2 üzerinde 3x3 lük pencere kullanılarak medyan filtre

uygulanmak suretiyle gürültü temizlenmiş ve elde edilen görüntüler üzerinde sözde faz hesaplaması yapıldığında (e) ve (f) şekillerinde görülen sonuç grafikleri elde edilmiştir. Bu sonuçlara göre gürültü temizlendikten sonra sözde faz hesaplaması başarılı sonuç vermektedir.

5. UYGULANAN YÖNTEM

- Matlab ile, 3B AKD sıkıştırma uygulaması hazırlanmıştır. Bu uygulamada;
- ‘avi’ biçimindeki video dosyasından ardışık görüntüler alınarak 3 boyutlu matris içerisine kopyalanmaktadır
- video matrisi, parametrik sayıdaki artarda N sayıda karelere bölünmektedir
- her bir küçük matris için 3B AKD uygulanmaktadır
- 3B AKD katsayıları parametrik budama değerine göre budanmaktadır
- budanmış AKD katsayıları tek boyutlu vektör haline getirilmektedir
- vektör Run-length ile sıkıştırılmaktadır
- Run-length ile sıkıştırılmış vektör Huffman kodlama ile kodlanmaktadır
- sıkıştırılmış vektörler artarda eklenerek sıkıştırılmış video görüntüsü oluşturulmaktadır
- sıkıştırılmış video görüntüsü vektörü açılmaktadır
- vektör tekrar 3 boyutlu matrisler haline dönüştürülmektedir
- 3 boyutlu matrislere AKD^{-1} uygulanmaktadır
- 3 boyutlu matrisler artarda eklenerek video görüntü matrisi oluşturulmaktadır
- video görüntü matrisinden yeni bir avi dosyası oluşturulmaktadır
- sonuç olarak sıkıştırma oranı ve sıkıştırılmış görüntünün, orijinal görüntüye göre Eş. 7.1’de ki eşitliğe göre MSE değeri hesaplanmaktadır

$$MSE = \frac{\sum_x \sum_y \sum_z \left[\frac{(f(x, y, z) - f'(x, y, z))}{255} \right]^2}{N_R N_C N_F} \quad (5.1)$$

Bu bölümdeki sıkıştırma hesaplamaları, sıkıştırma işlemi sonucu ortaya çıkan vektör uzunluğunun, video görüntüsünün sahip olduğu toplam piksel sayısına (genişlik x yükseklik x zaman) olan oranısıyla hesap edilmiştir. Huffman kodlama sonucunda, kodlama tablosunu da içeren ve her elemanı 8 bit uzunluğa sahip bir tek vektör elde edildiğinden dolayı toplam vektör uzunluğu alınmaktadır. Run-length sonucunda ise, AKD katsayılarını içeren vektör oluşmaktadır ve AKD katsayıları 8 bitten daha çok

sayıda bitle ifade edilebilmektedir. Bundan dolayı vektördeki en yüksek sayının ifade edildiği bit sayısı hesaplanarak, o sayının içerisinde ifade edilebileceği en küçük byte sayısı bulunmakta ve toplam vektör uzunluğu bu sayıyı da içereceğinden, hesaplanan byte sayısıyla çarpılmaktadır. Böylece sıkıştırma sonucunda video görüntüsünün gerektirdiği bellek miktarından yüzde olarak ne kadar kazanç sağlandığı hesaplanmıştır. Geri açılan sıkıştırılmış vektörlerden elde edilen görüntü ile orijinal görüntü arasında oluşan hata MSE ile hesaplanmaktadır.

Çizelge 5.1, Çizelge 5.2 ve Çizelge 5.3'te, farklı çerçeve sayıları için, farklı budama değerleri ile aynı video klipinin 70'er kareden oluşan 2 farklı bölümü için sonuçlar görülmektedir.

Çizelge 5.1. Çerçeve sayısı 3 alındığında uygulama sonuçları

Çerçeve sayısı = 3		Video görüntüsü 1, hızlı hareketler ve sık görüntü değişimleri içermektedir			Video görüntüsü 2, yavaş hareketler ve az görüntü değişimi içermektedir		
Sıra	Bud. değeri	RLE sık. oranı %	Huffman sık. oranı %	MSE	RLE Sık. oranı %	Huffman sık. oranı %	MSE
1	10	99,2991	99,7212	0,0156	99,2991	99,7479	0,0094
2	30	95,4451	98,2261	0,0055	95,4451	98,4472	0,0042
3	50	88,3132	95,7291	0,0028	88,3132	96,2555	0,0024
4	70	77,9033	92,4125	0,0015	77,9033	93,3228	0,0013
5	100	56,1423	86,1975	6,7331e-004	56,1423	87,8609	5,2208e-004
6	140	15,6546	76,1801	2,5595e-004	15,6546	79,1212	1,9736e-004

Çizelge 5.2. Çerçeve sayısı 5 alındığında uygulama sonuçları

Çerçeve sayısı = 5		Video görüntüsü 1, hızlı hareketler ve sık görüntü değişimleri içermektedir			Video görüntüsü 2, yavaş hareketler ve az görüntü değişimi içermektedir		
Sıra	Bud. değeri	RLE sık. oranı %	Huffman sık. oranı %	MSE	RLE Sık. oranı %	Huffman sık. oranı %	MSE
1	10	99,4018	99,7629	0,0159	99,4018	99,7815	0,0095
2	30	95,7141	98,3319	0,0056	95,7141	98,5407	0,0042
3	50	88,7484	95,8838	0,0028	88,7484	96,4070	0,0024
4	70	78,5047	92,6104	0,0015	78,5047	93,5196	0,0013
5	100	56,9930	86,4516	6,8433e-004	56,9930	88,1401	5,2643e-004
6	140	16,8378	76,4935	2,5987e-004	16,8378	79,4719	1,9892e-004

