

**TEKİL DÖNÜŞÜM MATRİSLERİ ÜRETEN BLOK
DALGACIK SÜZGEÇ TASARIMI**

Muzaffer DOĞAN

Doktora Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ağustos-2010

JÜRİ VE ENSTİTÜ ONAYI

Muzaffer Doğan'ın "**Tekil Dönüşüm Matrisleri Üreten Blok Dalgacık Süzgeç Tasarımı**" başlıklı **Elektrik-Elektronik Mühendisliği** Anabilim Dalındaki Doktora Tezi 05.07.2010 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

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

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

Enstitü Müdürü

ÖZET

Doktora Tezi

TEKİL DÖNÜŞÜM MATRİSLERİ ÜRETEN BLOK DALGACIK SÜZGEÇ TASARIMI

Muzaffer DOĞAN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ömer Nezih GEREK

2010, 88 sayfa

Bu tezde tekil dönüşüm matrisi üreten dalgacık süzgeç bankaları için çeşitli tasarım yöntemleri ve algoritmalar geliştirilmiştir. İlinti yapısı bilinen sinyaller için en büyük kodlama kazancını veren dönüşüm kodlama yöntemi Karhunen–Loéve dönüşümü olduğundan süzgeç bankalarının hesaplanmasında Karhunen–Loéve dönüşüm matrisi esas alınmış ve verilen bir sinyal için en iyi sıkıştırmayı veren süzgeç bankasının bulunması hedeflenmiştir. Bu amaçla hem nümerik, hem de analitik yöntemler üzerinde çalışılmış ve kullanılan yöntemlerin başarımları deney sonuçlarıyla birlikte sunulmuştur. *Blok Dalgacık Dönüşümü Ters Çevrimi* adı verilen ve kafes parametre belirleme yöntemine dayanan yeni bir yöntemle, hedeflenen dördün yansımali süzgeç bankalarının katsayılarının hesaplanabileceği gösterilmiştir.

Anahtar Kelimeler: Blok Dalgacık Dönüşümü, Süzgeç Bankası Tasarımı, Tekil Dönüşüm Matrisleri, Karhunen–Loéve Dönüşümü, Kafes Parametre Belirleme, Dördün Yansımali Süzgeç Bankası

ABSTRACT

PhD Dissertation

DESIGN OF BLOCK WAVELET FILTERS GENERATING UNITARY TRANSFORM MATRICES

Muzaffer DOĞAN

Anadolu University

Graduate School of Sciences

Electrical and Electronics Engineering Program

Supervisor: Prof. Dr. Ömer Nezh GEREK

2010, 88 pages

In this thesis several methods and algorithms for wavelet filter banks which generate unitary transform matrices are developed. It is aimed to find the filter bank which gives the best compression by using Karhunen–Loève transform matrix since Karhunen–Loève transform is the transform coding method which gives the best coding gain for a class of signals with known correlation structures. For this purpose, some numerical and analytical methods are studied and the successes of the methods are presented with the experimental results. It is shown that the coefficients of the aimed quadrature mirror filter banks can be computed by the new lattice parameterization based method which is named as *Block Wavelet Transform Inversion*.

Keywords: Block Wavelet Transform, Filter Bank Design, Unitary Transform Matrices, Karhunen–Loève Transform, Lattice Parameterization, Quadrature Mirror Filter Bank

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
TERİMLER DİZİNİ.....	ix
1. GİRİŞ	1
2. DÖNÜŞÜM KODLAMA YÖNTEMLERİ	6
2.1. KLT	6
2.1.1. KLT'nin Optimumluğu.....	7
2.1.2. Resimler Üzerinde KLT	12
2.2. DCT	16
3. BLOK DALGACIK DÖNÜŞÜMÜ	18
3.1. Dalgacık Dönüşümü	18
3.2. Çoklu Çözünürlük	21
3.3. QMF Süzgeç Bankaları	22
3.4. Blok Dalgacık Dönüşümü	24
3.4.1. BWT Matrisi	25
3.4.2. BWT Matrisinin Tekilliği	26
3.4.2.1. Tek Aşamalı BWT Matrisinin Tekilliği.....	26
3.4.2.2. İki Aşamalı BWT Matrisinin Tekilliği	29
3.4.2.3. İkiden Fazla Aşamalı BWT Matrisinin Tekilliği	34
4. QMF SÜZGEÇ BANKASI TASARIMI	36
4.1. En Dik İniş Yöntemiyle Süzgeç Tasarımı	36
4.1.1. BWT Matrisinin Parametrik Hale Getirilmesi.....	36
4.1.2. KLT Matrisini Üreten Süzgeçlerin En Dik İnişle Bulunması	40

4.1.3. En Dik İniş Algoritması.....	42
4.1.4. Örnek Uygulamalar	43
4.2. Kafes Parametre Belirleme Yöntemi.....	47
4.2.1. Kafes Parametre Belirleme ile Nümerik Yöntem.....	49
4.2.1.1. Kafes Parametre Belirleme ile En Dik İniş Algoritması.....	49
4.2.1.2. db2 Dönüşüm Matrisini Üreten Süzgeç Bankasının Bulunması	51
4.2.1.3. db4 Dönüşüm Matrisini Üreten Süzgeç Bankasının Bulunması	53
4.2.1.4. Lena'nın KLT Matrislerini Üreten Süzgeç Bankasının Bulunması	55
4.3. BWT Ters Çevrimi	61
4.3.1. Performans Analizi	66
4.4. 8×8 Boyutunda BWT Ters Çevrimi	71
5. SONUÇLAR	82
KAYNAKLAR	85

ŞEKİLLER DİZİNİ

2.1.	İlintili iki sinyalin saçılım grafiği	11
2.2.	KLT uygulandıktan sonraki saçılım grafiği.....	11
2.3.	Lena resminin KLT'sinden elde edilen ilk 50 katsayı	13
2.4.	Lena resmi ve Lena resminin ilk KLT taban resmi	14
2.5.	İlk KLT katsayıları kullanılarak geri çatılan Lena resimleri	15
2.6.	Lena resmi için 8×8 boyutlu taban resimleri	16
2.7.	8×8 boyutlu DCT taban matrisleri	17
3.1.	Ölçekleme fonksiyonlarının gerdiği iç-içe uzaylar.....	20
3.2.	3 aşamalı çoklu-çözünürlük ağaç yapısı	22
3.3.	Çoklu-çözünürlüklü yaklaşım ve detay sinyallerinden orijinal sinyalin geri çatılması	22
3.4.	Tek aşamalı blok dalgacık sistemi	24
3.5.	İki aşamalı blok dalgacık sistemi.....	25
4.1.	Nümerik yöntemde Daub-4 dönüşüm matrisini elde etmek için Haar süzgeçleriyle başlandığında elde edilen hata grafiği	44
4.2.	Nümerik yöntemde Daub-4 dönüşüm matrisini elde etmek için rastgele bir süzgeçle başlandığında hata grafiği	46
4.3.	db2'nin BWT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği	52
4.4.	db4'ün BWT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği.....	54
4.5.	KLT ile BWT matrisleri arasındaki satır eşleştirmesi	57
4.6.	Lena resminin satırlarına ait KLT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği	58
4.7.	Lena resminin sütunlarına ait KLT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği	61
4.8.	Test resimleri (Lena, Mandril, Peppers ve Bridge)	62
4.9.	Satırların ardışıklıklarının eşleştirilmesi için BWT ile KLT matrisleri arasında yapılması gereken sütun değişiklikleri	64

4.10. Lena ve Mandril test resimlerinin satırlarından elde edilen kodlama kazancı ve toplam kare hata grafikleri	67
4.11. Peppers ve Bridge test resimlerinin satırlarından elde edilen kodlama kazancı ve toplam kare hata grafikleri.....	68

ÇİZELGELER DİZİNİ

4.1. db4'ün dönüşüm matrisinin Haar süzgeçleriyle başlanarak nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-5}$ için μ adım büyüklüğüne karşı iterasyon sayıları	45
4.2. db2'nin dönüşüm matrisinin $\alpha = 0$ ile başladığında kafes parametre belirlemeli nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-6}$ için μ adım büyüklüğüne karşı iterasyon sayıları	53
4.3. db4'ün dönüşüm matrisinin $t_0 = 4.6790$, $t_1 = 1.1872$, $t_2 = 4.3151$ ile başladığında kafes parametre belirlemeli nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-5}$ için μ adım büyüklüğüne karşı iterasyon sayıları ...	55
4.4. Test resimlerinde toplam kare hatanın minimum, kodlama kazancının maksimum olduğu açı değerleri ve karşılık gelen kodlama kazançları	69
4.5. Test resimleri üzerinde Daubechies-2 süzgeçleriyle elde edilen kodlama kazançları	69
4.6. Test resimlerinin satır ve sütunlarından oluşturulan KLT matrisleri için elde edilen alçak-geçiren süzgeç katsayıları	70
4.7. BWT ters çevrimi, Daubechies-2 ve maksimum kodlama kazanç noktası yöntemleri ile elde edilen kanal varyansları	72
4.8. Test resimleri üzerinde en büyük kodlama kazancı veren BWT matrisinin ikinci satırının büyükten-küçüğe sıralamasını veren sütun sıralaması	75
4.9. 8×8 durumunda test resimleri üzerinde BWT ve KLT matrislerinin eşleştirilmeleriyle elde edilen kafes parametreleri, kodlama kazançları ve Daub-4 süzgeç bankasıyla elde edilen kodlama kazançları	78
4.10. 8×8 durumunda BWT ve KLT matrislerinin eşleştirilmesiyle elde edilen kanal varyanslarının Daub-4 süzgeç bankası ile elde edilen kanal varyanslarıyla karşılaştırılması	79

SİMGELER VE KISALTMALAR DİZİNİ

BWT	: Blok Dalgacık Dönüşümü (<i>Block Wavelet Transform</i>)
DCT	: Ayrık Kosinüs Dönüşümü (<i>Discrete Cosine Transform</i>)
DFT	: Ayrık Fourier Dönüşümü (<i>Discrete Fourier Transform</i>)
DST	: Ayrık Sinüs Dönüşümü (<i>Discrete Sine Transform</i>)
DWHT	: Ayrık Walsch-Hadamard Dönüşümü (<i>Discrete Walsch-Hadamard Transform</i>)
DWT	: Ayrık Dalgacık Dönüşümü (<i>Discrete Wavelet Transform</i>)
EBCOT	: En-iyi Kırpma ile Gömülü Blok Kodlama (<i>Embedded Block Coding with Optimal Truncation</i>)
EZBC	: Gömülü Sıfır-Blok Kodlama (<i>Embedded Zeroblock Coding</i>)
EZW	: Gömülü Sıfır-Ağaç Kodlama (<i>Embedded Zerotree Wavelet</i>)
FFT	: Hızlı Fourier Dönüşümü (<i>Fast Fourier Transform</i>)
JPEG	: <i>Joint Photographic Experts Group</i>
KLT	: Karhunen–Loève Dönüşümü (<i>Karhunen–Loève Transform</i>)
MPEG	: <i>Moving Picture Experts Group</i>
SPIHT	: Hiyerarşik Ağaçlarda Küme Bölüntüleme (<i>Set Partitioning In Hierarchical Trees</i>)
SSWT	: Uzamsal Bölütlenmiş Dalgacık Dönüşümü (<i>Spatially Segmented Wavelet Transform</i>)
STFT	: Kısa-Zamanlı Fourier Dönüşümü (<i>Short-Time Fourier Transform</i>)
QMF	: Dördün Yansımali Süzgeç (<i>Quadrature Mirror Filter</i>)

TERİMLER DİZİNİ

alçak-geçiren	: low-pass
altbant	: subband
altbant ayrıştırma	: subband decomposition
analiz	: analysis
ardışıklık	: sequency
aslına benzerlik	: fidelity
ayrık	: discrete
ayrıştırma	: decomposition
beklenen değer	: expectation value
birimdik	: orthonormal
bozulum	: distortion
çok-fazlı	: plyphase
çok-ölçekli	: multi-scale
çoklu-çözünürlük	: multi-resolution
dalgacık	: wavelet
dalgacıklanma	: ripple
detay	: detail
devrik	: transpose
dışdeğerleme	: extrapolation
diklik	: orthogonality
dönüşüm	: transform
dönüşüm kodlama	: transform coding
dördün yansımali süzgeç	: quadrature mirror filter
dürtü tepkisi	: impulse response
eğer nokta	: saddle point
en dik iniş	: steepest descent
eşleme	: mapping
faz	: phase
genlik	: amplitude
geri çatma	: reconstruction

ikikat	: octave
ilinti	: correlation
izdüşüm	: projection
izgesel	: spektral
kafes parametre belirleme	: lattice parameterization
kaplam	: extent
karşı-dik	: biorthogonal
kayıplı sıkıştırma	: lossy compression
kayıpsız sıkıştırma	: lossless compression
kesikli	: discrete
kodlama kazancı	: coding gain
köşegenleme	: diagonalization
nicemleme	: quantization
örtüşme	: aliasing
özbağlanımlı	: autoregressive (AR)
özdeğer	: eigenvalue
özilinti	: autocorrelation
özvektör	: eigenvector
rastgele süreç	: random process
sentez	: synthesis
sıkıştırma	: compression
sürekli	: continuous
süzgeç	: filter
süzgeç bankası	: filter bank
taban	: basis
taban resmi	: basis image
tam geri çatılma	: perfect reconstruction
tekil	: unitary
yaklaşım	: approximation
yüksek-geçiren	: high-pass

1. GİRİŞ

Sinyal işleme, günümüzde birçok uygulamada karşımıza çıkmaktadır. Haberleşme, havacılık ve uzay bilimleri, devre tasarımı, akustik, sismoloji, biyomedikal mühendisliği, enerji üretimi ve dağıtımı, kimyasal süreç kontrolü ve konuşma işleme gibi çok çeşitli alanlarda sinyal işleme uygulamaları kullanılmaktadır [1].

Sinyal işleme uygulamalarının büyük bir kısmında, verilen bir sistemin karakteristiği ve çeşitli girdilere nasıl çıktı verdiği incelenir. Diğer bir kısmında ise istenen amaca yönelik bir sistem tasarlanmaya çalışılır. Üzerinde çalışılan sinyal sürekli olabileceği gibi kesikli de olabilir [1].

Bir sinyalin Fourier dönüşümü alındığında sinyal tamamen frekans alanında ifade edilmekte ve zaman bilgisi gizlenmektedir. Bundan dolayı, Fourier dönüşümü ile bir sinyalde bir bileşenin varlığı saptanabilir, fakat bu bileşenin sinyalin hangi zaman aralığında bulunduğu bilinemez. Bu problemin üstesinden gelmek için *kısa-zamanlı Fourier dönüşümü* (STFT) geliştirilmiştir. Böylece önceden belirlenmiş bir pencere büyüklüğündeki zaman aralıklarında sinyalin frekans dağılımı incelenebilir ve sinyalde bir değişim meydana geldiğinde zaman bilgisiyle birlikte yeri tespit edilebilir.

Dalgacıklar da son on yıllarda geliştirilmiş, kısa zamanlı Fourier dönüşümü gibi belli zaman aralıklarındaki değişimleri ortaya çıkarmak için kullanılabilen yeni bir sinyal işleme yöntemidir. Dalgacıklar ile sinyaller farklı *ölçek* ve *çözünürlüklerde* incelenebilir. Dalgacıklar, sinyalin kaba bir yaklaşımını çıkarmasıyla sinyalin uzun zamanlı davranışını ortaya koyabildiği gibi, giderek küçülen pencere boyutuyla sinyaldeki düzensizlikleri de ortaya çıkartabilir [2]. Bu tarz sinyal işleme yöntemleri, veri ve sinyal sıkıştırma uygulamalarında da kullanılmaktadır.

Hızla büyüyen İnternet üzerinde gittikçe yaygınlaşan video iletimi, mobil iletişimin yaygınlaşması ve sayısal televizyon yayınlarının çoğalması verilerin sıkıştırılmasının önemini arttırmıştır.

Sıkıştırma, verilerin daha az sayıda bitlerle ifade edilmesi demektir. Sıkıştırma algoritmaları iki geniş sınıfa ayrılır: Geri çatılan veriyle (çıkıtı) orijinal

verinin (girdi) tıpatıp aynı olduğu *kayıpsız sıkıştırma* ve kayıpsız sıkıştırmaya oranla daha fazla sıkıştırma oranı sağlayan, fakat çıktının girdiden farklı olduğu *kayıplı sıkıştırma*. Metin sıkıştırma gibi orijinal veri ile geri çatılan veri arasında farklılığa müsaade edilmeyen uygulamalarda kayıpsız sıkıştırma algoritmaları kullanılır. Telefon konuşmaları, görüntülerin kodlanması ve video iletişimi gibi uygulamalarda ise verinin kendisinden çok içeriği önemli olduğundan belli bir miktara kadar kayıp tolere edilebilir. Bu yüzden bu tür uygulamalarda kayıplı sıkıştırma algoritmaları kullanılmaktadır [3].

Kayıplı sıkıştırmada kullanılan yöntemlerden biri de *dönüşüm kodlamadır*. Dönüşüm kodlama yöntemlerinde, ters çevrilebilir ve doğrusal bir dönüşüm (Fourier dönüşümü gibi) kullanılarak görüntü yeni bir dönüşüm katsayıları kümesine eşlenir. Bu katsayılar daha sonra nicemlenir ve kodlanır. Doğal görüntüler için katsayılar küçük olduğundan nicemleme kaba bir şekilde yapıldığında bir miktar bozuluma orijinal görüntü geri çatılabilir [4].

Kayıplı sıkıştırma yöntemleri içerisinde en iyi bilinen veri-bağımlı dönüşüm KLT'dir. Verinin kodlanmasında kullanılan KLT matrisinin satırları, veriyi ürettiği varsayılan rastgele sürecin özilinti matrisinin özvektörlerinden oluşur. Sadece o rastgele sürece has olan bu matris kullanılarak oluşturulan dönüşüm, diğer dönüşüm kodlama yöntemlerine oranla en büyük kodlama kazancını verir [3]. KLT ile kodlanmış bir sinyalin alıcı tarafında çözülebilmesi için kodlanmış sinyal yanında özilinti ya da dönüşüm matrisinin de bilinmesi gerekir. Fazladan gönderilen bu bilgiler, optimum dönüşümün sağladığı avantajları ortadan kaldıracak kadar büyük olabilir. Fakat sinyal istatistiklerinin çok fazla değişmediği ve dönüşüm matrisinin küçük tutulabildiği uygulamalarda KLT kullanışlı olabilir [5].