Çizelge 5.3. Çerçeve sayısı 12 alındığında uygulama sonuçları

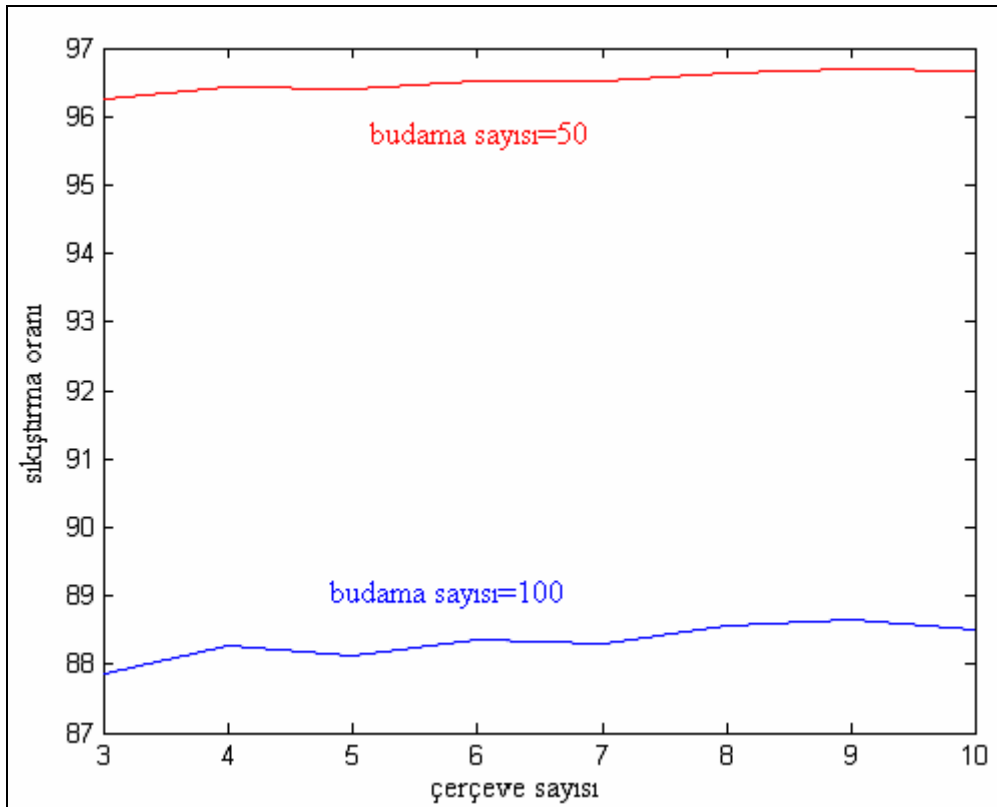
Çerçeve sayısı = 12		Video görüntüsü 1, hızlı hareketler ve sık görüntü değişimleri içermektedir			Video görüntüsü 2, yavaş hareketler ve az görüntü değişimi içermektedir		
Sıra	Bud. değeri	RLE sık. oranı %	Huffman sık. oranı %	MSE	RLE Sık. oranı %	Huffman sık. oranı %	MSE
1	10	99,6605	99,8624	0,0165	99,6605	99,6605	0,0100
2	30	96,5230	98,6410	0,0060	96,5230	98,7259	0,0044
3	50	90,1075	96,3637	0,0030	90,1075	96,6448	0,0025
4	70	80,4141	93,2342	0,0016	80,4141	93,7593	0,0014
5	100	59,7278	87,2731	7,1881e-004	59,7278	88,3101	5,4149e-004
6	140	20,6731	20,6731	2,7269e-004	20,6731	79,5314	2,0426e-004

Çizelge 5.1, Çizelge 5.2 ve Çizelge 5.3'e göre, budama değerinin büyümesiyle, görüntü içerisindeki değeri 0 yapılan AKD katsayılarının sayısı azalmakta, buna bağlı olarak daha az veri sıkıştırılabilmekte ve sıkıştırma oranı düşmektedir. Daha az sayıda AKD katsayısının budanması, görüntüye ait daha az verinin silinmesi anlamına geldiğinden, hata oranı düşmektedir.

Çerçeve sayısı değerinin yükselmesiyle, 3B AKD ve budama uygulanan matris boyutu büyümektedir. Aynı budama değeriyle, daha büyük matrisin budanması, daha fazla sayıda AKD katsayısının 0 yapılması anlamına gelmektedir. Dolayısıyla çerçeve sayısı büyüdükçe sıkıştırma oranı yükselmekte ve video görüntüsünün bir bölümünü gösteren veriye ait daha fazla bilginin silinmesine bağlı olarak sıkıştırmanın hata oranı da yükselmektedir.

Aynı çerçeve sayısında, iki video görüntüsü arasındaki ilişkiye bakıldığı zaman, aynı budama değerlerinde her iki video görüntüsü için de Run-length sıkıştırma oranı hep aynı olmaktadır. Çünkü işlem yapılan matris boyutları aynı ve budanan AKD katsayıları aynıdır. Ve AKD'nin karakteristiği olarak, budama yapılmayan kısımda ardarda aynı katsayılar bulunmadığı için, bu bölümde sıkıştırılabilir veri mevcut değildir. Dolayısıyla sıkıştırılan veri sayısı ve Run-length sıkıştırma oranı her iki video görüntüsü için aynıdır. Huffman kodlama Run-length sıkıştırma sonucu ortaya çıkan vektör üzerinde uygulanmaktadır. Run-length sonucu ortaya çıkan vektör

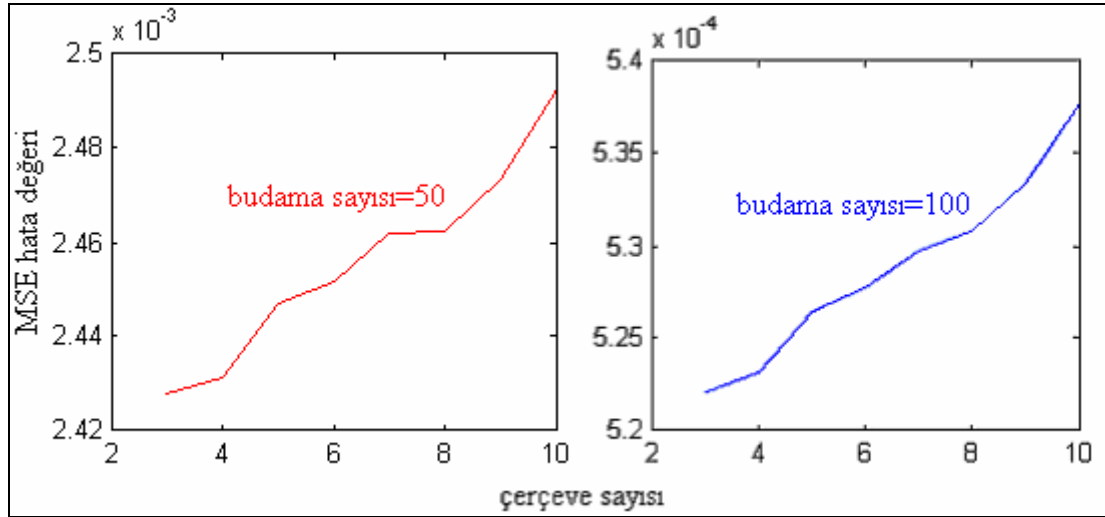
içerisinde AKD katsayıları bulunmaktadır. İkinci video görüntüsünde, birinci videoya göre daha yavaş değişimler bulunmasından dolayı, AKD katsayıları daha küçük değerlerle ifade edilmekte ve birbirlerine daha çok benzemektedirler. Bundan dolayı Huffman kodlama sonucu daha küçük sayıdaki birtlerle ifade edilebilen değerler ortaya çıkmaktadır ve Huffman sıkıştırma oranı ikinci video görüntüsünde daha yüksektir. İki video görüntüsü için hata değerlerinin ise farklı olduğu görülmektedir. Video 1 görüntüsü daha hızlı hareketler ve hızlı değişimler içermektedir. Buna bağlı olarak video görüntüsünün enerjisi, matris küp üzerinde daha geniş bir alana yayılmaktadır ve budamadan dolayı, video 1 görüntüsünün, video 2 görüntüsüne göre daha çok anlamlı katsayıları silinmektedir. Bu da iki video görüntüsünün hata değerlerinin farklı olmasına neden olmaktadır.