Bir sıkıştırma algoritmasının performansı, algoritmanın göreceli karmaşıklığına, algoritmayı çalıştırmak için gereken bellek miktarına, sıkıştırmanın verilen bir bilgisayarda ne kadar süre aldığına, elde edilen sıkıştırmanın miktarına ve sıkıştırıldıktan sonra geri çatılan verinin orijinaline ne kadar benzediğine göre değerlendirilebilir. Kayıplı sıkıştırma yöntemlerinde orijinal veri ile geri çatılan veri farklı olduklarından algoritmanın performansını

ölçmek için *bozulum*, *aslına benzerlik* ve *kalite* gibi performans ölçütleri kullanılmaktadır [3].

Ayrık kosinüs dönüşümü (DCT); JPEG, MPEG ve CCITT H.261 gibi birçok uluslar arası görüntü ve video kodlama standardında kullanılan popüler bir dönüşümdür. Birçok görüntü yüksek ilinti katsayılı bir Markov kaynağı olarak modellenebildiği için DCT, KLT'ye yakın bir sıkıştırma sağlar [3].

Sinyal ayrıştırma konusundaki araştırmalar genellikle görsel sinyal işleme ve kodlama uygulamaları için yapıldığından insan görme sistemi incelenerek elde edilen bulgular sinyal ayrıştırma adımları içerisinde kullanılmıştır. İnsan görme sisteminin doğası gereği çoklu-çözünürlüklü sinyal işleme yapıyor olmasının bulunmasından sonra çoklu-çözünürlüklü ayrıştırmaya olan ilgi artmıştır. Birçok uygulama için görüntü veya videonun düşük, yani kaba çözünürlüklü şekli yeterli olurken; bilgisayar görüsü ve görsel iletişim gibi uygulamalarda sinyal kalitesinin kabadan inceye doğru aşamalı bir şekilde iyileşmesi önem kazanmaktadır [6].

Sıkıştırmada kullanılan dönüşüm kodlama yöntemlerinden biri de *altbant kodlamadır*. Altbant kodlamada girdi verisi süzgeçler kullanılarak parçalara ayrılır. Bu parçalama işlemi sonucunda elde edilen çıktıların eleman sayısı, orijinal verininki ile aynı olmasına rağmen çıktıların karakteristikleri farklı olduğundan her bir çıktı üzerinde farklı bir sıkıştırma uygulanabilir ve toplamda yüksek bir sıkıştırma oranı elde edilebilir [3].

Altbant ayrıştırmasında kullanılan süzgeçlere *analiz süzgeçleri* denir. Ayrıştırılan çıktıların geri çatılması için kullanılan süzgeçlere ise *sentez süzgeçleri* denir. Analiz ve sentez süzgeçlerine beraber *süzgeç bankası* adı verilmektedir. Süzgeç bankası tasarımında göz önünde bulundurulacak özelliklerden biri süzgeçlerin dikliğidir. Dik süzgeç bankalarının yanısıra karşı-dik süzgeç bankaları da tasarlanabilir [7].

Altbant ayrıştırımında kullanılan dönüşümün etkinliği, enerji yoğunluğunun ne şekilde ayrıştırıldığı ile ölçülebilir. Bu ölçüm yöntemlerinden biri, dönüşüm katsayılarının varyanslarının aritmetik ortalamasının geometrik ortalamasına bölünmesiyle elde edilen *kodlama kazancıdır* [3]. Bu tezde geliştirilen süzgeç bankalarının performansı, altbant ayrıştırımı sonunda elde edilen kanallardaki kodlama kazancı ile ölçülecektir.

Dönüşüm kodlama yöntemlerinden biri olan *ayrık dalgacık dönüşümü* (DWT), çok-ölçekli sinyal ayrıştırma ve analiz için kullanışlı bir sinyal işleme aracıdır ve görüntü işlemede, özellikle görüntü sıkıştırma, geniş bir şekilde kullanılmaktadır [8]. DWT kodlaması, *ikikat altbant kodlaması (octave subband coding)* olarak da düşünülebilir. Ayrık dalgacık dönüşümüyle elde edilen görüntünün enerjisi az sayıda bantta yoğunlaştığından, EZW [9], SPIHT [10], EZBC [11] ve EBCOT [12] gibi çeşitli kodlayıcılar tasarlanmış, JPEG 2000 ve MPEG-Layer3'te de standartlaşmıştır.

Blok dalgacık dönüşümü (BWT) de görüntülerin kodlanmasında kullanılabilen özel bir dalgacık dönüşümüdür [13]. JPEG sıkıştırma algoritması içerisinde DCT matrisi yerine BWT matrisi kullanılarak görüntüler kodlanabilir. BWT matrisi, dalgacık süzgeç bankalarının altbant ayrıştırmasına uygulanmasıyla elde edilebilirler ve BWT matrisleri tekildir [14].

Blok dalgacık dönüşümündeki önemli nokta, görüntü sıkıştırma kullanılacak olan dönüşüm matrisini üreten süzgeç bankasının bulunmasıdır. Süzgeç bankası olarak, bilinen bazı dalgacık süzgeç bankaları kullanılabileceği gibi, sinyale has bir süzgeç bankası da kullanılabilir. Böylece sabit süzgeç bankalarına göre daha yüksek kodlama kazancı elde edilebilir.

Hem KLT matrisi, hem de BWT matrisi tekil olduğundan, BWT olarak ürettiği dönüşüm matrisi KLT'ye karşılık gelen bir süzgeç bankasının bulunabileceği düşünülmektedir. Bulunacak olan bu süzgeç bankası, bu tezde *BWT süzgeç bankası* olarak adlandırılacaktır. Böylece görüntü KLT matrisi kullanılarak kodlanmasına rağmen, büyük boyutlu dönüşüm matrisi yerine bu matrisi üreten süzgeç bankasının DWT formunda kullanılması mümkün olacaktır.

Süzgeç tasarımında kullanılan yöntemlerde genellikle bazı koşullar göz önünde bulundurulur ve bu koşulları optimize edecek şekilde süzgeç bankaları tasarlanır. Örneğin *maxflat* türündeki süzgeç bankalarında süzgeçlerin frekans tepkisinin $\omega = 0$ ve $\omega = \pi$ frekanslarında maksimum düzlükte olması hedeflenir. Ayrıca frekans tepkisindeki dalgacıklaşmaların enerjisini azaltacak şekilde süzgeç bankaları da tasarlanmıştır [6].

Bu tezde ise tekil dönüşüm matrisi üreten dalgacık süzgeç bankalarının hesaplanması için çeşitli tasarım yöntemleri ve algoritmalar geliştirilmiştir.

Süzgeç bankasının hesaplanmasında Karhunen–Loève dönüşüm (KLT) matrisi esas alınarak, verilen bir sinyalle aynı istatistiksel özellikler taşıyan sinyaller için en iyi sıkıştırma veren süzgeç bankasının bulunması hedeflenmiştir. Bu bakımdan önceki süzgeç bankalarının tasarımında kullanılan yöntemlerden farklı bir yöntem denenmiş olmaktadır. Süzgeç bankalarının hesaplanmasında hem nümerik, hem de analitik yöntemler üzerinde çalışılmış ve belirli test resimleri üzerinde yapılan deneylerle kodlama kazançları ve kodlama sonunda elde edilen kanalların varyansları hesaplanmıştır. Deney sonucunda, tasarlanan süzgeçler kullanıldığında maksimum kodlama kazancına yakın kodlama kazançları elde edildiği ve altbant ayrıştırmasında enerjinin büyük bölümünün ilk altsinyalde toplandığı görülmüştür.

Yapılan çalışmalardaki konu bütünlüğünün sağlanması amacıyla ikinci bölümde KLT'den bahsedilecektir. Üçüncü bölümde ise BWT tanıtılacak ve bunun öncesinde genel dalgacık dönüşümleri ve çoklu-çözünürlük konuları kısaca özetlenecektir. Dördüncü bölümde ise doktora çalışması sırasında yapılan BWT süzgeç bankası elde etme çalışmaları anlatılacaktır. Yapılan nümerik çalışmalar *En Dik İniş Yöntemiyle Süzgeç Tasarımı* başlığıyla sunulacaktır. Sonrasında ise serbest parametre sayısını azaltmak için kullanılan *Kafes Parametre Belirleme* yönteminden bahsedilecek ve son olarak önceki yöntemlere oranla daha hızlı bir şekilde süzgeç bankası tasarlamayı sağlayan ve *BWT Ters Çevrimi* ismi verilen yöntem detaylı bir şekilde anlatılacaktır. Yapılan bütün çalışmalar için deney sonuçları da sunulacaktır.

2. DÖNÜŞÜM KODLAMA YÖNTEMLERİ

Dönüşüm kodlama yöntemlerinde kodlanacak veriler, daha yüksek kodlama kazancı sağlayan başka bir diziye dönüştürülürler. Orijinal verideki bilginin çoğunluğu, dönüşüm sonunda az sayıdaki elemanda toplanabilirse veri sıkıştırması elde edilir. Bir verideki bilgi miktarı, verinin entropisiyle orantılıdır. Entropi varyansa bağlı olduğundan verinin hangi elemanda daha fazla toplandığını anlamak için dönüşüm sonundaki varyanslar incelenir [3].

Bu bölümde KLT ve DCT dönüşüm kodlama yöntemleri üzerinde durulacaktır. Yapılan çalışmada KLT kullanıldığı için bu dönüşüm detaylı bir şekilde anlatılırken DCT daha kısa bir şekilde anlatılacaktır.

2.1. KLT

İnsanların boyları ve ağırlıkları arasında yüksek bir ilinti vardır. Örneğin A.B.D.'de ergen erkeklerin ortalama boyu 176 cm. iken ortalama ağırlıkları 87 kg.'dır [15]. Buna göre 176 cm. boyundaki bir insanın ağırlıklarının 87 kg.'dan uzak değerlerde (mesela 20 kg. veya 150 kg.) bulunma olasılığının çok küçük olduğunu söyleyebiliriz. Aralarında bu tür ilinti bulunan veriler, iki boyutlu uzayda geometrik bir dönme dönüşümüyle yüksek sıkıştırma oranlarında kodlanabilirler. Dönme dönüşüm matrisinin elemanlarının, verinin istatistik bilgileri kullanılarak hesaplanabileceğini ilk ortaya koyan Hotelling [16] olmuştur. Benzer dönüşüm, sürekli fonksiyonlar üzerinde Karhunen [17] ve Loève [18] tarafından elde edilmiştir. Bu tür dönüşümler sıkıştırmada ilk defa Kramer ve Mathews [19] ile Huang ve Schultheiss [20] tarafından kullanılmıştır ve literatürde *Karhunen–Loève dönüşümü* (*KLT*) olarak adlandırılmaktadır.

KLT, bant enerjilerini derli toplu hale getirme bakımından en iyi dönüşümdür fakat dönüşüm katsayıları hem veri bağımlı olduğu hem de hesaplanması yavaş olduğu için pratikte DCT kadar yaygın kullanılmamaktadır.

İlerleyen bölümlerde KLT'nin optimum dönüşüm kodlama yöntemi olduğu gösterilip resimler üzerinde KLT'nin nasıl uygulandığı anlatılacaktır.

2.1.1. KLT'nin Optimumlugu

\mathbf{x} , n boyutlu reel elemanlı rastgele bir vektör; $\{\mathbf{u}_j, j = 1, 2, \dots, n\}$, elemanları n boyutlu reel elemanlı birer vektör olan birimlik bir vektör kümesi olsun. \mathbf{x} vektörü, $\{\mathbf{u}_j, j = 1, 2, \dots, n\}$ tabanında şu şekilde yazılabilir:

$$\mathbf{x} = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \dots + y_n \mathbf{u}_n. \quad (2.1)$$

Buradaki $y_j, j = 1, 2, \dots, n$ katsayıları, \mathbf{x} vektörünün $\{\mathbf{u}_j, j = 1, 2, \dots, n\}$ tabanındaki gösterimidir. $\{\mathbf{u}_j, j = 1, 2, \dots, n\}$ tabanı birimlik olduğundan taban vektörlerinin iç çarpımları Kronecker delta fonksiyonuyla ifade edilebilir:

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \quad (2.2)$$

\mathbf{x} vektörünün \mathbf{u}_i vektörüyle iç çarpımı y_i katsayısını verir:

$$\mathbf{u}_i^T \mathbf{x} = y_i \mathbf{u}_i^T \mathbf{u}_i + \sum_{\substack{j=1 \\ j \neq i}}^n y_j \mathbf{u}_i^T \mathbf{u}_j = y_i. \quad (2.3)$$

y_i katsayısı aynı zamanda \mathbf{x} vektörünün \mathbf{u}_i vektörü üzerine izdüşümünün yönlendirilmiş uzunluğu olarak da düşünülebilir.

$m \leq n$ olmak üzere \mathbf{x} vektörünün bir yaklaşımı, $\tilde{\mathbf{x}}$, (2.1)'deki ilk m terimin toplamı olsun:

$$\tilde{\mathbf{x}} = y_1 \mathbf{u}_1 + y_2 \mathbf{u}_2 + \dots + y_m \mathbf{u}_m = \sum_{i=1}^m y_i \mathbf{u}_i. \quad (2.4)$$

\mathbf{x} ve $\tilde{\mathbf{x}}$ vektörleri arasındaki fark, (2.1)'de bulunup (2.4)'te kullanılmayan terimlerdir:

$$\boldsymbol{\varepsilon} = \mathbf{x} - \tilde{\mathbf{x}} = \sum_{i=m+1}^n y_i \mathbf{u}_i. \quad (2.5)$$

$\tilde{\mathbf{x}}$ yaklaşımının orijinal \mathbf{x} vektörüne en yakın seçilebilmesi için en-küçüklenecek bir hata fonksiyonu, $\boldsymbol{\varepsilon}$ vektörünün elemanlarının karelerinin toplamının beklenen değeri olsun:

$$e = E\{|\boldsymbol{\varepsilon}|^2\} = E\{\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}\}. \quad (2.6)$$

Denklem (2.5)'ten

$$e = E \left\{ \left(\sum_{i=m+1}^n y_i \mathbf{u}_i^T \right) \left(\sum_{j=m+1}^n y_j \mathbf{u}_j \right) \right\} \quad (2.7)$$

bulunur. Denklem (2.2)'de verilen birimdiklik koşulu uygulandığında hata fonksiyonu,

$$e = \sum_{i=m+1}^n E\{y_i^2\} \quad (2.8)$$

şeklinde hesaplanır.

Diğer taraftan, bu denklemdeki $E\{y_i^2\}$ değerlerinin hesaplanması için (2.3)'te verilen y_i katsayıları kullanılabilir:

$$E\{y_i^2\} = E\{(\mathbf{u}_i^T \mathbf{x})(\mathbf{x}^T \mathbf{u}_i)\} = \mathbf{u}_i^T E\{\mathbf{x} \cdot \mathbf{x}^T\} \mathbf{u}_i. \quad (2.9)$$

Buradaki $E\{\mathbf{x} \cdot \mathbf{x}^T\}$ terimi, x vektörünün özilinti matrisidir ve \mathbf{R}_x şeklinde gösterilir. Reel elemanlı özilinti matrisleri simetrik Toeplitz matrisleridir ve pozitif yarı-tanımlıdır [21].

Denklem (2.9)'da hesaplanan beklenen değer (2.8)'de yerine konursa,

$$e = \sum_{i=m+1}^n \mathbf{u}_i^T \mathbf{R}_x \mathbf{u}_i \quad (2.10)$$

bulunur. Bu denklemde verilen hata fonksiyonu en-küçükleyecek şekilde $\mathbf{u}_i, i = m + 1, m + 2, \dots, n$ taban vektörlerini hesaplamaya çalışıyoruz. Bu taban vektörleri birimdik olduklarından, (2.2)'de verilen $\mathbf{u}_i^T \mathbf{u}_i = 1$ kısıtlarını da sağlamalıdır. Hata fonksiyonunun en-küçüklenmesi için bu kısıtlar (2.10)'a Lagrange çarpanlarıyla birlikte eklenmelidir:

$$e' = \sum_{i=m+1}^n \mathbf{u}_i^T \mathbf{R}_x \mathbf{u}_i + \sum_{i=m+1}^n \lambda_i (1 - \mathbf{u}_i^T \mathbf{u}_i). \quad (2.11)$$

Bu terim, hata fonksiyonunu en-küçüklerken aynı zamanda taban vektörlerinin uzunluğunun (normunun) 1 olmasını sağlayan son hata fonksiyonu olacaktır.

Optimize edilmeye çalışılan değişkenler \mathbf{u}_i olduğundan, en-küçükleme işlemi basitçe e' teriminin \mathbf{u}_i terimlerine göre kısmî türevlerinin sıfıra eşitlenmesiyle yapılır:

$$\frac{\partial e'}{\partial \mathbf{u}_i} = 0 \Rightarrow 2(\mathbf{R}_x \mathbf{u}_i - \lambda_i \mathbf{u}_i) = 0, \quad i = m + 1, m + 2, \dots, n. \quad (2.12)$$

$\mathbf{R}_x \mathbf{u}_i - \lambda_i \mathbf{u}_i = 0$ denklemi, doğrudan doğruya \mathbf{R}_x özilinti matrisinin özvektörlerinin bulunması problemidir. Hesaplanmaya çalışılan \mathbf{u}_i vektörleri, \mathbf{R}_x matrisinin özvektörleridir. Denklem (2.11)'de kullanılan λ_i Lagrange katsayıları ise bu özvektörlere karşılık gelen özdeğerlerdir.

Denklem (2.10)'daki $\mathbf{R}_x \mathbf{u}_i$ teriminin yerine (2.12)'de elde edilen $\lambda_i \mathbf{u}_i$ terimi yazılırsa, optimum \mathbf{u}_i tabanındaki yaklaşımın verdiği hata,

$$e = \sum_{i=m+1}^n \mathbf{u}_i^T \mathbf{R}_x \mathbf{u}_i = \sum_{i=m+1}^n \mathbf{u}_i^T \lambda_i \mathbf{u}_i = \sum_{i=m+1}^n \lambda_i \mathbf{u}_i^T \mathbf{u}_i \quad (2.13)$$

şeklinde hesaplanır. Denklem (2.2)'deki birimdiklik koşulundan $\mathbf{u}_i^T \mathbf{u}_i = 1$ olduğundan,

$$e = \sum_{i=m+1}^n \lambda_i \quad (2.14)$$

elde edilir. Bu denkleme göre (2.4)'te verilen yaklaşımın (2.1)'deki orijinal vektöre (2.10)'da verilen hata fonksiyonu bağlamında en yakın olması için (2.4)'te ihmal edilen terimler, \mathbf{R}_x matrisinin en küçük özdeğerli özvektörleri üzerine yapılan izdüşümlerden seçilmelidir.

Bu çıkarım, belirli bir m değeri için yapılmış olsa da, herhangi bir m değeri için sonuç genellenebilir. Buna göre herhangi bir m değerine göre dönüşümün optimum olması için vektörün \mathbf{R}_x özilinti matrisinin özdeğerleri büyükten küçüğe sıralanmalı ve bu sıralı özdeğerlere karşılık gelen özvektörler aynı sırayla yan yana dizilerek bir dönüşüm matrisi oluşturulmalıdır. Bu durumda, en sonda kalan $k = n - m$ özdeğerin toplamı, ilk m özvektör üzerine yapılan izdüşümlerin toplamıyla elde edilecek hatayı verir.

Örnek olarak ilinti katsayısı 0.95 olan özbağlanımlı bir rastgele \mathbf{x} sinyalin 3×3 boyutlu özilinti matrisini inceleyelim. Bahsedilen özilinti matrisi, sinyalin varyansı σ_x^2 olmak üzere şu şekilde ifade edilir:

$$\mathbf{R} = \sigma_x^2 \begin{bmatrix} 1 & 0.95 & 0.9025 \\ 0.95 & 1 & 0.95 \\ 0.9025 & 0.95 & 1 \end{bmatrix}. \quad (2.15)$$

\mathbf{R} matrisinin özdeğerleri $\lambda_1 = 0.034$, $\lambda_2 = 2.8685$ ve $\lambda_3 = 0.0975$ olup bunlara karşılık gelen özvektörler sırasıyla

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1.9669 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1.0169 \\ 1 \end{bmatrix} \text{ ve } \mathbf{v}_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad (2.16)$$

şeklinde bulunur. Taban vektörlerinin birimdik olması gerektiğinden bu vektörler normlarına bölünmelidir. Bölme işlemi sonunda şu vektörler elde edilir:

$$\mathbf{u}_1 = \begin{bmatrix} 0.4128 \\ -0.8119 \\ 0.4128 \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} 0.5741 \\ 0.5838 \\ 0.5741 \end{bmatrix} \text{ ve } \mathbf{u}_3 = \begin{bmatrix} -0.7071 \\ 0 \\ 0.7071 \end{bmatrix} \quad (2.17)$$

Özvektörler, karşılık gelen özdeğerlerine göre büyükten küçüğe sıralandığında KLT matrisi oluşturulur:

$$\mathbf{K} = \begin{bmatrix} 0.5741 & -0.7071 & 0.4128 \\ 0.5838 & 0 & -0.8119 \\ 0.5741 & 0.7071 & 0.4128 \end{bmatrix}. \quad (2.18)$$

İkinci bir örnek olarak ilinti katsayısı ρ olan özbağımlı bir rastgele \mathbf{x} sinyalinin 2×2 boyutlu özilinti matrisini inceleyelim. Özilinti matrisi

$$\mathbf{R} = \sigma_x^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \quad (2.19)$$

olduğundan özdeğerler

$$\lambda_1 = 1 + \rho, \quad \lambda_2 = 1 - \rho \quad (2.20)$$

ve bunlara karşılık gelen özvektörler, α ve β sıfırdan farklı herhangi iki reel sayı olmak üzere

$$\mathbf{v}_1 = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} \beta \\ -\beta \end{bmatrix} \quad (2.21)$$

şeklinde hesaplanır. Elde edilmeye çalışılan tabanın birimdik olması gerektiğinden $\alpha = \beta = 1/\sqrt{2}$ olmalıdır. $\rho \geq 0$ olduğundan özdeğerlere göre sıralanmış özvektörler, KLT matrisini

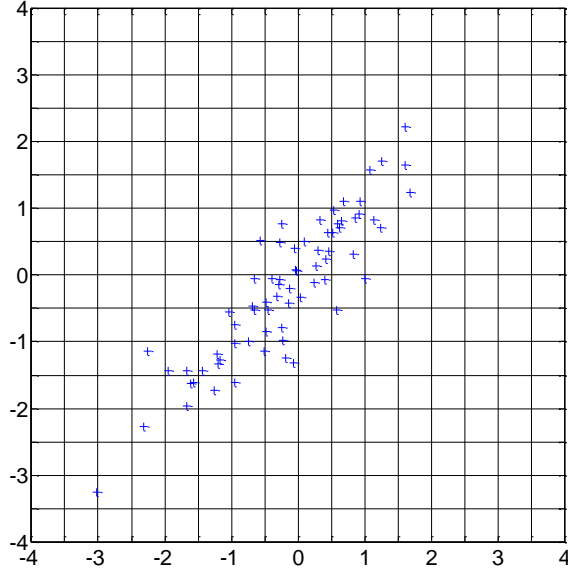
$$\mathbf{K} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.22)$$

şeklinde oluşturur. Bu, komşu elemanları arasında ilinti bulunan sinyalleri sıkıştırmak için kullanılacak optimum 2×2 boyutlu dönüşüm matrisidir. 2×2 boyutlu KLT, FFT, DCT, Haar ve Hadamard dönüşüm matrislerinin hepsi bu formdadır.

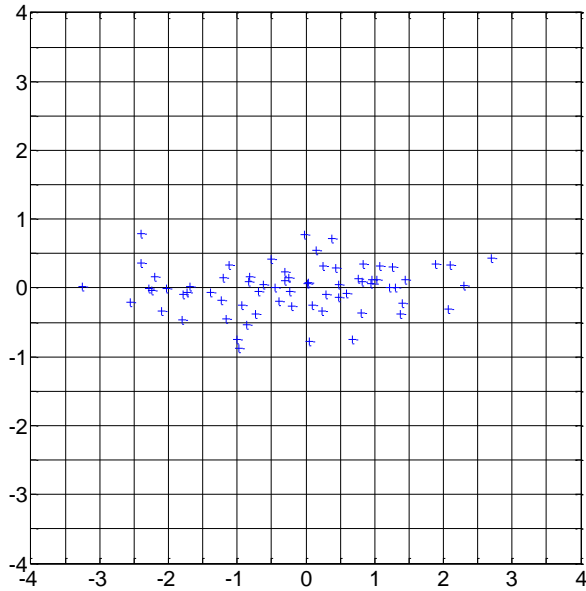
Birimdik vektörlerden oluşturulan matrislere *tekil matrisler* denir. Tekil matrislerin sütunları birbirlerine dik olduklarından tekil matrisler kullanılarak yapılan dönüşümler, *eksen döndürme* dönüşümleri olarak düşünülebilir. Örneğin (2.22)'de verilen dönüşüm, iki boyutlu Kartezyen düzlemde 45 derecelik bir eksen dönmesine karşılık gelmektedir.

\mathbf{x}_1 ve \mathbf{x}_2 vektörleri, dağılım grafikleri Şekil 2.1'deki gibi olan ve aralarında kuvvetli bir ilinti bulunan iki sinyale ait örnekleri gösterebiliriz. Bu vektörleri nicemlemek için 16'şar aralık kullandığımızı varsayalım. Bu durumda \mathbf{x}_1 ve \mathbf{x}_2 vektörleri 4'er bit kullanılarak kodlanabilir. Denklem 2.22'de verilen dönüşüm uygulandıktan sonra saçılım grafiği Şekil 2.2'deki gibi olacaktır. Dönüşümden sonra yatay ekseninde gösterilen sinyal 16 aralıklı, dikey ekseninde gösterilen sinyal

ise 4 aralıkla nicemlenebilir. Böylece dönüşüm sayesinde sinyaller toplam 8 bit yerine 6 bite kodlanabilirler.



Şekil 2.1. İlişkili iki sinyalin saçılım grafiği



Şekil 2.2. KLT uygulandıktan sonraki saçılım grafiği

2.1.2. Resimler Üzerinde KLT

Resimler, iki boyutlu sinyaller olarak düşünülebilirler. Önceki bölümde tek boyutlu sinyaller üzerinde KLT hesaplanmıştı. Resimler için bu dönüşümün iki boyuta genişletilmesi gerekir. Bu amaçla resmin satırları ve sütunları için ayrı ayrı özilinti matrisleri hesaplanır ve bu matrislerden üretilen KLT matrisleri kullanılarak resmin KLT dönüşümü hesaplanır.

$M \times N$ boyutlu bir \mathbf{A} resminin satırları $\mathbf{r}_i, i = 1, 2, \dots, M$; sütunları $\mathbf{c}_i, i = 1, 2, \dots, N$ vektörleriyle gösterilsin. Satır vektörleri $1 \times N$, sütun vektörleri $M \times 1$ boyutlu olacaktır. Satırlar ve sütunlar için ayrı ayrı özilinti matrisleri hesaplanıp ortalaması alındığında, resim için satır ve sütun özilinti matrisleri hesaplanmış olur. Yani resmin satırları için özilinti matrisi

$$\mathbf{R}_r = \frac{1}{M} \sum_{i=1}^M \mathbf{r}_i^T \mathbf{r}_i, \quad (2.23)$$

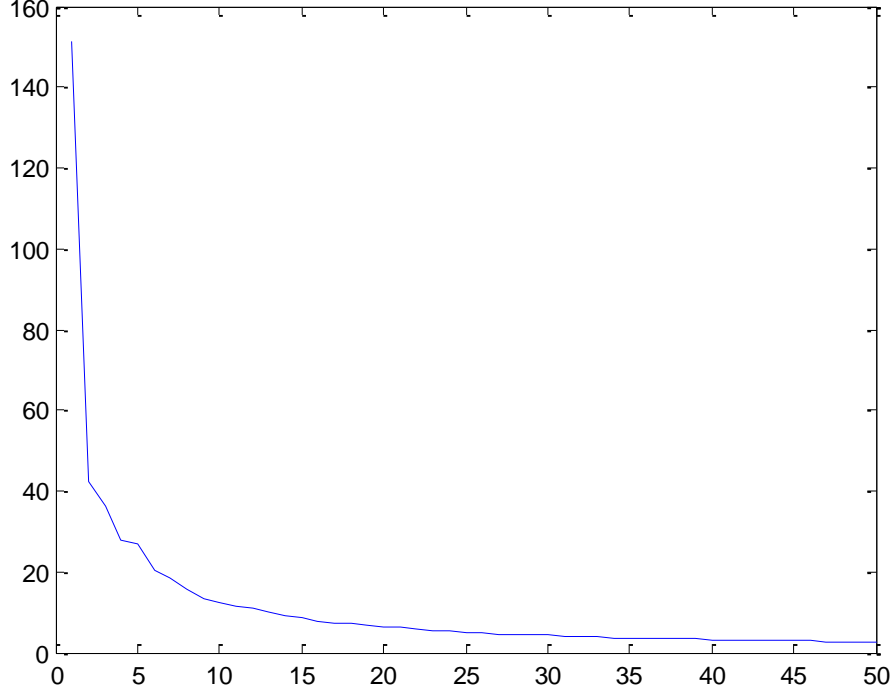
sütunları için özilinti matrisi ise

$$\mathbf{R}_c = \frac{1}{N} \sum_{i=1}^N \mathbf{c}_i \mathbf{c}_i^T \quad (2.24)$$

şeklinde hesaplanır. $N \times N$ boyutlu \mathbf{R}_r ve $M \times M$ boyutlu \mathbf{R}_c matrislerinin özdeğerleri ve özvektörleri hesaplanıp özvektörler özdeğerlere göre büyükten küçüğe sıralandığında elde edilen KLT matrisleri sırasıyla \mathbf{T}_r ve \mathbf{T}_c olsun. Satır dönüşüm matrisinin devriği soldan, sütun dönüşüm matrisi sağdan resme uygulandığında resimle aynı boyutta olan bir dönüşüm elde edilir:

$$\mathbf{B} = \mathbf{T}_r^T \mathbf{A} \mathbf{T}_c. \quad (2.25)$$

\mathbf{T}_r ve \mathbf{T}_c dönüşümleri \mathbf{A} resminin satırları ve sütunları üzerinde birer köşegenleme işlemi yaptıklarından \mathbf{B} matrisinin köşegenleri dışında kalan elemanları sıfır olacaktır. Satır ve sütun dönüşümlerinde özdeğerler büyükten küçüğe sıralandığından, \mathbf{B} matrisinin köşegendeki elemanları da mutlak değer bakımından büyükten küçüğe sıralanmış olacaktır. Lena resmi için elde edilen \mathbf{B} matrisinin ilk 50 köşegen elemanı Şekil 2.3'te bir grafik şeklinde gösterilmiştir.



Şekil 2.3. Lena resminin KLT'sinden elde edilen ilk 50 katsayı

\mathbf{B} dönüşümünden orijinal resmin elde edilmesi için satır ve sütun dönüşüm matrislerinin tekilliği kullanılabilir. Böylece orijinal resim,

$$\mathbf{A} = \mathbf{T}_r \mathbf{B} \mathbf{T}_c^T \quad (2.26)$$

ile geri çatlabilir.

\mathbf{B} matrisinin elemanları B_{ij} , \mathbf{T}_r ve \mathbf{T}_c matrislerinin sütunları sırasıyla $\mathbf{t}_r^i, i = 1, 2, \dots, M$ ve $\mathbf{t}_c^i, i = 1, 2, \dots, N$ olmak üzere (2.26)'da verilen matris çarpımı şu şekilde de yazılabilir:

$$\mathbf{A} = \sum_{i=1}^M \sum_{j=1}^N B_{ij} (\mathbf{t}_r^i) (\mathbf{t}_c^j)^T. \quad (2.27)$$

\mathbf{B} matrisi köşegen bir matris olduğundan (2.27)'deki toplam sadece köşegen elemanları üzerinden alınabilir:

$$\mathbf{A} = \sum_{i=1}^M B_{ii} (\mathbf{t}_r^i) (\mathbf{t}_c^i)^T. \quad (2.28)$$

Burada her B_{ii} katsayısının $(\mathbf{t}_r^i) (\mathbf{t}_c^i)^T$ matrisleriyle çarpılıp toplandığı görülmektedir. Yani satırlardan elde edilen \mathbf{T}_r matrisinin i 'inci satırı ile

sütunlardan elde edilen T_c matrisinin i 'inci satırının devriğinin çarpılması sonunda elde edilen *taban resimler*, bu taban resimlere karşılık gelen katsayılarla çarpılıp toplanmaktadır. Şekil 2.4'te Lena resmi ve Lena resminin ilk KLT taban resmi gösterilmektedir. İlk taban resminin, açık ve koyu bölgeler dikkate alındığında orijinal resme en çok benzeyen taban resmi olduğunu söyleyebiliriz. Orijinal resim içerisindeki parlaklık bilgisinin büyük bir kısmı bu taban resminde depolanmıştır.

B matrisinin köşegenlerindeki elemanların mutlak değerleri Şekil 2.3'teki gibi sürekli azaldığından, resim içerisindeki bilginin büyük çoğunluğunun ilk taban vektörlerinde tutulduğunu söyleyebiliriz. O halde dönüşüm katsayılarının hepsi yerine baştaki belirli bir kısmının kullanılmasıyla orijinal resmin bir yaklaşımı elde edilebilir. Şekil 2.5'te Lena resmi için **B** matrisinin köşegenindeki ilk 5, 15 ve 40 katsayı kullanılarak geri çatılan resimler gösterilmiştir. Görüldüğü üzere kullanılan katsayı sayısı arttıkça geri çatılan resim orijinal resme daha fazla benzemektedir. İlk kaç katsayının kullanılacağına karar vermek için, yaklaşımla orijinal resim arasındaki ortalama kare hata kullanılabilir. Denklem (2.14)'e göre bu hata miktarı, kullanılmayan katsayıların mutlak değerlerinin toplamı kadar olacaktır. Buna göre Lena resmindeki bilginin yaklaşık %20'si ilk katsayıda, %40'ı ilk 6 katsayıda ve %80'i ilk 70 katsayıda tutulabilmektedir.

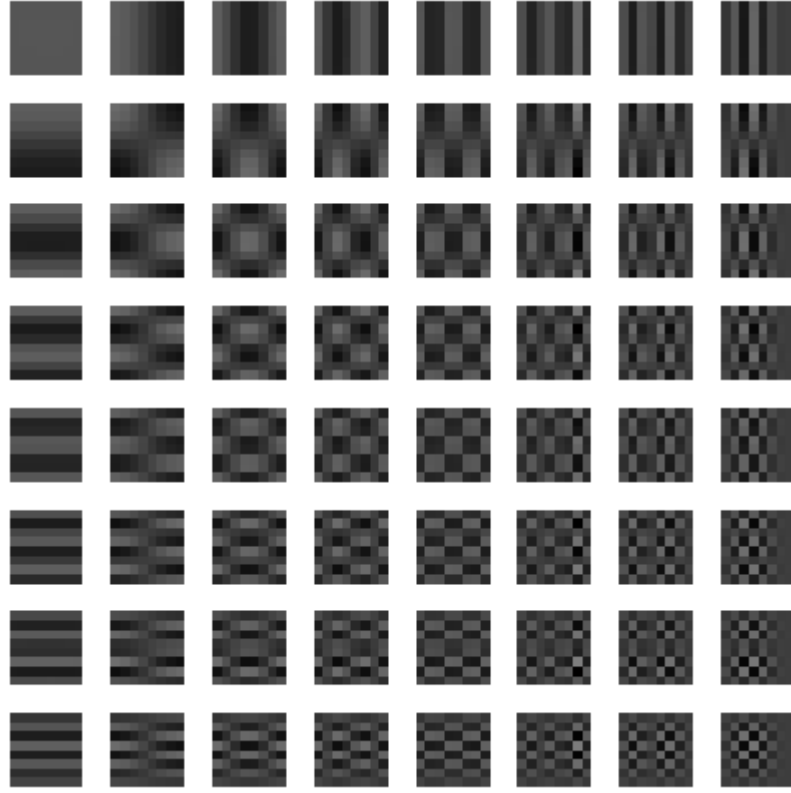


Şekil 2.4. Lena resmi ve Lena resminin ilk KLT taban resmi



Şekil 2.5. İlk KLT katsayıları kullanılarak geri çatılan Lena resimleri (a) Orijinal resim (b) 5 katsayı (c) 15 katsayı (d) 40 katsayı

Denklem (2.25)'te verilen dönüşüme göre, \mathbf{A} resmi $M \times N$ piksel değeri yerine M katsayıyla kodlanabilmektedir. Fakat kodlanan resmin geri çatılabilmesi için bu katsayıların yanında taban resimlerine de ihtiyaç vardır. Taban resimleri \mathbf{T}_r ve \mathbf{T}_c matrislerinden elde edilebiliyor olsa da, bu matrislerin büyüklükleri $M \times M$ ve $N \times N$ olduğundan toplamda orijinal resimden daha büyük olmaktadır. Bu durumda katsayılarla birlikte dönüşüm matrislerinin de kodlamaya dâhil edilmesi herhangi bir kodlama kazancı vermeyecektir. Fakat belirli bir uygulamada kullanılan resimlerin istatistiksel özellikleri birbirine yakınsa, önceden belirlenmiş optimum taban resimleri kullanılarak katsayılar belirlenebilir ve yüksek kodlama kazançları elde edilebilir.