Şekil 5.1. Çerçeve sayısı ile sıkıştırma oranı arasındaki ilişkiyi gösterir grafik

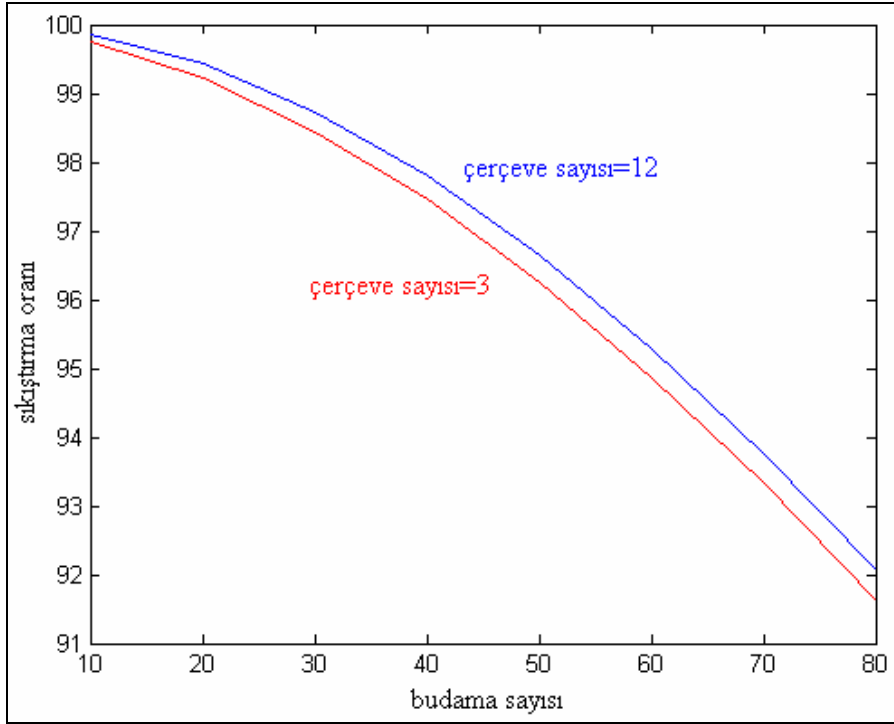
Şekil 5.1'de, çerçeve sayısının değişimine bağlı olarak sıkıştırma oranının nasıl etkilendiği görülmektedir. İki farklı budama değeri için, çerçeve sayısının artmasıyla

blok içerisindeki toplam katsayı artmaktadır. Ancak budama sonucunda görüntüyü ifade edecek katsayı sayısı aynı kaldığından, daha fazla görüntü, aynı sayıda katsayıyla ifade edilmesinden dolayı sıkıştırma artmaktadır.



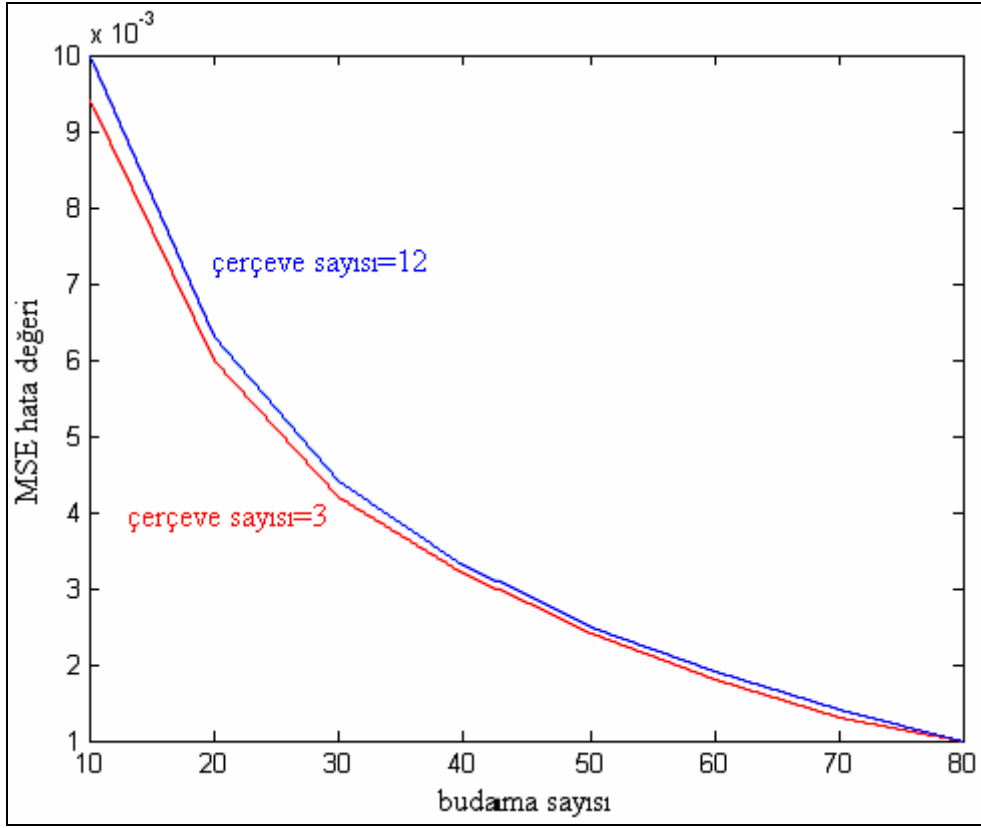
Şekil 5.2. Çerçeve sayısı ile MSE hata değeri arasındaki ilişkiyi gösterir grafik

Şekil 5.2'de iki farklı sabit budama sayısına göre, çerçeve sayısı ile MSE hata değeri arasındaki ilişki gösterilmektedir. Çerçeve sayısı arttıkça, Şekil 5.1'de gösterildiği gibi sıkıştırma oranı artmaktadır. Bu da görüntünün kalitesinin bozulması yani hatanın artması anlamına gelmektedir.