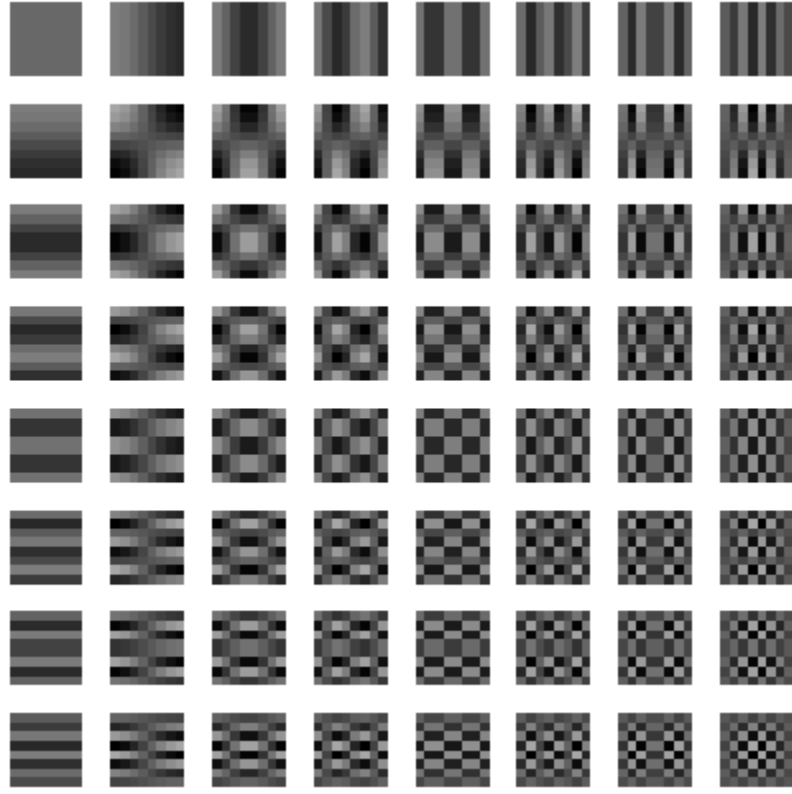
Şekil 2.6. Lena resmi için 8×8 boyutlu taban resimleri

Lena resminin tamamı yerine resmin 8×8'lik blokları üzerinden dönüşüm matrisi hesaplanabilir. Bu durumda dönüşüm matrisinin boyutu 8×8 olacaktır. Her bloğa ait katsayılar ayrı ayrı hesaplanarak resim kodlanabilir. Şekil 2.6'da Lena resminin 8×8'lik blokları kullanılarak hesaplanan taban resimleri gösterilmiştir.

Doğal resimlerde pikseller arasında kuvvetli bir ilinti olduğu göz önüne alındığında, bütün doğal resimlerde kullanılmak üzere Lena'dakine benzer, elemanları sabit bir taban resmi kümesi oluşturulabilir. Böylece dönüşüm matrisinin iletimine gerek kalmadan sadece taban resimlerine karşılık gelen katsayılar kullanılarak resimler kodlanabilir.

2.2. DCT

Ayrık kosinüs dönüşümü (DCT), ismini dönüşüm matrisini oluşturmakta kullanılan kosinüs fonksiyonlarından alır. $N \times N$ boyutlu DCT dönüşüm matrisinin (C) elemanları şu şekilde ifade edilir:



Şekil 2.7. 8×8 boyutlu DCT taban matrisleri

$$C_{i,j} = \begin{cases} \sqrt{1/N} \cos \frac{(2j+1)i\pi}{2N}, & i = 0, j = 0, 1, \dots, N-1 \\ \sqrt{2/N} \cos \frac{(2j+1)i\pi}{2N}, & i = 1, 2, \dots, N-1, j = 0, 1, \dots, N-1 \end{cases} \quad (2.29)$$

8 × 8 boyutlu DCT matrisinin satırlarının dış-çarpımları dönüşümün kullandığı taban matrislerini verir. Şekil 2.7’de gösterilen bu matrislerin, Şekil 2.6’deki Lena resmine ait KLT taban resimleriyle benzerlikleri kolayca görülmektedir. DFT’deki kompleks sayı hesaplamaları DCT’de olmadığı için DCT algoritması DFT’nikinden daha hızlı çalışır. Birçok veri kaynağı, yüksek ilinti katsayısına sahip Markov süreçleri şeklinde modellenebildiği için ve DCT bu tür kaynaklar üzerinde KLT’ye yakın sıkıştırma sağlar. Bu yüzden JPEG, MPEG, CCITT H.261 gibi birçok uluslararası sıkıştırma standardında DCT kullanılmaktadır [3].

3. BLOK DALGACIK DÖNÜŞÜMÜ

Blok dalgacık dönüşümü (BWT), sinyalin tamamı üzerinde değil, sinyalin küçük parçalara ayrılmasıyla elde edilen bloklar üzerinde dalgacık süzgeçleri yardımıyla uygulanan bir altbant ayrıştırma işlemidir. Bu işlem sonunda sinyal, çoklu-çözünürlükte ifade edilmiş olur. Bu yüzden, BWT'den önce dalgacık dönüşümü ve çoklu-çözünürlükten bahsetmek yerinde olacaktır.

3.1. Dalgacık Dönüşümü

Dalgacıklar üzerine ilk çalışmayı 20. yüzyılın başlarındaki çalışmasıyla Haar'a [22] atfetmek mümkündür. Haar bu çalışmasında *dalgacık* terimini kullanmamış olsa da, üzerinde çalıştığı fonksiyonlar kümesi kendisinden sonra *Haar dalgacık ailesinin* geliştirilmesine öncülük etmiştir. *Dalgacık* kelimesi ilk kez Morlet ve Grossmann'ın sismik sinyallerin analizi üzerine yaptıkları çalışmadan [23] sonra yayımladıkları ve dönüşümden sonra sinyalin kayıpsız geri çatılabildiğini gösterdikleri çalışmada [24] kullanılmıştır. Dalgacıkların yaygın kullanımı ise Mallat [25] ve Meyer [26] tarafından önerilen çoklu-çözünürlük analizi ile olmuştur. Bundan sonra araştırmacılar kendi dalgacık ailelerini matematiksel temellere dayalı olarak geliştirmeye başlamıştır. En popüler dalgacık ailesi Daubechies [27] tarafından ortaya konmuştur. Daubechies dalgacık ailesi diktir ve basit sayısal süzgeç teknikleriyle oluşturulabilir. Daubechies'den sonra atmosfer bilimi, ekonomi, biyoloji, elektrik ve makine mühendislikleri gibi birçok disiplinde dalgacık dönüşümleri üzerine araştırmalar yapılmıştır [28].

Burada dalgacık dönüşümünün Fourier dönüşümüne analogi oluşturacak şekilde kısa bir anlatımı sunulacaktır.

Bir $f(t)$ sinyali veya fonksiyonu, $f(t) = \sum_k a_k \psi_k(t)$ şeklinde, belli bazı fonksiyonların doğrusal kombinasyonu olarak ifade edildiğinde daha iyi analiz edilebilir ya da işlenebilir. Buradaki k , sonlu veya sonsuz toplamın tamsayı indeks değeri; a_k , reel değerli açılım katsayıları ve $\psi_k(t)$, t 'nin reel değerli bir fonksiyonlar kümesidir. Eğer bu açılım tekse, $\psi_k(t)$ fonksiyonları kümesine bir

taban denir. Fourier açılımında kullanılan taban fonksiyonları $\sin(k\omega_0 t)$ ve $\cos(k\omega_0 t)$ olmaktadır.

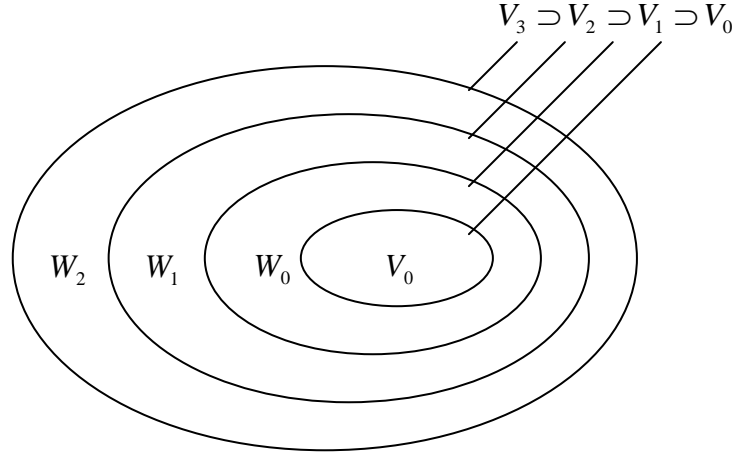
Dalgacık açılımında ise $f(t) = \sum_j \sum_k a_{j,k} \psi_{j,k}(t)$ şeklinde iki parametrelili bir sistem kullanılmaktadır. $a_{j,k}$ katsayılarının oluşturduğu kümeye *kesikli dalgacık dönüşümü* (*discrete wavelet transform – DWT*) denmektedir.

Dalgacık sistemi; bir sinyali oluşturmak veya ifade etmek için kullanılan yapı malzemeleri olarak düşünülebilir. Dalgacık açılımı, sinyalin zaman-frekans yerleşimini verir. Yani sinyalin enerjisinin büyük bir kısmı, birkaç $a_{j,k}$ katsayısıyla iyi bir şekilde ifade edilebilir. Ayrıca N uzunluğundaki sinyallerden katsayıların üretilmesi işlemi, birçok sistemler için $O(N)$; genel sistemler için $O(N \log N)$ mertebelerinde hızlı ve verimli bir şekilde yapılabilir [29].

İki parametrelili $\psi_{j,k}(t)$ ölçekleme fonksiyonu, tek bir ölçekleme fonksiyonundan basit *ölçekleme* ve *öteleme* işlemleriyle, yani $\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k)$ fonksiyonu ile elde edilebilir. Bunun yanında dalgacık sistemleri *çoklu-çözünürlük* (*multiresolution*) koşullarını da sağlarlar. Yani bir sinyaller kümesi, $\varphi(t - k)$ formunda fonksiyonlarla ifade edilebiliyorsa, daha büyük bir sinyaller kümesi, çözünürlüğü daha fazla olan $\varphi(2t - k)$ fonksiyonlarıyla ifade edilebilir. Ayrıca, düşük çözünürlüklü katsayılar, yüksek çözünürlüklü katsayılardan *süzgeç bankası* denilen ağaç yapılı bir algoritmayla türetilbilirler.

Dalgacık sistemlerinde iki fonksiyon kullanılmaktadır. Bunlar *ölçekleme fonksiyonu* ile *dalgacık fonksiyonudur*. Yukarıda bahsedilen $\varphi(t)$ fonksiyonunun tamsayı ötelenmeleriyle oluşturulan $\varphi_k(t) = \varphi(t - k)$ fonksiyonları, *ölçekleme fonksiyonlarıdır*.

Ölçekleme fonksiyonlarının gerdiği uzay V_0 olsun. 2 'nin üsleri ile ölçeklenmiş $\varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k)$ fonksiyonlarının gerdiği uzay da V_j olsun. Bu durumda $V_\infty = L^2$ (Kare integrallenebilir fonksiyonlar kümesi) ve $V_{-\infty} = \{0\}$ olur. Böylece çoklu-çözünürlük analizi için $\dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \subset L^2$ şeklinde iç-içe çözünürlük uzayları oluşturulmuş olur. Bu uzayların basit bir gösterimi şematik olarak Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Ölçekleme fonksiyonlarının gerdiği iç-içe uzaylar

V_{j+1} uzayında olup V_j uzayında bulunmayan altuzay W_j olsun. Çözünürlük uzayları arasında kalan bu W_j altuzaylarını geren ölçeklenmiş ve ötelenmiş $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$ fonksiyonlarına *dalgacık fonksiyonları* denmektedir.

Ölçekleme fonksiyonu $\varphi(t)$ ve dalgacık fonksiyonu $\psi(t)$; $\varphi(2t)$ fonksiyonunun ötelenmiş şekillerinin doğrusal kombinasyonu şeklinde şöyle yazılabilirler:

$$\begin{aligned} \varphi(t) &= \sum_n h_0(n)\sqrt{2}\varphi(2t - n) \\ \psi(t) &= \sum_n h_1(n)\sqrt{2}\varphi(2t - n) \end{aligned} \quad (3.1)$$

Buradaki $h_0(n)$ katsayılarına *ölçekleme fonksiyonu katsayıları*; $h_1(n)$ katsayılarına ise *dalgacık katsayıları* denir.

Bütün çözünürlük uzayları ve dalgacık uzayları birbirlerine diktirler. Bundan dolayı $h_0(n)$ ve $h_1(n)$ katsayıları arasında şu bağıntı yazılabilir [30]:

$$h_1(n) = \pm(-1)^n h_0(1 - n). \quad (3.2)$$

Sonlu N sayıda uzayın bulunduğu durum için ise bağıntı şu şekildedir:

$$h_1(n) = \pm(-1)^n h_0(N - 1 - n). \quad (3.3)$$

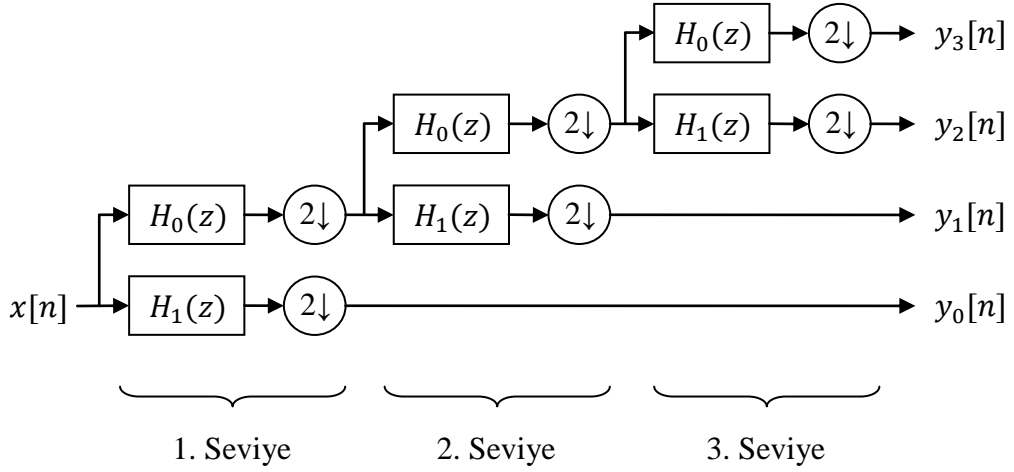
3.2. Çoklu Çözünürlük

Birçok uygulamada görüntünün veya videonun düşük çözünürlüklü şekli yeterli olurken bilgisayar görüşü ve görsel iletişim gibi uygulamalarda sinyal kalitesinin kabadan inceye doğru aşamalı bir şekilde iyileşmesi kullanışlı olabilmektedir. Örneğin, bazı web sayfalarına konan büyük boyutlu resimler web sayfası içerisinde yüklenmekte iken resmin önce düşük çözünürlükteki şekli ekranda görünmekte ve daha sonra resme ait diğer bilgiler geldikçe resim üzerindeki detaylar ortaya çıkmaktadır. Bu, resmin *çoklu-çözünürlük*le kodlanmış olmasından kaynaklanmaktadır.

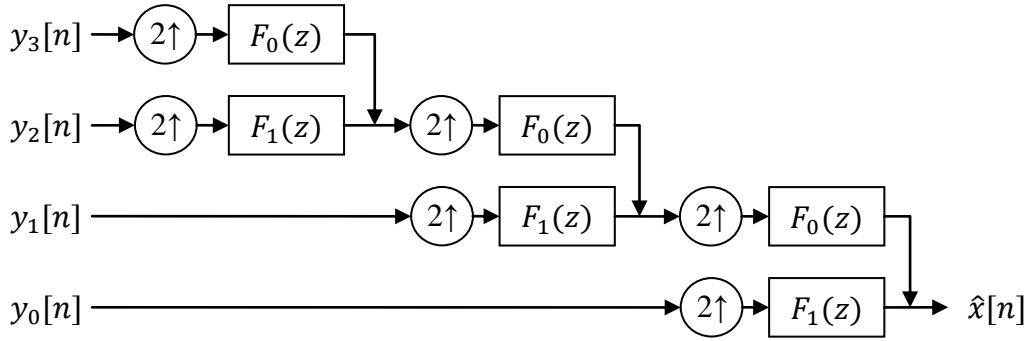
Çoklu-çözünürlüğü tek boyutlu bir sinyal üzerinde göstermek için Şekil 3.2'deki sistemi ele alalım.