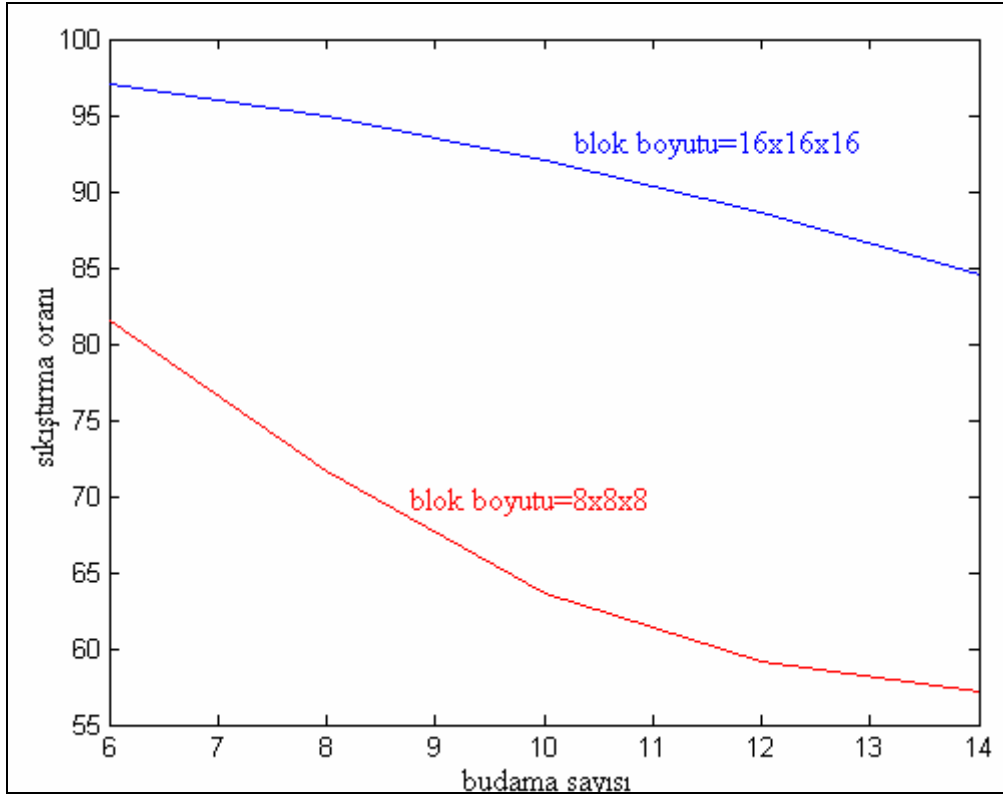
Şekil 5.3. Budama sayısı ile sıkıştırma oranı arasındaki ilişkiyi gösterir grafik

Şekil 5.3'te sabit bir çerçeve sayısı değerinde, budama sayısı artırıldıkça sıkıştırma oranının düştüğü görülmektedir. Budama sayısının artması ile AKD katsayıları arasından daha fazla katsayı sıkıştırılarak görüntü dosyasını oluşturmaktadır. Bu da saklanan veri miktarını artırmaktadır, dolayısıyla sıkıştırma oranı düşmektedir. Çerçeve sayısı değeri 3 iken, aynı budama değerinde, çerçeve sayısı 12 alındığındaki duruma göre daha az sıkıştırma olduğu görülmektedir. Çerçeve sayısı yükseltilerek, AKD katsayısı artırılmakta, ancak aynı budama değerinde aynı sayıda katsayı saklanmaktadır. Bu sebeple sıkıştırma artmaktadır.



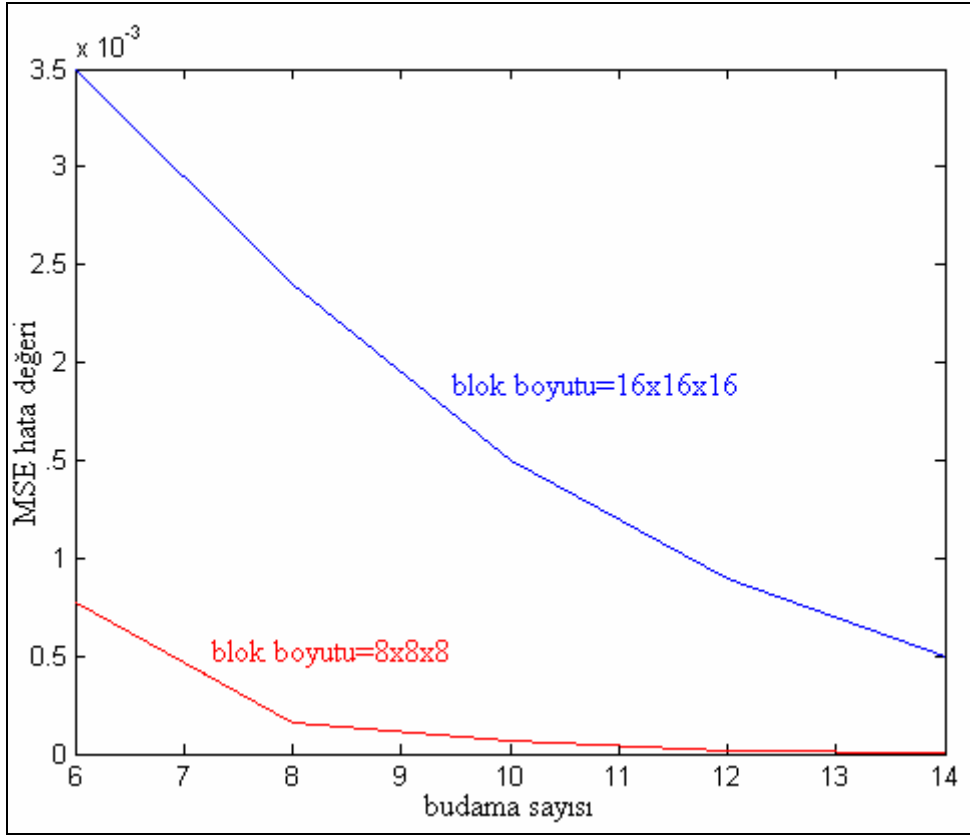
Şekil 5.4. Budama sayısı ile MSE hata değeri arasındaki ilişkiyi gösterir grafik

Şekil 5.4'te sabit çerçeve sayısında, budama değeri yükseldikçe MSE hata değerinin düştüğü görülmektedir. Şekil 5.3'e göre budama sayısı arttıkça sıkıştırma oranının düştüğü görülmektedir. Daha az sıkıştırma görüntü üzerinde daha az hata oluşumu anlamına gelmektedir. Aynı şekilde çerçeve sayısının artmasına bağlı olarak sıkıştırma oranının artmasından dolayı, yüksek çerçeve sayısında daha yüksek hata görülmektedir.



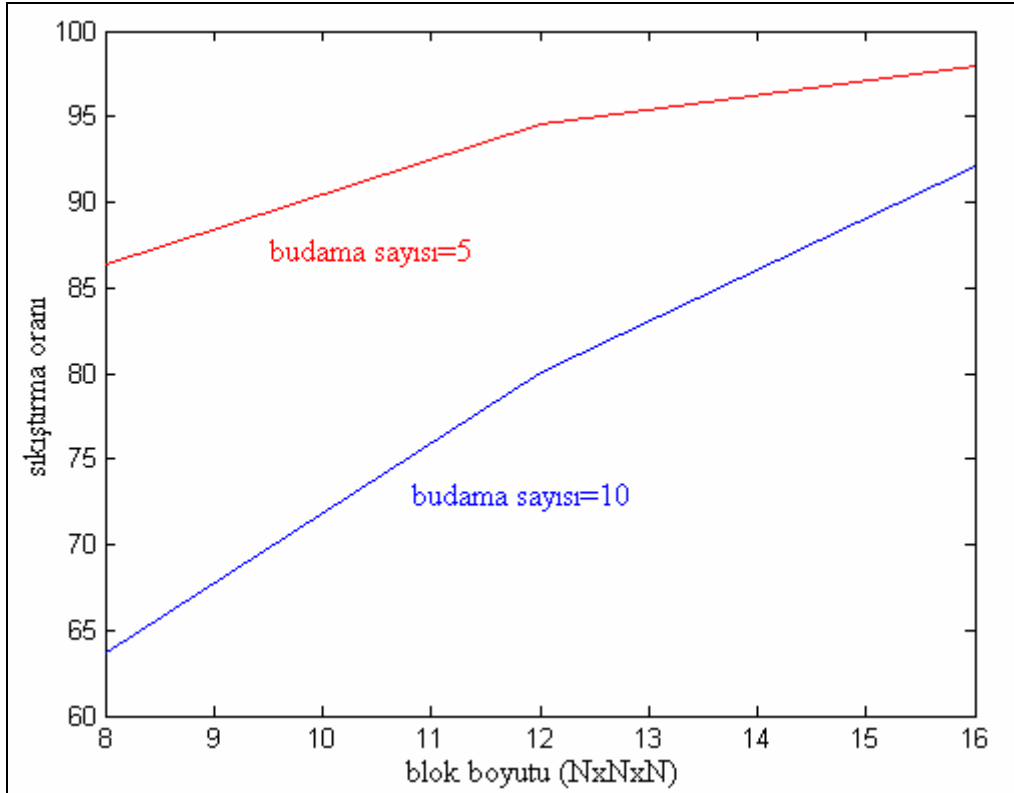
Şekil 5.5. Sıkıştırma, küçük küpler üzerinde yapıldığında, sıkıştırma oranı ile budama sayısı arasındaki ilişkiyi gösterir grafik

Sıkıştırma işlemi video görüntüsünde artarda gelen çerçevelerin tümü üzerinde yapılmak yerine, video matrisi $N \times N \times N$ 'lik küplere bölünerek sıkıştırma uygulanmıştır. Şekil 5.5'te sıkıştırma $8 \times 8 \times 8$ 'lik ve $16 \times 16 \times 16$ 'lık küpler üzerinde uygulandığında, sıkıştırma oranı ve budama sayısı arasındaki ilişki gösterilmiştir. Budama sayısı arttıkça, daha fazla AKD katsayısı saklanmaktadır, bu da sıkıştırma oranını azaltmaktadır. Aynı budama değerinde, $8 \times 8 \times 8$ 'lik bloklarda, $16 \times 16 \times 16$ 'lık bloklara oranla daha fazla AKD katsayısı saklanmaktadır. Bu yüzden blok sayısı azaldıkça, görüntü daha fazla AKD katsayısıyla ifade edilmekte ve sıkıştırma oranı azalmaktadır.