Buradaki $H_0(z)$ ve $H_1(z)$, sırasıyla alçak geçiren ve yüksek geçiren süzgeçler olsunlar. Bir $x[n]$ sinyali bu süzgeçlerden geçirilip 2 ile aşağı örneklendiğinde $x[n]$ sinyaline ait *yaklaşım* ve *detay* sinyalleri elde edilir. Yaklaşım sinyali, şekil olarak $x[n]$ sinyaline benzer ve detay sinyaline oranla daha fazla bilgi içerir. Yaklaşım sinyali aynı süzgeçlerden tekrar geçirildiğinde yeni birer yaklaşım ve detay sinyalleri elde edilir. Yaklaşım sinyalleri bu şekilde belli bir seviyeye kadar süzgeçlerden geçirildikten sonra elde edilen sinyaller kodlanır. Şekil 3.2'de elde edilen son yaklaşım sinyali olan $y_3[n]$ kaliteli bir şekilde; detay sinyalleri olan $y_0[n]$, $y_1[n]$ ve $y_2[n]$ sinyalleri kaba bir şekilde kodlandığında yüksek kodlama kazançları elde edilebilir.

Şekil 3.2'de elde edilen $y_i[n]$, $i = 0, 1, 2, 3$ sinyallerinden $x[n]$ sinyalinin geri çatılması için Şekil 3.3'te verilen sistem kullanılabilir. H_0 , H_1 , F_0 ve F_1 süzgeçlerinin seçimine göre orijinal $x[n]$ sinyali ile geri çatılan $\hat{x}[n]$ sinyali arasında bazı farklılıklar olabilir. Eğer $x[n] = \hat{x}[n]$ koşulu sağlanıyorsa orijinal sinyalin *tam geri çatılmış* olduğu söylenir.



Şekil 3.2. 3 aşamalı çoklu-çözünürlük ağaç yapısı



Şekil 3.3. Çoklu-çözünürlüklü yaklaşım ve detay sinyallerinden orijinal sinyalin geri çatılması

3.3. QMF Süzgeç Bankaları

Şekil 3.2’de verilen sistemde orijinal sinyali altbantlara ayırmak için kullanılan H_0 ve H_1 süzgeçlerine *analiz süzgeçleri* adı verilir. Şekil 3.2’deki gibi bir sistemde altbant sinyallerinden orijinal sinyalin geri çatılması için kullanılan F_0 ve F_1 süzgeçlerine ise *sentez süzgeçleri* denir. Analiz ve sentez süzgeçlerinin hepsine birlikte *süzgeç bankası* denir.

Orijinal $x[n]$ ve geri çatılmış $\hat{x}[n]$ sinyalleri üç sebepten farklılık gösterebilir [31]:

- 1) Örtüşme
- 2) Genlik bozulumu
- 3) Faz bozulumu.

Dördüncü bir sebep olarak altbant sinyallerinin kodlanması ve nicemlenmesi ile meydana gelen farklılıklar sayılabilir. Fakat bu çalışmada altbant niceme hataları ihmal edilerek üzerinde durulmamıştır.

Analiz ve sentez süzgeçleri,

$$F_0(z) = H_1(-z), \quad F_1(z) = -H_0(-z) \quad (3.4)$$

şeklinde seçilirse örtüşme hataları giderilmiş olur.

$$\hat{X}(z) = cz^{-n_0}X(z) \text{ veya } \hat{x}(n) = cx(n - n_0), \quad c \neq 0 \quad (3.5)$$

koşulunu sağlayan; örtüşme, genlik bozulumu ve faz bozulumu hatalarından arındırılmış süzgeç bankalarının *tam geri-çatılma* koşulunu sağladıkları söylenir. Tam geri-çatılmada elde edilen çıktı, girdi sinyalinin ötelenmiş ve ölçeklenmiş şeklindedir.

İlk zamanlarda kullanılan süzgeç bankalarındaki süzgeçler,

$$H_1(z) = H_0(-z) \quad (3.6)$$

koşulunu sağlayacak şekilde seçiliyordu. Reel katsayılı durumda bu koşul $|H_1(e^{j\omega})| = |H_0(e^{j(\pi-\omega)})|$ şekline dönüşür. Bu da $|H_1(e^{j\omega})|$ fonksiyonunun $|H_0(e^{j\omega})|$ fonksiyonunun $2\pi/4$ (2π 'nin çeyrek açısı) frekansına göre ayna görüntüsü olması demek olduğundan bu tür süzgeç bankaları *dördün yansımali süzgeç bankası* (QMF bank = *quadrature mirror filter bank*) şeklinde isimlendirilmiştir.

Denklem (3.6)'daki seçimden sonra (3.4)'teki örtüşme hatalarını gideren koşul,

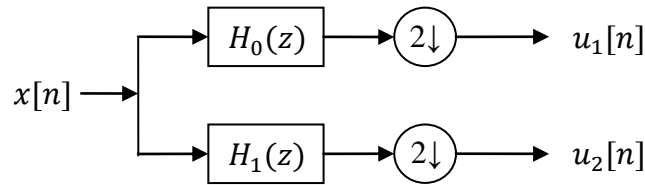
$$F_0(z) = H_0(z), \quad F_1(z) = -H_1(z) \quad (3.7)$$

şekline dönüşür. Buna göre analiz ve sentez için kullanılan dört süzgeç, sadece $H_0(z)$ süzgeci ile belirlenmiş olur. Böylece QMF bankası tasarlanırken tasarlayıcının sadece $H_0(z)$ süzgecini tasarlaması yeterli olur [31].

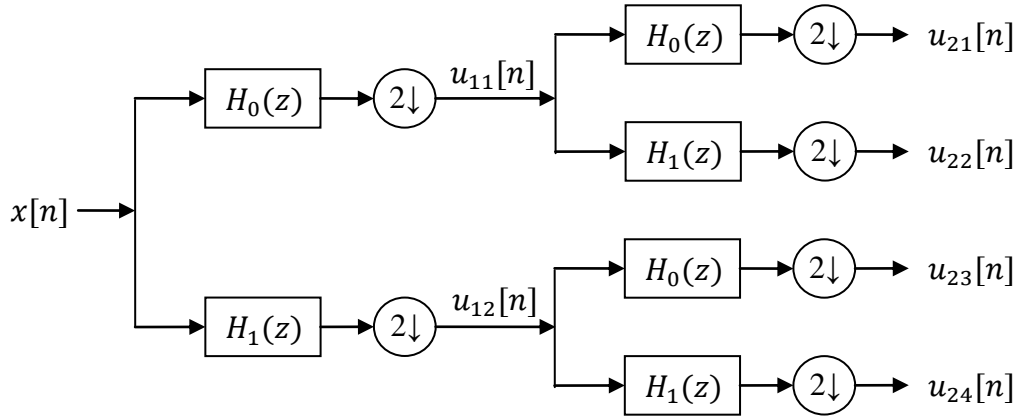
3.4. Blok Dalgacık Dönüşümü

DCT tabanlı kodlayıcılara oranla DWT tabanlı kodlayıcılar, geri çatılmış görüntülerde daha iyi görsel sonuçlar verirler ve bunun yanında DWT kodlayıcılar işaret-gürültü oranı ve çözünürlük üzerinde ölçeklenebilirlik özelliğini de sunarlar [8]. Fakat büyük boyutlu resimler üzerinde DWT kodlama, resimler genellikle bir bütün olarak ele alındığı için büyük bellek alanlarına ihtiyaç duyarlar. Gerekli bellek alanını azaltmak için dalgacık dönüşümünün yerel olarak uygulanması gerekir. Bu amaçla görüntüyü bloklara ayırarak her blok üzerinde dönüşüm uygulamak için *blok dalgacık dönüşümü* (BWT) adı verilen çeşitli yöntemler geliştirilmiştir. Çetin, Gerek ve Ulukuş [13] tarafından önerilen yöntemde dürtü tepkisi kullanılarak dönüşüm matrisi hesaplanmaktadır. Bu yöntemdeki blok sınırlarında oluşan bozulmaları ortadan kaldırmak için eksik pikselleri polinom dışdeğerlemesi ile hesaplayan bir yöntem Williams ve Amaratunga [32] tarafından önerilmiştir. Blok sınırlarındaki bozulmaları ortadan kaldırmak için Eom ve Kim [33], komşu bloklara ait verileri orijinal bloğun taşınan verileri olarak kullanan bir yöntem geliştirmiştir. JPEG-2000 standardında [34] *uzamsal bölüntülü dalgacık dönüşümü* (SSWT) kullanılmıştır. Chrysafis ve Ortega [35] gerekli bellek miktarını azaltmak için çizgi-tabanlı bir dönüşüm planı önermişlerdir.

Şekil 3.4'te tek aşamalı blok dalgacık sistemi gösterilmiştir. Burada $H_0(z)$ ve $H_1(z)$, tam geri çatımlı bir süzgeç bankasının sırasıyla alçak-geçiren ve yüksek-geçiren süzgeçleridir. $x[n]$ sinyali bu süzgeçlerden geçirilir ve iki ile aşağı örneklenerek $u_1[n]$ ve $u_2[n]$ altsinyalleri elde edilir. Bu altsinyaller,



Şekil 3.4. Tek aşamalı blok dalgacık sistemi



Şekil 3.5. İki aşamalı blok dalgacık sistemi

$$\begin{aligned}
 u_1[n] &= \sum_k h_0[n] x[2n - k] \\
 u_2[n] &= \sum_k h_1[n] x[2n - k]
 \end{aligned} \tag{3.8}$$

formülleriyle ifade edilirler. $u_1[n]$ sinyali, orijinal $x[n]$ sinyalinin alçak-geçiren bilgisinin içerirken, $u_2[n]$ sinyali yüksek-geçiren bilgisini içerir.

Sinyal işleme ve görüntü kodlama yöntemlerinde bu ayrıştırma işlemi altsinyaller üzerinde ağaca benzer bir yapıda tekrarlanır ve en sonda elde edilen altsinyaller değişik kodlama yöntemleriyle sıkıştırılır [36]. İki aşamalı blok dalgacık sistemi Şekil 3.5'te gösterilmiştir. Ağaç yapısı tekrar ettirilerek daha fazla aşamalı dalgacık sistemleri oluşturulabilir.

3.4.1. BWT Matrisi

Çetin, Gerek ve Ulukuş [13] tarafından önerilen blok dönüşüm yönteminde dürtü tepkisi kullanılarak bir dönüşüm matrisi oluşturulur ve resimdeki bloklar bu dönüşüm matrisiyle çarpılarak kanallara ayrılır ve kodlanır. Bu yöntemde ilk önce $N = 2^n$ kaplamalı ve N periyotlu bir $x[n] = [1 \ 0 \ 0 \ \dots \ 0]^T$ sinyali, n aşamalı bir blok dalgacık sistemine girdi sinyali olarak verilir. Sinyal her aşamada iki ile aşağı örneklendiği için son aşamada elde edilen her alt sinyalin periyodu 1

olur. Her altsinyalde tekrar eden elemanlar, dönüşüm matrisinin ilk sütununu oluşturur. Daha sonra $x[n]$ sinyali 1 birim sağa kaydırılır ve $[0 \ 1 \ 0 \ 0 \ \dots \ 0]^T$ sinyali elde edilir. Bu sinyalin dalgacık sistemine verilmesiyle dönüşüm matrisinin ikinci sütunu oluşturulur ve bu şekilde devam edilerek dönüşüm matrisi elde edilir.

3.4.2. BWT Matrisinin Tekilliği

Blok dalgacık sisteminde kullanılan $h_0(n)$ ve $h_1(n)$ süzgeç fonksiyonları ile bu süzgeçlerin zaman bölgesinde ikinin katı kadar ötelenmiş şekillerinin birleştirilmesiyle oluşan fonksiyonlar kümesi, birimdir bir kümedir [13]. Bu fonksiyonların dikliği Doğan ve Gerek [14] tarafından analitik olarak da gösterilmiştir. Takip eden bölümlerde tek aşamalı ve iki aşamalı BWT matrislerinin tekilliği gösterildikten sonra elde edilen sonuçlar daha fazla aşamalı BWT matrisleri için genellenecektir.

3.4.2.1. Tek Aşamalı BWT Matrisinin Tekilliği

Tekrar eden örnekleri a_0 ve a_1 reel sayıları olan 2 periyotlu bir $x[n]$ sinyali Şekil 3.4'te gösterilen BWT sistemine uygulansın ve $u_1[n]$ ve $u_2[n]$ altsinyalleri elde edilsin. Verilen koşullar altında $x[n]$ sinyali şu şekilde ifade edilir:

$$x[n] = \sum_r a_0 \delta[n - 2r] + \sum_r a_1 \delta[n - 2r + 1]. \quad (3.9)$$

$x[n]$ sinyalinin $h_0[n]$ ve $h_1[n]$ süzgeçlerinden geçirilip 2 ile aşağı örneklemeyle elde edilen $u_1[n]$ ve $u_2[n]$ altsinyalleri şu şekilde hesaplanır:

$$\begin{aligned} u_1[n] &= \sum_r \{a_0 h_0[2n - 2r] + a_1 h_0[2n - 2r + 1]\} \\ u_2[n] &= \sum_r \{a_0 h_1[2n - 2r] + a_1 h_1[2n - 2r + 1]\}. \end{aligned} \quad (3.10)$$

$u_1[n]$ ve $u_2[n]$ sinyallerinin periyodu 1 olduğundan bu sinyallerin bütün n anındaki değerleri, $n = 0$ anındaki değerlerine eşittir. Denklem (3.10)'da n yerine 0 yazılıp $r \rightarrow -r$ dönüşümü yapılırsa,

$$\begin{aligned}
u_1[n] &= \sum_r \{a_0 h_0[2r] + a_1 h_0[2r + 1]\} \\
u_2[n] &= \sum_r \{a_0 h_1[2r] + a_1 h_1[2r + 1]\}
\end{aligned} \tag{3.11}$$

bulunur. Tek aşamalı BWT matrisi, 2×2 boyutlu \mathbf{A}_2 matrisi olsun. \mathbf{A}_2 dönüşüm matrisinin ilk sütununu oluşturmak için sisteme a_0 yerine $a_0^{(1)} = 1$ ve a_1 yerine $a_1^{(1)} = 0$ değerleri konularak oluşturulan $e_1[n] = \sum_m \delta[n + 2m]$ girdi sinyali uygulanırsa, elde edilen altsinyal örnekleri (3.11)'den,

$$\begin{aligned}
u_1^{(1)} &= \sum_r h_0[2r] \\
u_2^{(1)} &= \sum_r h_1[2r]
\end{aligned} \tag{3.12}$$

olur. \mathbf{A}_2 dönüşüm matrisinin ikinci sütununu oluşturmak için sisteme a_0 yerine $a_0^{(2)} = 0$ ve a_1 yerine $a_1^{(2)} = 1$ değerleri konularak oluşturulan $e_2[n] = \sum_m \delta[n - 1 + 2m]$ girdi sinyali uygulanmalıdır. Bu durumda altsinyal örnekleri,

$$\begin{aligned}
u_1^{(2)} &= \sum_r h_0[2r + 1] \\
u_2^{(2)} &= \sum_r h_1[2r + 1]
\end{aligned} \tag{3.13}$$

şeklinde elde edilir. Denklem (3.12) ve (3.13)'teki değerler, vektör haline getirilip yan yana yazıldığında \mathbf{A}_2 dönüşüm matrisi elde edilir:

$$\mathbf{A}_2 = \begin{bmatrix} \sum_r h_0[2r] & \sum_r h_0[2r + 1] \\ \sum_r h_1[2r] & \sum_r h_1[2r + 1] \end{bmatrix}. \tag{3.14}$$

\mathbf{A}_2 matrisinin bir özelliği, $[a_0 \ a_1]^T$ vektörüyle çarpıldığında, elemanları (3.11)'de verilen $[u_1[n] \ u_2[n]]^T$ vektörünü vermesidir:

$$\mathbf{A}_2 \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \sum_r h_0[2r] + a_1 \sum_r h_0[2r + 1] \\ a_0 \sum_r h_1[2r] + a_1 \sum_r h_1[2r + 1] \end{bmatrix} = \begin{bmatrix} u_1[n] \\ u_2[n] \end{bmatrix}. \tag{3.15}$$

$\mathbf{A}_2^T \mathbf{A}_2$ matrisini \mathbf{C} ile gösterelim ve \mathbf{C} matrisinin elemanları c_{ij} , $i, j = 1, 2$ olsun. Bu elemanlar şu şekilde hesaplanır:

$$\begin{aligned}
c_{11} &= \sum_{r,r'} \{h_0[2r]h_0[2r'] + h_1[2r]h_1[2r']\} \\
c_{12} &= \sum_{r,r'} \{h_0[2r]h_0[2r' + 1] + h_1[2r]h_1[2r' + 1]\} \\
c_{21} &= \sum_{r,r'} \{h_0[2r + 1]h_0[2r'] + h_1[2r + 1]h_1[2r']\} \\
c_{22} &= \sum_{r,r'} \{h_0[2r + 1]h_0[2r' + 1] + h_1[2r + 1]h_1[2r' + 1]\}.
\end{aligned} \tag{3.16}$$

h_0 ve h_1 süzgeçleri ile bunların ikinin katları kadar ötelenmiş şekilleri birimdik bir küme oluşturduğundan şu koşul yazılabilir:

$$\sum_r \{h_0[m - 2r]h_0[n - 2r] + h_1[m - 2r]h_1[n - 2r]\} = \delta_{m,n} \tag{3.17}$$