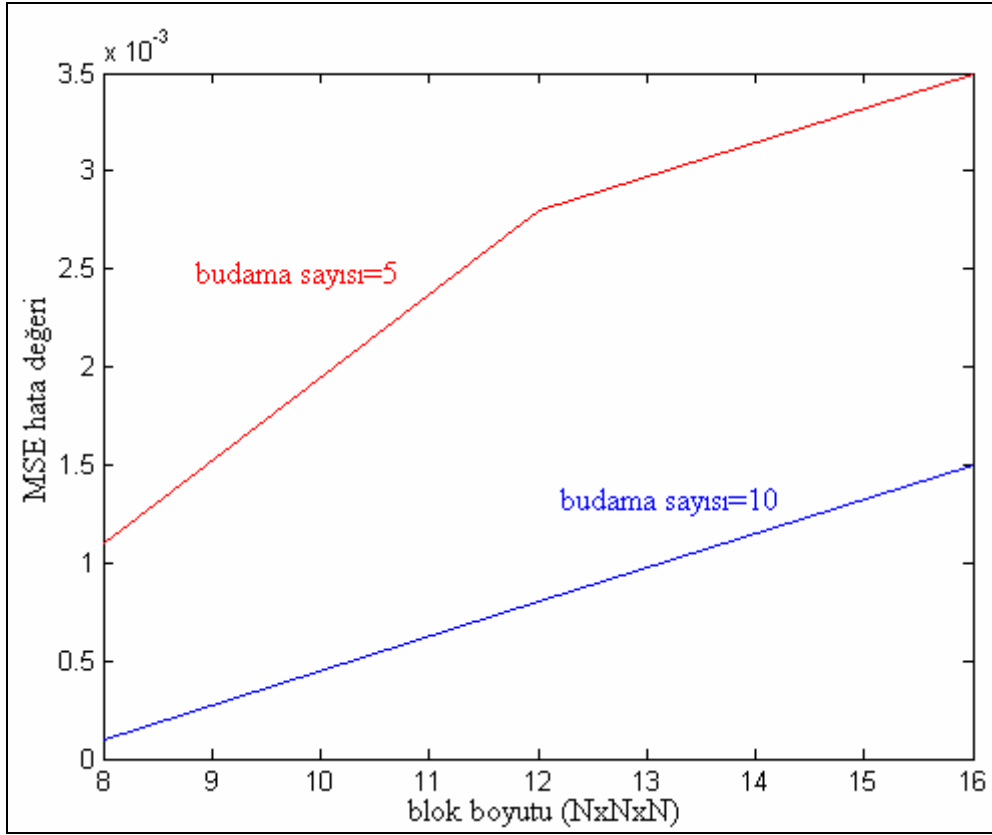
Şekil 5.6. Sıkıştırma, küçük küpler üzerinde yapıldığında, MSE hata değeri ile budama sayısı arasındaki ilişkiyi gösterir grafik

Şekil 5.6'da $N \times N \times N$ lik bloklar halinde sıkıştırma yapılması durumunda, MSE hata değeri ile budama sayısı arasındaki ilişkiyi göstermektedir. Şekil 5.5'e göre budama sayısı arttıkça sıkıştırma oranı azalmaktadır, bu da görüntüdeki hatanın azalması anlamına gelmektedir. $8 \times 8 \times 8$ 'lik bloklar aynı budama sayısında $16 \times 16 \times 16$ 'lık bloklara göre daha az sıkıştırıldığı için daha az hata ile sıkıştırılabilmektedir.



Şekil 5.7. Sıkıştırma, küçük küpler üzerinde yapıldığında, Sıkıştırma oranı ile blok kenar ölçütü arasındaki ilişkiyi gösterir grafik











Şekil 5.7’de blok kenar ölçütü (N) ile sıkıştırma oranı arasındaki ilişki görülmektedir. blok kenar ölçütü arttıkça, daha büyük bloklar yani daha çok AKD katsayıları aynı budama değeri ile budandığında, aynı sayıda AKD katsayısı ile ifade edilmektedir. Bu da daha fazla sıkıştırma anlamına gelmektedir. Budama sayısı düşük iken, aynı görüntü, daha az sayıda katsayı ile ifade edildiğinden sıkıştırma oranı artmaktadır.







Şekil 5.8. Sıkıştırma, küçük küpler üzerinde yapıldığında, MSE hata değeri ile blok kenar ölçütü arasındaki ilişkiyi gösterir grafik

Şekil 5.8'de blok kenar ölçütü arttıkça, MSE hata değerindeki değişim görülmektedir. Şekil 5.7'ye göre blok kenar ölçütü (N) arttıkça sıkıştırma arttığından dolayı, MSE hata değeri de artmaktadır.

Resim 5.1'de budama işleminin görüntü üzerindeki etkisi ve hata görüntüleri görülmektedir.

Bud. Katsay.	Sık. Oranı	MSE hata değeri	Sıkıştırılmış görüntü	Hata görüntüsü
10	99,7205	0,0124		
30	98,3346	0,0051		
50	96,0136	0,0029		
70	92,8686	0,0017		
100	86,8080	$8,8956 \times 10^{-4}$		

Resim 5.1. Uygulanan yöntemin hata analizi

140	76,8038	3.4551 $\times 10^{-4}$		
200	59,8649	8.6336 $\times 10^{-5}$		

Resim 5.1. Devam Uygulanan yöntemin hata analizi

6. SONUÇ VE ÖNERİLER

Göreceli bir kavram olan görüntünün anlaşılabilmesi, görüntü içerisindeki detayların ayrıştırılabilmesine bağlıdır. Bu ayrıştırma, insan beyni için, aynı görüntünün daha önceden hafızada kaydedilmiş olması gibi farklı parametrelere bağlı olduğu gibi, görüntünün fiziksel özelliklerine de bağlıdır. Beyin tarafından daha önce görülmüş olsa da, çok az detay bilgilere sahip görüntüler, ayrıştırılamayabilir, dolayısıyla kalitesi düşük olarak adlandırılır. Kaliteyi belirleyen fiziksel özellik, detay bilgilerin saklanması ve sunulmasıdır.

Saklama alanından tasarruf etmek, düşük hat kapasiteli ağlardan gerçek zamanlı görüntü aktarımı gibi maliyete dayalı sebeplerden ötürü görüntü sıkıştırma yöntemleri uygulanmaktadır. Sıkıştırma oranı, kalite değeriyle ters orantı içerisinde. Bu sebeple, daha yüksek sıkıştırma elde etmek için uygulanan yöntemlerde, kalite değeri kontrol edilebilmelidir. Bu iki parametredeki değişimin en iyi görüldüğü düzlem frekans düzlemi olmasından dolayı, video görüntüleri 3B AKD ile dönüştürüldükten sonra, frekans düzlemindeki AKD katsayıları üzerinde sıkıştırma yapılmaktadır.

Video sıkıştırma işlemi sırasıyla; 3B AKD katsayılarının hesaplanması, katsayıların nicemlenmesi veya budanması, ayrılmış katsayıların sıkıştırılması alt işlemlerinden oluşmaktadır. Nicemleme işlemi sıkıştırma oranı ve görüntü kalitesini uygulayan işlemdir. Tüm bu alt işlemler için, kalite, sıkıştırma oranı, işlem hızı gibi parametreleri eniyileyelemek için çeşitli yöntemler geliştirilmektedir.

Veri kaybı, 3B AKD katsayıları üzerinde sağlandığı için, görüntüdeki periyodik desenler, değişim hızı gibi içeriğe bağlı parametreler de, kalite üzerinde etkili olduğu uygulanan yöntem sonucunda görülmüştür.