Denklem (3.16)'da verilen değerler üzerinde $r' \rightarrow k = r' - r$ ve ardından $r \rightarrow -r$ dönüşümleri yapılırsa,

$$\begin{aligned}
c_{11} &= \sum_{r,k} \{h_0[-2r]h_0[2k - 2r] + h_1[-2r]h_1[2k - 2r]\} \\
c_{12} &= \sum_{r,k} \{h_0[-2r]h_0[2k + 1 - 2r] + h_1[-2r]h_1[2k + 1 - 2r]\} \\
c_{21} &= \sum_{r,k} \{h_0[1 - 2r]h_0[2k - 2r] + h_1[1 - 2r]h_1[2k - 2r]\} \\
c_{22} &= \sum_{r,k} \{h_0[1 - 2r]h_0[2k + 1 - 2r] + h_1[1 - 2r]h_1[2k + 1 - 2r]\}
\end{aligned} \tag{3.18}$$

bulunur. Denklem (3.17)'de verilen diklik koşulu burada uygulanırsa,

$$\begin{aligned}
c_{11} &= \sum_k \delta_{0,2k} = 1 \\
c_{12} &= \sum_k \delta_{0,2k+1} = 0 \\
c_{21} &= \sum_k \delta_{1,2k} = 0 \\
c_{22} &= \sum_k \delta_{1,2k+1} = 1
\end{aligned} \tag{3.19}$$

hesaplanır. Bu denklemdeki elemanlar, $\mathbf{C} = \mathbf{A}_2^T \mathbf{A}_2$ matrisindeki yerlerine yazılırsa,

$$\mathbf{A}_2^T \mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.20}$$

olduğu; yani \mathbf{A}_2 dönüşüm matrisinin tekil olduğu gösterilmiş olur. \square

3.4.2.2. İki Aşamalı BWT Matrisinin Tekilliği

Şekil 3.5'te gösterilen iki aşamalı dalgacık sistemine girdi olarak, tekrar eden örnekleri a_0, a_1, a_2 ve a_3 olan 4 periyotlu $x[n]$ sinyalinin uygulandığını düşünelim. $x[n]$ sinyali, Kronecker fonksiyonu yardımıyla şu şekilde yazılabilir:

$$x[n] = \sum_r \sum_{s=0}^3 a_s \delta[n - 4r + s]. \quad (3.21)$$

Bu sinyalin h_0 ve h_1 süzgeçlerinden geçirilip 2 ile aşağı örneklemeyle elde edilen u_{11} ve u_{12} sinyalleri şu şekilde hesaplanır:

$$u_{11}[n] = \sum_r \sum_{s=0}^3 a_s h_0[2n - 4r + s] \quad (3.22)$$
$$u_{12}[n] = \sum_r \sum_{s=0}^3 a_s h_1[2n - 4r + s].$$

$x[n]$ sinyalinin periyodu 4 olduğundan 2 ile aşağı örnekleme sebebiyle u_{11} ve u_{12} sinyallerinin periyotları 2 olur. u_{11} sinyalinin çift zamanlı örneklerini b_0 , tek zamanlı örneklerini b_1 ile; aynı şekilde u_{12} sinyalinin çift zamanlı örneklerini b_2 , tek zamanlı örneklerini b_3 ile gösterelim. Böylece u_{11} ve u_{12} sinyalleri, (3.9)'da verilen girdi sinyaline benzer şekilde ifade edilmiş olur:

$$u_{11}[n] = \sum_r b_0 \delta[n - 2r] + \sum_r b_1 \delta[n - 2r + 1] \quad (3.23)$$
$$u_{12}[n] = \sum_r b_2 \delta[n - 2r] + \sum_r b_3 \delta[n - 2r + 1].$$

Tanım gereği $b_i, i = 0, 1, 2, 3$ değerleri u_{11} ve u_{12} sinyallerinin tek veya çift zamanlı örnekleri olduklarından şu denklemler yazılabilir:

$$b_0 = u_{11}[0] = \sum_r \sum_{s=0}^3 a_s h_0[-4r + s]$$
$$b_1 = u_{11}[1] = \sum_r \sum_{s=0}^3 a_s h_0[2 - 4r + s] \quad (3.24)$$
$$b_2 = u_{12}[0] = \sum_r \sum_{s=0}^3 a_s h_1[-4r + s]$$
$$b_3 = u_{12}[1] = \sum_r \sum_{s=0}^3 a_s h_1[2 - 4r + s].$$

Bu denklemler üzerinde $r \rightarrow -r$ dönüşümünün ardından r üzerinden uygun öteleme işlemleri yapıldıktan sonra şu denklemler bulunur:

$$\begin{aligned}
b_0 &= \sum_r \{a_0 h_0[4r] + a_1 h_0[4r + 1] + a_2 h_0[4r + 2] \\
&\quad + a_3 h_0[4r + 3]\} \\
b_1 &= \sum_r \{a_0 h_0[4r + 2] + a_1 h_0[4r + 3] + a_2 h_0[4r] \\
&\quad + a_3 h_0[4r + 1]\} \\
b_2 &= \sum_r \{a_0 h_1[4r] + a_1 h_1[4r + 1] + a_2 h_1[4r + 2] \\
&\quad + a_3 h_1[4r + 3]\} \\
b_3 &= \sum_r \{a_0 h_1[4r + 2] + a_1 h_1[4r + 3] + a_2 h_1[4r] \\
&\quad + a_3 h_1[4r + 1]\}.
\end{aligned} \tag{3.25}$$

Denklem (3.23)'te verilen u_{11} ve u_{12} sinyalleri h_0 ve h_1 süzgeçlerinden geçirilip 2 ile aşağı örneklendikten sonra u_{21} , u_{22} , u_{23} ve u_{24} altsinyalleri şu şekilde hesaplanır:

$$\begin{aligned}
u_{21}[n] &= \sum_r \{b_0 h_0[2n - 2r] + b_1 h_0[2n - 2r + 1]\} \\
u_{22}[n] &= \sum_r \{b_0 h_1[2n - 2r] + b_1 h_1[2n - 2r + 1]\} \\
u_{23}[n] &= \sum_r \{b_2 h_0[2n - 2r] + b_3 h_0[2n - 2r + 1]\} \\
u_{24}[n] &= \sum_r \{b_2 h_1[2n - 2r] + b_3 h_1[2n - 2r + 1]\}.
\end{aligned} \tag{3.26}$$

Bu sinyaller, 2-periyotlu sinyallerin 2 ile aşağı örneklemeyle oluşturuldukları için 1-periyodiktir. Yani altsinyallerin bütün terimleri, $n = 0$ anındaki terime eşittir. Denklem (3.26)'da n yerine 0 yazılıp $r \rightarrow -r$ dönüşümü yapılırsa şu denklemler elde edilir:

$$\begin{aligned}
u_{21}[n] &= \sum_r \{b_0 h_0[2r] + b_1 h_0[2r + 1]\} \\
u_{22}[n] &= \sum_r \{b_0 h_1[2r] + b_1 h_1[2r + 1]\} \\
u_{23}[n] &= \sum_r \{b_2 h_0[2r] + b_3 h_0[2r + 1]\}
\end{aligned} \tag{3.27}$$

$$u_{24}[n] = \sum_r \{b_2 h_1[2r] + b_3 h_1[2r + 1]\}.$$

Bir periyot içerisindeki örnekleri $a_0^{(1)}$, $a_1^{(1)}$, $a_2^{(1)}$ ve $a_3^{(1)}$ olan bir $e^{(1)}[n]$ sinyali, iki aşamalı dalgacık sistemine verildiğinde, (3.25) ile hesaplanan katsayılar $b_0^{(1)}$, $b_1^{(1)}$, $b_2^{(1)}$ ve $b_3^{(1)}$; elemanları (3.27) ile hesaplanan çıktı vektörü de $\mathbf{u}^{(1)} = [u_{21}^{(1)} \ u_{22}^{(1)} \ u_{23}^{(1)} \ u_{24}^{(1)}]^T$ olsun. Aynı şekilde, bir periyot içerisindeki örnekleri $a_0^{(2)}$, $a_1^{(2)}$, $a_2^{(2)}$ ve $a_3^{(2)}$ olan bir $e^{(2)}[n]$ sinyalinin (3.25) ile ürettiği katsayılar $b_0^{(2)}$, $b_1^{(2)}$, $b_2^{(2)}$ ve $b_3^{(2)}$; (3.27) ile ürettiği çıktı vektörü $\mathbf{u}^{(2)} = [u_{21}^{(2)} \ u_{22}^{(2)} \ u_{23}^{(2)} \ u_{24}^{(2)}]^T$ olsun.

$\mathbf{u}^{(1)}$ ve $\mathbf{u}^{(2)}$ vektörlerinin iç-çarpımları şu şekilde hesaplanır:

$$\begin{aligned} \langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle &= (\mathbf{u}^{(1)})^T \mathbf{u}^{(2)} \\ &= u_{21}^{(1)} u_{21}^{(2)} + u_{22}^{(1)} u_{22}^{(2)} + u_{23}^{(1)} u_{23}^{(2)} + u_{24}^{(1)} u_{24}^{(2)}. \end{aligned} \quad (3.28)$$

Denklem (3.27)'deki değerler $e^{(1)}[n]$ ve $e^{(2)}[n]$ için hesaplanıp (3.28)'de yerine yazıldığında,

$$\begin{aligned} \langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle &= b_0^{(1)} b_0^{(2)} \sum_{r,r'} (h_0[2r] h_0[2r'] + h_1[2r] h_1[2r']) \\ &\quad + b_0^{(1)} b_1^{(2)} \sum_{r,r'} (h_0[2r] h_0[2r' + 1] + h_1[2r] h_1[2r' + 1]) \\ &\quad + b_1^{(1)} b_0^{(2)} \sum_{r,r'} (h_0[2r + 1] h_0[2r'] + h_1[2r + 1] h_1[2r']) \\ &\quad + b_1^{(1)} b_1^{(2)} \sum_{r,r'} (h_0[2r + 1] h_0[2r' + 1] + h_1[2r + 1] h_1[2r' + 1]) \\ &\quad + b_2^{(1)} b_2^{(2)} \sum_{r,r'} (h_0[2r] h_0[2r'] + h_1[2r] h_1[2r']) \\ &\quad + b_2^{(1)} b_3^{(2)} \sum_{r,r'} (h_0[2r] h_0[2r' + 1] + h_1[2r] h_1[2r' + 1]) \\ &\quad + b_3^{(1)} b_2^{(2)} \sum_{r,r'} (h_0[2r + 1] h_0[2r'] + h_1[2r + 1] h_1[2r']) \\ &\quad + b_3^{(1)} b_3^{(2)} \sum_{r,r'} (h_0[2r + 1] h_0[2r' + 1] + h_1[2r + 1] h_1[2r' + 1]) \end{aligned} \quad (3.29)$$

elde edilir. Burada $r' \rightarrow k = r' - r$ ve ardından $r \rightarrow -r$ dönüşümleri yapıldıktan sonra (3.17)'de verilen diklik koşulunun uygulandığında,

$$\begin{aligned}
\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle &= b_0^{(1)} b_0^{(2)} \sum_k \delta_{0,2k} + b_0^{(1)} b_1^{(2)} \sum_k \delta_{0,2k+1} \\
&+ b_1^{(1)} b_0^{(2)} \sum_k \delta_{1,2k} + b_1^{(1)} b_1^{(2)} \sum_k \delta_{1,2k+1} \\
&+ b_2^{(1)} b_2^{(2)} \sum_k \delta_{0,2k} + b_2^{(1)} b_3^{(2)} \sum_k \delta_{0,2k+1} \\
&+ b_3^{(1)} b_2^{(2)} \sum_k \delta_{1,2k} + b_3^{(1)} b_3^{(2)} \sum_k \delta_{1,2k+1}
\end{aligned} \tag{3.30}$$

bulunur. Kronecker fonksiyonunun özelliğinden, denklemdaki toplam ifadeleri 0 ya da 1 değerini alır ve

$$\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle = b_0^{(1)} b_0^{(2)} + b_1^{(1)} b_1^{(2)} + b_2^{(1)} b_2^{(2)} + b_3^{(1)} b_3^{(2)} \tag{3.31}$$

elde edilir. Denklem sağı tarafındaki ifade, ilk aşama sonunda elde edilen b_i değerleri kullanılarak iki durum için oluşturulan vektörlerin iç-çarpımı demektir. Yani, $\mathbf{b}^{(1)} = [b_0^{(1)} \quad b_1^{(1)} \quad b_2^{(1)} \quad b_3^{(1)}]^T$ ve $\mathbf{b}^{(2)} = [b_0^{(2)} \quad b_1^{(2)} \quad b_2^{(2)} \quad b_3^{(2)}]^T$ olmak üzere,

$$\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle = \langle \mathbf{b}^{(1)}, \mathbf{b}^{(2)} \rangle \tag{3.32}$$

olur.

Denklem (3.31)'de verilen değerlerin $a_i^{(j)}, i = 0, 1, 2, 3; j = 1, 2$ terimleri cinsinden hesaplanabilmesi için, (3.25)'te verilen $b_i^{(j)}, i = 0, 1, 2, 3; j = 1, 2$ değerlerinin (3.31)'de yerlerine yazılması gerekir. Burada örnek olarak, yalnızca elde edilecek ifadedeki $a_0^{(1)} a_0^{(2)}$ ve $a_0^{(1)} a_1^{(2)}$ 'li terimlerin katsayıları incelenecektir. Bu terimlerin katsayılarından elde edilecek sonuç, diğer terimlerin katsayıları için genelleştirilecektir. $a_0^{(1)} a_0^{(2)}$ 'li terimin katsayısı, c_{00} , şu şekilde bulunur:

$$\begin{aligned}
c_{00} &= \sum_{r,r'} \{h_0[4r]h_0[4r'] + h_0[4r+2]h_0[4r'+2] \\
&+ h_1[4r]h_1[4r'] + h_1[4r+2]h_1[4r'+2]\}.
\end{aligned} \tag{3.33}$$

Burada yine $r' \rightarrow k = r' - r$ ve ardından $r \rightarrow -r$ dönüşümleri yapılırsa,

$$\begin{aligned}
c_{00} &= \sum_{r,k} \{(h_0[-4r]h_0[4k-4r] \\
&+ h_0[2-4r]h_0[2+4k-4r]) \\
&+ (h_1[-4r]h_1[4k-4r] \\
&+ h_1[2-4r]h_1[2+4k-4r])\}
\end{aligned} \tag{3.34}$$

bulunur. Parantezler içerisinde gruplandırılan toplamlara bakıldığında, h_0 ve h_1 süzgeçlerinin mod 4'e göre 0 ve 2'ye karşılık gelen terimlerinin çarpılıp toplandığı görülür. Bu da mod 2'ye göre 0'a karşılık gelen terimler demektir. Buna göre (3.34) şu şekilde yazılabilir:

$$c_{00} = \sum_{r,k} \{h_0[-2r]h_0[4k-2r] + h_1[-2r]h_1[4k-2r]\} \quad (3.35)$$

Denklem (3.17)'deki diklik koşulu uygulandığında,

$$c_{00} = \sum_k \delta_{0,4k} = 1 \quad (3.36)$$

bulunur.

$a_0^{(1)}a_1^{(2)}$ 'li terimin katsayısı, c_{01} , ise şu şekilde hesaplanır:

$$\begin{aligned} c_{01} = \sum_{r,r'} \{ & h_0[4r]h_0[4r'+1] + h_0[4r+2]h_0[4r'+3] \\ & + h_1[4r]h_1[4r'+1] \\ & + h_1[4r+2]h_1[4r'+3]\}. \end{aligned} \quad (3.37)$$

Burada da $r' \rightarrow k = r' - r$ ve ardından $r \rightarrow -r$ dönüşümleri yapılırsa,

$$\begin{aligned} c_{01} = \sum_{r,k} \{ & h_0[-4r]h_0[4k+1-4r] \\ & + h_0[2-4r]h_0[4k+3-4r] \\ & + h_1[-4r]h_1[4k+1-4r] \\ & + h_1[2-4r]h_1[4k+3-4r]\} \end{aligned} \quad (3.38)$$

bulunur. Mod 4'e göre 1 ve 3 numaralı indisler, mod 2'ye göre 1'e; 0 ve 2 numaralı indisler 0'a karşılık geldiğinden,

$$\begin{aligned} c_{01} = \sum_{r,k} \{ & h_0[-2r]h_0[4k+1-2r] \\ & + h_1[-2r]h_1[4k+1-2r]\} \end{aligned} \quad (3.39)$$

yazılabilir. Denklem (3.17)'deki diklik koşulu uygulandığında,

$$c_{01} = \sum_k \delta_{0,4k+1} = 0 \quad (3.40)$$

bulunur.

Benzer şekilde diğer $a_i^{(1)}a_j^{(2)}$, $i, j = 0, 1, 2, 3$ terimlerinin katsayıları, c_{ij} , hesaplanırsa şu sonuca varılır:

$$c_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad i, j = 0, 1, 2, 3. \quad (3.41)$$

Bu durumda, (3.31)'deki $\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle$ terimi şöyle bulunur:

$$\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle = \sum_{i=0}^3 \sum_{j=0}^3 c_{ij} a_i^{(1)} a_j^{(2)} = \sum_{i=0}^3 a_i^{(1)} a_i^{(2)}. \quad (3.42)$$

Bu denklemin sağ tarafı, $\mathbf{a}^{(1)} = [a_0^{(1)} \ a_1^{(1)} \ a_2^{(1)} \ a_3^{(1)}]^T$ ve $\mathbf{a}^{(2)} = [a_0^{(2)} \ a_1^{(2)} \ a_2^{(2)} \ a_3^{(2)}]^T$ vektörlerinin iç-çarpımı demektir. Denklem (3.32) ile (3.42) birleştirildiğinde şu sonuç elde edilir:

$$\langle \mathbf{u}^{(1)}, \mathbf{u}^{(2)} \rangle = \langle \mathbf{b}^{(1)}, \mathbf{b}^{(2)} \rangle = \langle \mathbf{a}^{(1)}, \mathbf{a}^{(2)} \rangle. \quad (3.43)$$

BWT matrisini oluşturmak için, periyot içerisindeki elemanları $\mathbf{a}^{(1)} = [1 \ 0 \ 0 \ 0]^T$ vektörü ile ifade edilen sinyalin ve bu sinyalin örneklerinin birer sağa kaydırılmasıyla oluşturulan sinyallerin dalgacık sistemine verilmesi gerekir. Bu sinyallerin bir periyot içerisinde kalan terimleriyle $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$, $\mathbf{a}^{(3)}$ ve $\mathbf{a}^{(4)}$ vektörleri oluşturulsun. Her sinyal için dalgacık sisteminin verdiği yanıtlar $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, $\mathbf{u}^{(3)}$ ve $\mathbf{u}^{(4)}$ vektörlerini oluştursun. $\mathbf{u}^{(i)}$, $i = 0, 1, 2, 3$ vektörlerinin yan yana dizilmesiyle \mathbf{A}_4 dönüşüm matrisi oluşturulur:

$$\mathbf{A}_4 = \begin{bmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \mathbf{u}^{(1)} & \mathbf{u}^{(2)} & \mathbf{u}^{(3)} & \mathbf{u}^{(4)} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix}. \quad (3.44)$$

$\mathbf{a}^{(i)}$, $i = 0, 1, 2, 3$ vektörleri birimdik bir küme oluşturduğundan ve (3.43)'ten $\mathbf{u}^{(i)}$, $i = 0, 1, 2, 3$ vektörleri de birimdik bir küme oluştururlar. O halde $\mathbf{A}_4^T \mathbf{A}_4$ matrisi birim matris olur. Bu da \mathbf{A}_4 dönüşüm matrisinin tekil olduğu manasına gelir. □

3.4.2.3. İki Aşamalı Fazla Aşamalı BWT Matrisinin Tekilliği

İki aşamalı BWT matrisinin tekilliğini göstermek için yapılan işlemlerden görüldüğü üzere, her aşamada karşılaşılan denklemler, bir düşük aşamadakinden daha karmaşık ve daha fazla sayıda olmaktadır. Fakat (3.32) ve (3.43)'teki iç-çarpımlardaki benzerlikler dikkat çekmektedir. Bu denklemlerin genellenmesiyle ikiden fazla aşamalı BWT matrisinin de tekil olduğu söylenebilir.

İkiden fazla aşamalı dalgacık sistemlerinde, altbantlardaki terimler $\mathbf{u}^{(i)}$ vektörleriyle gösterilsin. Bu vektörlerin iç-çarpımları, (3.31)'deki gibi bir önceki seviyede elde edilen sinyallerin örnekleriyle ifade edilebilir. Her seviyedeki örnekler, bir önceki seviyedeki örnekler cinsinden yazıldıktan ve katsayılar (3.33) ve (3.37)'deki gibi hesaplandıktan sonra $r' \rightarrow k = r' - r$ ve $r \rightarrow -r$ dönüşümleri uygulanırsa, (3.34) ve (3.38)'dekine benzer denklemler elde edilir. Bu denklemlerde çift ve tek indisli terimler bir araya toplanıp (3.17)'deki diklik koşulu uygulandığında terimlerin katsayılarının ya 0 ya da 1 olduğu görülür. Böylece bir seviyede elde edilen vektörlerin iç çarpımlarının, bir önceki seviyede bulunan vektörlerin iç çarpımlarına (3.43)'teki gibi eşit oldukları bulunur. Sonuç olarak ikiden fazla aşamalı dalgacık sistemlerinde BWT matrisinin sütunlarının birimlik bir küme oluşturduğu görülür ve BWT matrisinin tekil olduğu söylenir. \square

Denklem (3.43)'e göre $x[n]$ girdi sinyalinin değerlerinin yalnız 1 ve 0'lardan oluşması gerekmez. BWT matrisinin oluşturulması için sisteme verilmesi gereken sinyallerin periyot içerisindeki elemanları ile oluşturulan vektörlerin birimlik bir küme oluşturması yeterlidir. Fakat elemanlar 1 ve 0'lardan seçilirse işlemler daha basit bir şekilde girer ve pratikte kullanımı daha kolay olur.

4. QMF SÜZGEÇ BANKASI TASARIMI

Bölüm 2.1.1’de KLT matrisini oluşturan vektörlerin, özilinti matrisinin özvektörleri olduğundan bahsedilmişti. Tanım gereği bu özvektörler birimlik bir küme oluştururlar. Dolayısıyla KLT matrisi tekildir. Bölüm 3.2.2’de de BWT matrisinin tekiliğinden bahsedilmişti. Hem KLT, hem de BWT matrislerinin tekil olmasından hareketle, KLT matrisini üretecek ve KLT’nin optimumluğunun avantajını kullanacak şekilde BWT süzgeç bankalarının tasarlanabileceği düşünülmektedir.

Bu amaçla tasarlanacak süzgeçler, (3.17)’de verilen diklik koşulunu sağlamalıdır. KLT matrisini üreten BWT süzgeçleri değişik formlarda olabilir. Fakat bu çalışmada ağırlıklı olarak QMF süzgeçleri üzerinde durulacaktır.

Süzgeç tasarımı için takip eden bölümlerde önce *en dik iniş* yöntemi ile nümerik çözümler bulunmaya çalışılacak, daha sonra süzgeçler üzerindeki kısıtları arttırarak işlemleri hızlandıran *kafes parametre belirleme yöntemi* kullanılarak geliştirilen ve *BWT ters çevrimi* adı verilen yöntem anlatılacaktır.

4.1. En Dik İniş Yöntemiyle Süzgeç Tasarımı

En dik iniş algoritmasının temelinde, parametrik olarak ifade edilen bir fonksiyonun rastgele alınan bir noktasındaki gradientin tersi yönünde küçük adımlarla ilerlenerek fonksiyonun minimum noktasına ulaşmaya çalışılması yatar. Bu yüzden BWT süzgeçleri parametrelere bağlı olarak vektör biçiminde yazılacak ve 3 aşamalı BWT sonunda, verilen bir KLT matrisine ortalama kare bağlamında en yakın BWT matrisini veren 8-uzunluklu süzgeçler bulunmaya çalışılacaktır. 8 dışındaki uzunluklar için verilen yöntem benzer şekilde geliştirilebilir.

4.1.1. BWT Matrisinin Parametrik Hale Getirilmesi

BWT sisteminin alçak-geçiren süzgeci

$$\mathbf{h}_0 = [w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \ w_7 \ w_8]^T \quad (4.1)$$

olsun. Buradaki $w_i, i = 1, 2, \dots, 8$ değerleri, süzgecin $n = 0, 1, 2, \dots, 7$ anlarına karşılık gelen reel katsayılarıdır. Süzgecin diğer anlarındaki katsayıları sıfırdır.

Alçak-geçiren \mathbf{h}_0 süzgecine karşılık gelen yüksek-geçiren QMF süzgeç, (3.3) ile hesaplanır. Buna göre \mathbf{h}_0 süzgecinin katsayıları ters çevrilip çift indisli terimlerin işaretleri değiştirilmelidir. Böylece yüksek-geçiren süzgeç,

$$\mathbf{h}_1 = [w_8 \quad -w_7 \quad w_6 \quad -w_5 \quad w_4 \quad -w_3 \quad w_2 \quad -w_1]^T \quad (4.2)$$

şeklinde vektörel olarak gösterilir.

Blok dalgacık dönüşümünün ilk aşamasında sisteme uygulanan ilk girdi sinyali; kendisini 8 periyoduyla tekrar eden, $n = 0$ anındaki örneği 1, periyot içerisindeki diğer örnekleri 0 olan \mathbf{x}_1 sinyali olsun. Bu sinyalin bir periyot içerisindeki örneklerini vektörel olarak şöyle gösterelim:

$$\mathbf{x}_1 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T. \quad (4.3)$$

\mathbf{x}_1 sinyali, \mathbf{h}_0 ve \mathbf{h}_1 süzgeçlerinden geçirildikten sonra elde edilen sinyallerin 8-uzunluklu periyot içindeki terimleri şu şekilde bulunur:

$$\begin{aligned} \mathbf{x}_1 * \mathbf{h}_0 &= [w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6 \quad w_7 \quad w_8]^T \\ \mathbf{x}_1 * \mathbf{h}_1 &= [w_8 \quad -w_7 \quad w_6 \quad -w_5 \quad w_4 \quad -w_3 \quad w_2 \quad -w_1]^T. \end{aligned} \quad (4.4)$$

Bu sinyallerin 2 ile aşağı örneklenmesi sonucunda ise aşağıdaki sinyaller elde edilir:

$$\begin{aligned} 2 \downarrow (\mathbf{x}_1 * \mathbf{h}_0) &= [w_1 \quad w_3 \quad w_5 \quad w_7 \quad w_1 \quad w_3 \quad w_5 \quad w_7]^T \\ 2 \downarrow (\mathbf{x}_1 * \mathbf{h}_1) &= [w_8 \quad w_6 \quad w_4 \quad w_2 \quad w_8 \quad w_6 \quad w_4 \quad w_2]^T \end{aligned} \quad (4.5)$$

Denklem (4.4)'te verilen sinyaller 8-periyotlu olduğundan, 2 ile aşağı örnekleme sonucunda elde edilen (4.5)'teki sinyaller 4-periyotludur. Vektör içerisine sadece periyot içerisindeki örnekleri yazıyor olmamıza rağmen, ilerleyen kısımlarda kullanılacak matris çarpımlarındaki boyut uyumsuzluğunu engellemek için (4.5)'teki sinyaller 8-periyotlu olarak gösterilmiştir.

\mathbf{x}_1 sinyalinin bir sağa kaydırılmasıyla elde edilen ikinci girdi sinyalini $\mathbf{x}_2 = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$ vektörü ile; kaydırmaya devam ederek elde edilen son girdi sinyalini $\mathbf{x}_8 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^T$ vektörü ile gösterelim. Bu tanıma göre bütün girdi vektörlerinin elemanları şu şekilde ifade edilmiş olur:

$$\mathbf{x}_i[n] = \begin{cases} 1, & n = i \\ 0, & n \neq i \end{cases}, \quad i = 1, 2, \dots, 8. \quad (4.6)$$

Bütün bu girdi vektörlerinin yan yana dizilmesiyle oluşan matrisi \mathbf{X} ile gösterelim. Denklem (4.6)'dan \mathbf{X} matrisinin 8×8 boyutlu birim matris olduğu görülür.

3 aşamalı dalgacık sisteminin ilk aşaması sonunda elde edilen sinyallerin matris ve vektör çarpımı şeklinde ifade edilmesi mümkündür. Bütün $\mathbf{x}_i, i = 1, 2, \dots, 8$ vektörleri göz önüne alındığında, ilk aşama sonunda elde edilen sinyaller şu şekilde ifade edilebilirler:

$$\begin{aligned} 2 \downarrow (\mathbf{x}_i * \mathbf{h}_0) &= \mathbf{H}_0 \mathbf{x}_i \\ 2 \downarrow (\mathbf{x}_i * \mathbf{h}_1) &= \mathbf{H}_1 \mathbf{x}_i \end{aligned} \quad (4.7)$$

Buradaki \mathbf{H}_0 ve \mathbf{H}_1 matrislerinin şu şekilde olduğu görülür:

$$\mathbf{H}_0 = \begin{bmatrix} w_1 & w_8 & w_7 & w_6 & w_5 & w_4 & w_3 & w_2 \\ w_3 & w_2 & w_1 & w_8 & w_7 & w_6 & w_5 & w_4 \\ w_5 & w_4 & w_3 & w_2 & w_1 & w_8 & w_7 & w_6 \\ w_7 & w_6 & w_5 & w_4 & w_3 & w_2 & w_1 & w_8 \\ w_1 & w_8 & w_7 & w_6 & w_5 & w_4 & w_3 & w_2 \\ w_3 & w_2 & w_1 & w_8 & w_7 & w_6 & w_5 & w_4 \\ w_5 & w_4 & w_3 & w_2 & w_1 & w_8 & w_7 & w_6 \\ w_7 & w_6 & w_5 & w_4 & w_3 & w_2 & w_1 & w_8 \end{bmatrix} \quad (4.8)$$

$$\mathbf{H}_1 = \begin{bmatrix} w_8 & -w_1 & w_2 & -w_3 & w_4 & -w_5 & w_6 & -w_7 \\ w_6 & -w_7 & w_8 & -w_1 & w_2 & -w_3 & w_4 & -w_5 \\ w_4 & -w_5 & w_6 & -w_7 & w_8 & -w_1 & w_2 & -w_3 \\ w_2 & -w_3 & w_4 & -w_5 & w_6 & -w_7 & w_8 & -w_1 \\ w_8 & -w_1 & w_2 & -w_3 & w_4 & -w_5 & w_6 & -w_7 \\ w_6 & -w_7 & w_8 & -w_1 & w_2 & -w_3 & w_4 & -w_5 \\ w_4 & -w_5 & w_6 & -w_7 & w_8 & -w_1 & w_2 & -w_3 \\ w_2 & -w_3 & w_4 & -w_5 & w_6 & -w_7 & w_8 & -w_1 \end{bmatrix}$$

Bu matrislerin ilk dört satırının ikinci dört satırda tekrar ettiği görülmektedir. Bu durum 2 ile aşağı örneklemeden kaynaklanmaktadır.

İkinci aşama sonunda elde edilen sinyaller, birinci aşamada elde edilen sinyallerin \mathbf{h}_0 ve \mathbf{h}_1 süzgeçlerinden geçirilmesiyle oluşturulduğundan, ikinci aşamada elde edilen 4 sinyal, (4.7)'deki gibi \mathbf{H}_0 ve \mathbf{H}_1 matrisleri kullanılarak hesaplanabilir:

$$\begin{aligned} 2 \downarrow [2 \downarrow (\mathbf{x}_i * \mathbf{h}_0) * \mathbf{h}_0] &= \mathbf{H}_0 \mathbf{H}_0 \mathbf{x}_i \\ 2 \downarrow [2 \downarrow (\mathbf{x}_i * \mathbf{h}_0) * \mathbf{h}_1] &= \mathbf{H}_1 \mathbf{H}_0 \mathbf{x}_i \\ 2 \downarrow [2 \downarrow (\mathbf{x}_i * \mathbf{h}_1) * \mathbf{h}_0] &= \mathbf{H}_0 \mathbf{H}_1 \mathbf{x}_i \\ 2 \downarrow [2 \downarrow (\mathbf{x}_i * \mathbf{h}_1) * \mathbf{h}_1] &= \mathbf{H}_1 \mathbf{H}_1 \mathbf{x}_i. \end{aligned} \quad (4.9)$$

Benzer şekilde, üçüncü aşama sonunda elde edilen sinyaller de (4.9)'daki denklemlerin önüne \mathbf{H}_0 ve \mathbf{H}_1 çarpanlarının eklenmesiyle hesaplanabilir.

\mathbf{x}_i girdisi üçüncü aşama sonunda en üstteki altsinyalde $\mathbf{H}_0\mathbf{H}_0\mathbf{H}_0\mathbf{x}_i$ çıktısını verecektir. Bu çıktının periyodu 1 olduğundan vektördeki bütün elemanlar aynı olacaktır. Tekrar eden örnek, 8×8 boyutlu dönüşüm matrisinin ilk satırının i 'inci sütündeki elemanını oluşturacaktır. Bütün $\mathbf{x}_i, i = 1, 2, \dots, 8$ girdi vektörlerinin sisteme verilmesiyle en üstteki altsinyalde elde edilen vektörler yan yana yazılarak \mathbf{Y}_1 matrisinin oluşturulduğunu düşünelim. Bu durumda \mathbf{Y}_1 matrisi,

$$\mathbf{Y}_1 = \mathbf{H}_0\mathbf{H}_0\mathbf{H}_0\mathbf{X} \quad (4.10)$$

formülü ile ifade edilecektir. Aynı işlemin diğer altsinyaller için de tekrarlandığını ve her bant için $\mathbf{Y}_i, i = 1, 2, \dots, 8$ matrislerinin oluşturulduğunu düşünelim. Her bir matrisin satırları birbirinin aynısı olacaktır.

Üç aşamalı dalgacık sisteminin ürettiği dönüşüm matrisi \mathbf{Y} olsun. \mathbf{Y} matrisi, alt bantlarda elde edilen $\mathbf{Y}_i, i = 1, 2, \dots, 8$ matrislerinin tekrar edilen satırlarının birleştirilmesiyle oluşturulur. \mathbf{Y}_i matrisinde tekrar eden satır, \mathbf{Y} matrisinin i 'inci satırını oluşturur. $\mathbf{Y}_i, i = 1, 2, \dots, 8$ matrislerinin satırlarını ayırtırmak için 8×8 boyutlu $\mathbf{C}_i, i = 1, 2, \dots, 8$ matrislerini, i 'inci satırı 1'lerden, diğer satırları 0'lardan oluşacak şekilde tanımlayalım:

$$\mathbf{C}_i = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \leftarrow i\text{'inci satır} \quad (4.11)$$

Matrislerin satırlarını ayırtırmak amacıyla bir \odot operatörünü, sağındaki ve solundaki matrislerin elemanlarının birebir çarpılmasını sağlayan operatör olarak tanımlayalım (MATLAB'daki \cdot * operatörü gibi):

$$\mathbf{A} \odot \mathbf{B} = \mathbf{C} \quad \Leftrightarrow \quad \mathbf{C}_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j}. \quad (4.12)$$

Bu tanımlamalardan sonra \mathbf{Y} dönüşüm matrisi, $\mathbf{Y}_i, i = 1, 2, \dots, 8$ matrisleri ile şu şekilde yazılır:

$$\mathbf{Y} = \sum_{i=1}^8 \mathbf{C}_i \odot \mathbf{Y}_i. \quad (4.13)$$

$\mathbf{Y}_i, i = 1, 2, \dots, 8$ matrisleri, (4.10)'daki gibi $\mathbf{H}_0, \mathbf{H}_1,$ ve \mathbf{X} matrisleri cinsinden yazılıp, X matrisinin birim matris olduğu göz önüne alındığında şu şekilde ifade edilebilir:

$$\mathbf{Y}_i = \mathbf{H}_{k(i)}\mathbf{H}_{l(i)}\mathbf{H}_{m(i)}, i = 1, 2, \dots, 8. \quad (4.14)$$

Buradaki $k(i), l(i)$ ve $m(i)$ fonksiyonları, 0 veya 1 değerini alan ve $\mathbf{H}_0, \mathbf{H}_1$ matrislerinden hangisinin kullanılacağını gösteren fonksiyonlardır ve matematiksel olarak şu şekilde ifade edilebilirler:

$$\begin{aligned} k(i) &= \text{mod}(i - 1, 2) \\ l(i) &= \text{mod}([i - 1 - k(i)]/2, 2) \\ m(i) &= \text{mod}([i - 1 - k(i) - 2l(i)]/2, 2) \end{aligned} \quad (4.15)$$

$k(i), l(i)$ ve $m(i)$ değerleri, $i - 1$ sayısının ikilik düzendeki karşılığının basamaklarının tersten sıralanmasıyla da elde edilebilirler. Örneğin $i = 7$ için $7 - 1 = 6$ 'nın ikilik düzendeki karşılığı 110 olduğundan, $k(7) = 0, l(7) = 1$ ve $m(7) = 1$ olarak hesaplanır.