Budama sayısı sabit bir çerçeve sayısı için yükseltildiğinde daha çok AKD katsayısı görüntüyü ifade etmek üzere saklanmaktadır, dolayısıyla sıkıştırma oranı azalmaktadır. Geri dönüşümde ise, daha çok frekans bileşenine göre dönüştürme

yapıldığından, görüntüde daha fazla detay elde edilmektedir. Dolayısıyla hata azalmakta, kalite artmaktadır. Sabit bir budama sayısı için, çerçeve sayısı artırıldığında ise aynı sayıda AKD katsayısı ile daha fazla görüntü verisi ifade edildiğinden dolayı sıkıştırma oranı artmakta, buna bağlı olarak ta hata artmakta ve kalite düşmektedir. Sıkıştırma işlemi tüm görüntü karesi üzerinde değilde $N \times N \times N$ 'lik küçük küpler üzerinde yapıldığında ise, budama sayısı arttıkça daha fazla AKD katsayısı saklanmakta ve sıkıştırma oranı ve hata azalmaktadır. Sabit bir budama sayısı için N değeri arttıkça da, daha fazla görüntü verisi aynı sayıda AKD katsayısıyla ifade edilmektedir. Buna bağlı olarak ta sıkıştırma oranı ve hata artmaktadır.

Literatür taramasında ve de uygulanan yöntem sonuçlarında, video sıkıştırma için, 3B AKD sıkıştırmanın, zamansal düzlemdeki benzerliklerin ve aynılıkların sıkıştırılmasını sağlamalarından dolayı iyi sonuç verdiği görülmüştür. Ancak alt işlemlerin kullandıkları yöntemler ve de parametreler, video görüntüleri içeriğine bağlı olarak, en iyi sonucu alabilmek için değişmektedir. Bu sebeple alt işlemlerin ve parametrelerinin adaptif uygulanması gerekmektedir.

Literatür taramasında da karşılaşılan bir yöntem, nicemleme değerlerinin veya budama katsayısının, görüntünün enerjisine göre adaptif seçilmesidir. Bu sayede sıkıştırma oranı istenilen değerde tutulurken, kalitenin değişimi kontrol edilebilmektedir. Önemli olan 3B AKD katsayılarında, anlamlı olanların saklanması, daha az anlamlı olanların ise sıkıştırma alt işlemi uygunluğu için ayrılanmasıdır. Bu yüzden aynı işlem içerisinde adaptif olarak, 3B AKD katsayılarının farklı bölgelerinde farklı nicemleme değerleri matrisinin kullanılması veya budama ve nicemlemenin beraber kullanılabileceği bir yöntem izlenebilir. Ayrıca budama işlemi yüksek frekanslar üzerinde gerçekleştirilirken, anlamlı veriler silinebilmektedir. Bu sebeple yine enerjiye bağlı olarak budama işleminin aralıklarla yapılması, hatta enerji yoğun olmayan bölgelerde de, budamanın uygulanmayarak, aralardaki düşük enerjili yüksek frekanslardaki verilerin korunması sağlanabilir. Bu yöntemler ile görüntü kalitesi istenilen seviyede tutulurken sıkıştırmanın en fazla yapılması sağlanabilir.

Görüntü dizilerinin sıkıştırılmasında kullanılabilecek artıklıklardan bir tanesi, zaman düzleminde ortaya çıkan artıklıklardır. Bu da görüntü içerisindeki harekete bağlı olarak ortaya çıkmaktadır. Hareketin olmadığı durumda ortaya çıkan artıklıklar işlenebileceği gibi, hareket tanıma yapılarak, tüm görüntü bilgisi yerine hareket bilgisi taşınarak da, sıkıştırma yapılabilmektedir.

Hareket tanıma, hem endüstriyel kullanım amacıyla hem de sıkıştırma uygulamasındaki kullanımıyla önem taşımaktadır. Hareket tanıma uzaysal düzlemde yapılabildiği gibi, frekans düzleminde de yapılabilmektedir. AKD düzleminde yapılan hareket tanıma uygulamaları, mevcut görüntü biçimleri ve sıkıştırma teknikleri üzerinde çalışabilmek için daha uygundur. Doğrudan AKD düzleminde yapılan hareket tanıma uygulamasında, sıkıştırılmış verilerin geri açılmasına gerek kalmadan, işlem kazancı sağlanarak daha hızlı uygulamalar geliştirilebilmektedir.

Bu çalışmada sonuç olarak, görüntü dizilerinin işlenmesinde, yaygın olarak iki boyutlu sinyaller üzerinde çalışıldığı görülmüştür. Ancak 3 boyutlu sinyallerin işlenmesi ile görüntü işlemede başarılı sonuçlar elde edilebilmektedir. Yeni yapılacak çalışmaların bu alanda olması, literatürü zenginleştirecektir.

Sözde faz ile hareket tanıma yöntemi, görüntülerin üzerinde yapılan işlemler sayesinde daha başarılı hale getirilmiştir. Görüntü dizisi içerisinde hareket tanıma yapılacak iki görüntüler üzerindeki hata oranı bu yöntemin başarısını olumsuz etkilemektedir. Başarıyı artırabilmek amacıyla, uzaysal düzlemde görüntüye medyan filtre uygulanarak gürültünün temizlenmesi amaçlanmıştır. Medyan filtre, görüntü içerisindeki şekillerin kenarlarını korumaktadır. Böylece hareket eden nesnenin hareket bilgisi kaybolmamakta ve gürültü temizlenerek sözde faz hesaplaması yapılabilmektedir. Nesne ile arka plan arasındaki ayırt edilebilirliğin düşük olmasının, sözde faz hesaplamasını olumsuz etkilediği görülmüştür. Bu sorunun çözümü olarak ta iki yöntem uygulanmış ve her ikisinde de başarılı sonuç alınmıştır. Bu yöntemlerden ilki, uzaysal düzlemde üssel fonksiyon ile gri seviyesi dönüşümü yapılarak, nesnenin daha belirgin hale gelmesi sağlanmıştır. İkinci olarak ta Otsu'nun yöntemi ile eşikleme uygulanarak nesne ile arka planın ayrılması sağlanmıştır.

KAYNAKLAR

1. Chan, R. K. W. ve Lee, M.C., “3D-DCT Quantization as a Compression Technique for Video Sequences”, *1997 International Conference on Virtual Systems and MultiMedia*, Geneva, 188-196 (1997).
2. Walmsley, N. P., Skodras A. N. ve Curtis K. M., “A Fast Picture Compression Technique”, *IEEE Transactions on Consumer Electronics*, 40 (1): 11-19 (1994).
3. Kim, D. ve Shin, D. “Energy-Based Adaptive DCT/IDCT for Video Coding”, *International Conference on Multimedia and Expo 2003*, Baltimore, 557-560 (2003).
4. Arazaki, S., Saigusa, M., Hashiguchi, S. ve Ohki, M. “Image Data Compression by DCT with Adaptive Run-Length Coding”, *IEEE Transactions on Consumer Electronics*, 37 (4), 860-866 (1991).
5. Boussakta, S. ve Alshibami, H. O., “Fast Algorithm for the 3-D DCT-II”, *IEEE Transactions on Signal Processing*, 52 (4), 992-1001 (2004).
6. İnternet: Harvey Mudd Koleji, “Definition of DCT” <http://fourier.eng.hmc.edu/e161/lectures/dct/node1.html> (2004).
7. Kou, W. ve Fjällbrant, T. “A Direct Computation of DCT Coefficient for a Signal Block Taken from Two Adjacent Blocks”, *IEEE Transactions on Signal Processing*, 39 (7): 1692-1695 (1991).
8. Pao, I. ve Sun, M. “Approximation of Calculations for Forward Discrete Cosine Transform”, *IEEE Transactions on Circuits and Systems for Video Technology*, 8 (3): 264-268 (1998).
9. Shen, B. ve Sethi, I. K., “Inner-Block Operations On Compressed Images”, *ACM Multimedia 95 - Electronic Proceedings*, San Fransisco, 489-498 (1995).
10. Hu, Q. ve Panchanathan, S. “Image/Video Spatial Scalability in Compressed Domain”, *IEEE Transactions on Industrial Electronics*, 45 (1), 23-30 (1998).
11. Shen, B., Sethi, I. K. ve Bhaskaran, V., “DCT Convolution and Its Application in Compressed Domain”, *IEEE Transactions on Circuits and Systems for Video Technology*, 8 (8), 947-952 (1998).
12. Khayam, S. A., “Properties of DCT”, *The Discrete Cosine Transform (DCT): Theory and Application*, *Michigan State University*, Michigan, 8-14 (2003).
13. Fryza, T. ve Hanus, S., “Relation Between Character of Real Video Sequences and 3D-DCT Compression”, *4th EURASIP Conference Focused on Video Image Processing and Multimedia Communications*, Zagreb, 107-112 (2003).

14. Kiu, M., Du, X., Moorhead, R. J., Banks, D. C. ve Machiraju, R., “Two Dimensional Sequence Compression Using MPEG”, *Visual communications and Image Processing '98*, San Jose, 914-921 (1998).
15. Westwater, R. ve Furht, B., “Three-Dimensional DCT Video Compression Technique Based on Adaptive Quantizers”, *IEEE International Conference on Engineering of Complex Computer Systems*, Montreal, 189-198 (1996).
16. Bhaskaran, V., Konstantinides, K. ve Beretta, G. “Text and Image Sharpening of Scanned Images in The Jpeg Domain”, *International Conference on Image Processing*, Washington, 326-329 (1997).
17. Lakhani, G., “Optimal Huffman Coding of DCT Blocks”, *IEEE Transactions on Circuits and Systems for Video Technology*, 14 (4), 522-527 (2004).
18. Le Gall, D. J., “The MPEG Video Compression Standard”, *Digest of Papers, Comcon Spring*, San Fransisco, 334-335 (1991).
19. Tudor, P.N., “MPEG-2 video compression”, *Electronics & Communication Engineering Journal*, 257-264 (1995).
20. Yalçın, S., “H.264 Motion Estimator Design”, Yüksek Lisans Tezi, *Sabancı Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 7-8 (2005).
21. Peacock, A.M., Renshaw, D. ve Hannah, J., “Motion direction estimates from differenced DCT images”, *Electronics Letters*, 37 (3): 163-164 (2001).
22. Roma, N. ve Sousa, “Least Squares Motion Estimation Algorithm in the Compressed DCT Domain for H.26x / MPEG-x Video Sequences”, *IEEE Conference on Advanced Video and Signal Based Surveillance*, Como Lago, 576-585 (2005).
23. Xue, X. ve Fan, C., “A New Motion Compensation Algorithm in DCT Domain for H.261 Video Encoder”, *IEEE TENCON - Digital Signal Processing Applications*, Perth, 867-869 (1996).
24. Koc, U. Ve Liu, R., “Discrete-Cosine/Sine-Transform Based Motion Estimation”, *IEEE International Conference on Image Processing*, Austin, 771-775 (1994).
25. Koc, U. Ve Liu, R., “DCT Based Motion Estimation”, *IEEE Transactions on Image Processing*, 7 (7): 948-965 (1998).

EKLER

EK-1 Türkçe-İngilizce terimler karşılığı

Ayrık Kosinüs Dönüşümü:	Discrete Cosine Transform
Ayrık Kosinüs/Sinüs Dönüşümü:	Discrete Cosine/Sine Transform
Ayrık Sinüs/Kosinüs Dönüşümü:	Discrete Sine/Cosine Transform
Ayrık Sinüs Dönüşümü:	Discrete Sine Transform
Ayrık Fourier Dönüşümü:	Discrete Fourier Transform
Görüntü İşleme:	Image Processing
Hareket Tanıma:	Motion Estimation
İnterpolasyon:	Interpolation
Konvolüsyon:	Convolution
Korelasyon:	Correlation
Luminans:	Luminance
Nicemleme:	Quantization
Sıkıştırma:	Compression
Sinüzoidal:	Sinusoidal
Ortogonallik:	Orthogonality
Taban Görüntüler:	Basis Images
Tepi:	Impulse

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ATALAR, Mustafa
Uyruğu : T.C.
Doğum tarihi ve yeri : 31.05.1981 Ankara
Medeni hali : Bekar
Telefon : 0 (312) 280 11 47
e-posta : matalar@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Çankaya Üni./ Bilg. Müh. Böl.	2003
Lise	Eryaman Lisesi	1999

İş Deneyimi

Yıl	Yer	Görev
2004-2006	Birtel İletişim	Ağ Mühendisi
2006-2008	Sibera İletişim	Teknik Müdür

Yabancı Dil

İngilizce

Hobiler

Fotoğraf, teknoloji, doğa sporları