Böylece üç aşamalı BWT matrisi, \mathbf{h}_0 süzgecini oluşturan $w_i, i = 1, 2, \dots, 8$ katsayıları cinsinden, (4.8)'deki $\mathbf{H}_0, \mathbf{H}_1$ matrisleriyle ve (4.14)'teki \mathbf{Y}_i matrisleri kullanılarak (4.13)'teki gibi parametrik olarak ifade edilebilir.

4.1.2. KLT Matrisini Üreten Süzgeçlerin En Dik İnişle Bulunması

Bir KLT dönüşümü sonucunda elde edilen 8×8 boyutlu tekil dönüşüm matrisi \mathbf{A} olsun. \mathbf{A} matrisini üreten BWT süzgeçlerini bulmaya çalıştığımızı düşünelim. Alçak-geçiren \mathbf{h}_0 süzgecinin örneklerini başlangıçta rastgele değerler kullanarak oluşturalım. Bu değerlerle oluşturulan BWT matrisi \mathbf{Y} olsun. \mathbf{Y} matrisi ile \mathbf{A} matrisi arasındaki farkı kullanarak \mathbf{h}_0 süzgecinin katsayılarını adım adım güncelleyeceğiz.

En dik iniş algoritmasını kullanabilmek için \mathbf{Y} matrisinin \mathbf{A} matrisine ne kadar benzediğini ortaya koyan bir hata fonksiyonu belirlememiz gerekir. Bunun için iki matris arasındaki fark matrisinin elemanlarının kare toplamı kullanılabilir.

$$\mathbf{E} = \mathbf{Y} - \mathbf{A} \quad (4.16)$$

olmak üzere hata fonksiyonunu,

$$e = \text{topla}\{\mathbf{E} \odot \mathbf{E}\} \quad (4.17)$$

şeklinde gösterebiliriz. Buradaki “topla” fonksiyonu, parametre olarak aldığı matrisin bütün elemanlarının toplamını döndüren bir fonksiyon olarak tanımlanmıştır. “ \odot ” operatörü de (4.12)’de tanımlanan operatördür.

En dik iniş algoritmasında başlangıç noktasındaki gradientin tersi yönünde adımlar atmak gerektiğinden (4.17) ile verilen hata fonksiyonunun süzgeç katsayıları olan $w_n, n = 1, 2, \dots, 8$ parametrelerine göre türevlerini, $\frac{\partial e}{\partial w_n}$, hesaplamalıyız:

$$\frac{\partial e}{\partial w_n} = \frac{\partial}{\partial w_n} \text{topla}\{\mathbf{E} \odot \mathbf{E}\} = 2 \times \text{topla}\left\{\frac{\partial \mathbf{E}}{\partial w_n} \odot \mathbf{E}\right\}. \quad (4.18)$$

Denklem (4.16)’daki \mathbf{E} değeri yerine yazılıp \mathbf{A} matrisinin sabit bir matris olmasından ötürü türevlerinin sıfır olduğu göz önünde bulundurulduğunda,

$$\frac{\partial e}{\partial w_n} = 2 \times \text{topla}\left\{\frac{\partial \mathbf{Y}}{\partial w_n} \odot \mathbf{E}\right\} \quad (4.19)$$

bulunur. Bu denkleme göre hata fonksiyonunun gradientinin hesaplanabilmesi için $\frac{\partial \mathbf{Y}}{\partial w_n}$ türevlerinin hesaplanması gerekmektedir. Denklem (4.13) ve (4.14) kullanılarak şu hesaplamalar yapılabilir:

$$\begin{aligned} \frac{\partial \mathbf{Y}}{\partial w_n} &= \sum_{i=1}^8 \mathbf{C}_i \odot \frac{\partial \mathbf{Y}_i}{\partial w_n} \\ &= \sum_{i=1}^8 \mathbf{C}_i \odot \left\{ \frac{\partial \mathbf{H}_{k(i)}}{\partial w_n} \mathbf{H}_{l(i)} \mathbf{H}_{m(i)} + \mathbf{H}_{k(i)} \frac{\partial \mathbf{H}_{l(i)}}{\partial w_n} \mathbf{H}_{m(i)} \right. \\ &\quad \left. + \mathbf{H}_{k(i)} \mathbf{H}_{l(i)} \frac{\partial \mathbf{H}_{m(i)}}{\partial w_n} \right\} \end{aligned} \quad (4.20)$$

$k(i)$, $l(i)$ ve $m(i)$ fonksiyonları sadece 0 ve 1 değerlerini aldığından, gradient hesaplaması \mathbf{H}_0 ve \mathbf{H}_1 matrislerinin türevlerinin hesaplanmasına indirgenmiş olur. Denklem (4.8)’de verilen \mathbf{H}_0 ve \mathbf{H}_1 tanımlarından türevler kolayca hesaplanabilir. Burada bu matrislerin örnek olarak sadece w_1 parametresine göre türevlerini veriyoruz:

$$\frac{\partial \mathbf{H}_0}{\partial w_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.21)$$

$$\frac{\partial \mathbf{H}_1}{\partial w_1} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

\mathbf{H}_0 ve \mathbf{H}_1 matrislerinin diğer parametrelere göre türevleri de benzer şekilde -1 , 0 , ve $+1$ değerlerinden oluşacaktır.

Süzgeç katsayı, (4.19)'da verilen gradientin tersi yönünde, belirlenen bir adım miktarınca ilerletilerek güncellendikçe BWT matrisi \mathbf{Y} 'nin hedeflenen KLT matrisi \mathbf{A} 'ya yaklaşması beklenir.

n 'inci adımdaki süzgeç katsayıları $w_i^n, i = 1, 2, \dots, 8$ olsun. Uygun bir μ adım büyüklüğü seçildikten sonra $(n + 1)$ 'inci adımdaki süzgeç katsayıları en dik iniş algoritmasına göre şöyle hesaplanabilir:

$$w_i^{(n+1)} = w_i^{(n)} - \mu \frac{\partial e}{\partial w_i}. \quad (4.22)$$

4.1.3. En Dik İniş Algoritması

Denklem (4.22)'de verilen katsayı güncelleme formülü, aşağıdaki algoritma ile kullanılabilir:

1. Normalize bir başlangıç katsayı kümesi belirleyerek \mathbf{h}_0 ve \mathbf{h}_1 süzgeçlerini oluştur.
2. \mathbf{h}_0 ve \mathbf{h}_1 süzgeçlerini kullanarak BWT matrisi \mathbf{Y} 'yi hesapla.
3. Hedeflenen \mathbf{A} matrisi ile \mathbf{Y} matrisi arasındaki hatayı, e , hesapla.
4. Hata, belirlenen bir ε değerinden küçükse algoritmayı sonlandır. Değilse sonraki adıma geç.
5. Denklem (4.22)'de verilen formülle süzgeç katsayılarını güncelleyerek \mathbf{h}_0 süzgecini yeniden oluştur.
6. \mathbf{h}_0 süzgecini, $|\mathbf{h}_0|$ uzunluğuna bölerek normalize et.
7. \mathbf{h}_0 süzgecinin elemanlarını tersten sıralayıp çift sıradaki elemanların işaretlerini değiştirerek \mathbf{h}_1 süzgecini oluştur.
8. 2. adıma geri dön.

Bu algoritmanın 5. adımında katsayılar güncellenirken, süzgeçlerin uzunluğu 1'den farklı bir değer olabilir. Süzgeçlerin uzunluğunun 1 olması gerektiğinden, 6. adımda süzgeçlerin normunun 1 olması koşulu gerçekleştirilmektedir.

4.1.4. Örnek Uygulamalar

Algoritmanın başarımını görmek için ilk olarak bilinen bir süzgeç bankasının ürettiği dönüşüm matrisini kullanalım. Bu amaçla hedef dönüşüm matrisi \mathbf{A} 'yı, Daubechies-4 süzgeç bankasının ürettiği BWT matrisi olarak alalım. Normalize edilmiş Daubechies-4 süzgeçleri şu şekildedir:

$$\mathbf{h}_0 = \begin{bmatrix} 0.2304 \\ 0.7148 \\ 0.6309 \\ -0.0280 \\ -0.1870 \\ 0.0308 \\ 0.0329 \\ -0.0106 \end{bmatrix}, \quad \mathbf{h}_1 = \begin{bmatrix} -0.0106 \\ -0.0329 \\ 0.0308 \\ 0.1870 \\ -0.0280 \\ -0.6309 \\ 0.7148 \\ -0.2304 \end{bmatrix}. \quad (4.23)$$

Bu süzgeçlerle üretilen BWT dönüşüm matrisi şöyledir:

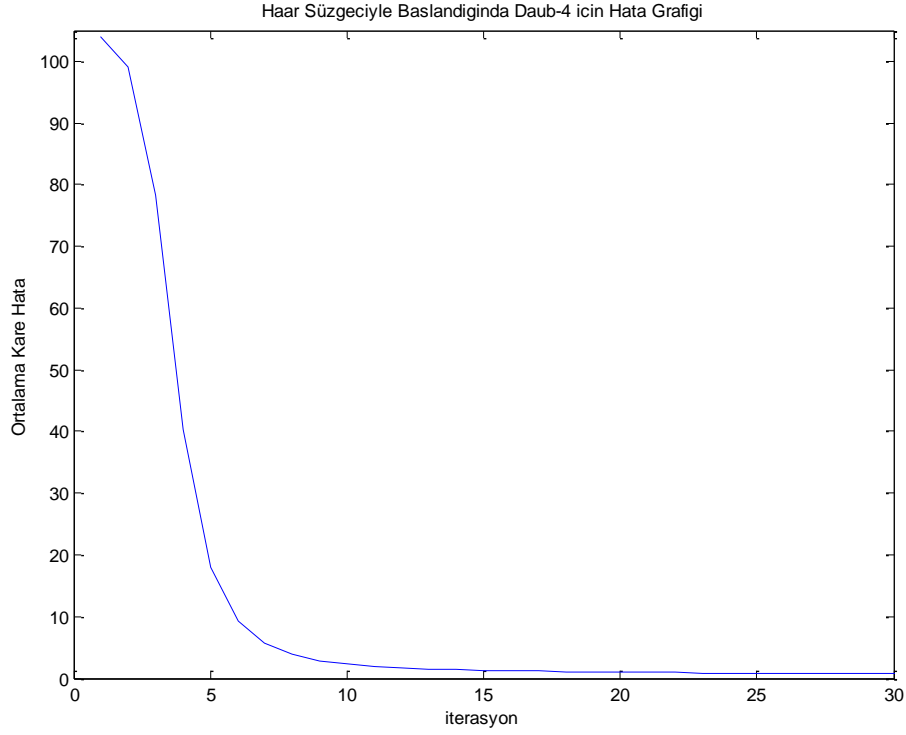
$$\mathbf{A} = \begin{bmatrix} 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 \\ -0.51 & -0.39 & 0.03 & 0.29 & 0.51 & 0.39 & -0.03 & -0.29 \\ -0.31 & -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 \\ 0.03 & 0.29 & 0.51 & 0.39 & -0.03 & -0.29 & -0.51 & -0.39 \\ 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 \\ 0.37 & -0.15 & -0.31 & 0.49 & -0.37 & 0.15 & 0.31 & -0.49 \\ -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 & -0.31 \\ -0.31 & 0.49 & -0.37 & 0.15 & 0.31 & -0.49 & 0.37 & -0.15 \end{bmatrix} \quad (4.24)$$

En dik iniş algoritmasının başlangıç süzgeçleri olarak 8-uzunluklu Haar süzgeçlerini seçelim:

$$\mathbf{h}_0^{(0)} = \frac{[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T}{2\sqrt{2}} \quad (4.25)$$

$$\mathbf{h}_1^{(0)} = \frac{[1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1]^T}{2\sqrt{2}}.$$

Algoritmanın 4. basamağında kullanılan kabul edilebilir hata miktarı $\varepsilon = 10^{-5}$ ve (4.22)'de kullanılan adım büyüklüğü $\mu = 10^{-3}$ alındığında 791 adım sonunda hata miktarı ε değerinin altına düşmektedir. İlk 30 adım için (4.17) ile verilen hatanın grafiği Şekil 4.1'de gösterilmiştir (30. adımdan sonra hata çok hızlı değişmediğinden sadece ilk 30 adım gösterilmiştir).



Şekil 4.1. Nümerik yöntemde Daub-4 dönüşüm matrisini elde etmek için Haar süzgeçleriyle başlandığında elde edilen hata grafiği

Hata, ilk adımlarda çok hızlı bir şekilde düşmesine rağmen sonraları daha yavaş düşmektedir. 791 adım sonunda çözüm olarak elde edilen süzgeçler şöyledir:

$$\mathbf{h}_0^{(791)} = \begin{bmatrix} 0.2298 \\ 0.7149 \\ 0.6309 \\ -0.0285 \\ -0.1873 \\ 0.0315 \\ 0.0334 \\ -0.0111 \end{bmatrix}, \quad \mathbf{h}_1^{(791)} = \begin{bmatrix} -0.0111 \\ -0.0334 \\ 0.0315 \\ 0.1873 \\ -0.0285 \\ -0.6309 \\ 0.7149 \\ -0.2298 \end{bmatrix}. \quad (4.26)$$

Bu süzgeçler, (4.23)'te verilen çözümün hemen hemen aynısıdır. Bu süzgeçlerle elde edilen dönüşüm matrisi şöyledir:

$$\mathbf{Y} = \begin{bmatrix} 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 \\ -0.52 & -0.39 & 0.03 & 0.29 & 0.52 & 0.39 & -0.03 & -0.29 \\ -0.31 & -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 \\ 0.03 & 0.29 & 0.52 & 0.39 & -0.03 & -0.29 & -0.52 & -0.39 \\ 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 \\ 0.37 & -0.15 & -0.31 & 0.49 & -0.37 & 0.15 & 0.31 & -0.49 \\ -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 & -0.31 \\ -0.31 & 0.50 & -0.37 & 0.15 & 0.31 & -0.49 & 0.37 & -0.15 \end{bmatrix} \quad (4.27)$$

Denklem (4.27)'deki Y matrisinin elemanları, (4.24)'teki A matrisinin elemanlarından, üçüncü veya dördüncü ondalık basamaktan itibaren farklılık göstermektedir.

Matlab'da yazılan kodun çalışması, Pentium Core 2 Duo 2.20 GHz işlemcili bilgisayarda 1.5 saniye civarında sürmüştür.

En dik iniş algoritması, seçilen adım büyüklüğüne, μ , doğrudan bağlıdır. Büyük adımlar seçildiğinde güncelleme sırasında minimum nokta atlanır ve hatayı minimum yapan noktanın bulunması imkânsız hale gelebilir. Adım büyüklüğü küçük seçildiğinde ise minimum noktaya ulaşılması çok uzun sürebilir. $\varepsilon = 10^{-5}$ alındığında, seçilen bazı adım büyüklükleri için elde edilen iterasyon sayıları Çizelge 4.1'de verilmiştir.

Çizelge 4.1'den optimum adım büyüklüğü 0.0004'ten 0.0014'e kadar arttırıldığında iterasyon sayısının azaldığı görülmektedir. Fakat adım büyüklüğü 0.0015 olduğunda çözüm bulunamamaktadır. Buna göre adım büyüklüğünün dikkatli seçilmesi gerektiği ortaya çıkmaktadır.

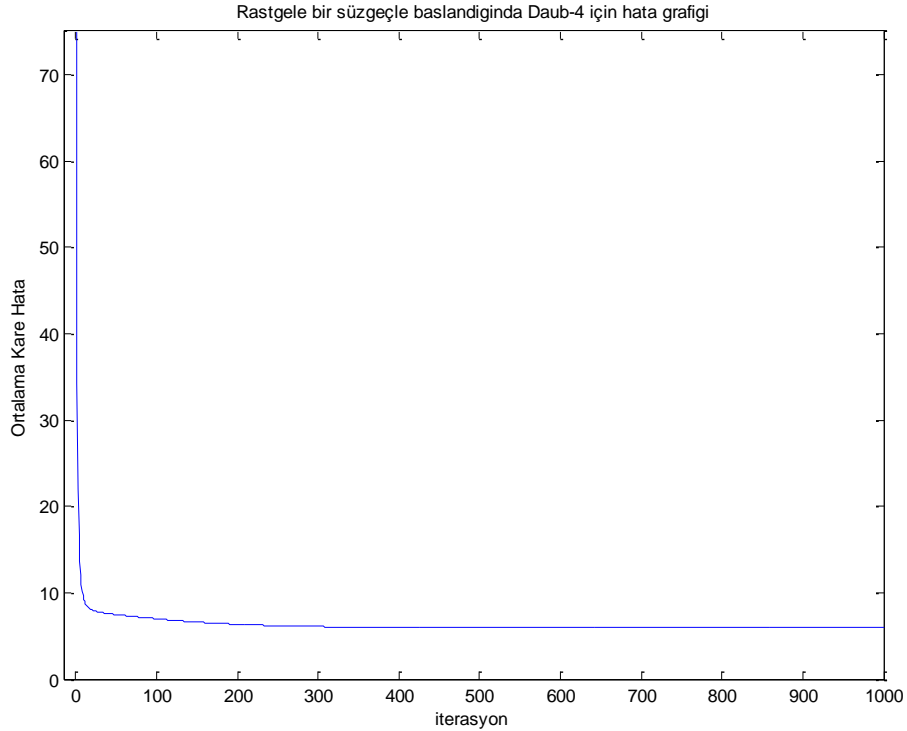
Çizelge 4.1. db4'ün dönüşüm matrisinin Haar süzgeçleriyle başlanarak nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-5}$ için μ adım büyüklüğüne karşı iterasyon sayıları

Adım büyüklüğü (μ)	İterasyon sayısı
0.0004	1959
0.0005	1570
0.0006	1310
0.0007	1125
0.0008	986
0.0009	878
0.0010	791
0.0011	720
0.0012	661
0.0013	611
0.0014	569
0.0015	> 10000 (çözüm yok)

En dik iniş yönteminde başlangıç noktasının önemi büyüktür. Başlangıç noktası doğru seçilmediğinde global minimum yerine yerel bir minimum bulunabilir. Örneğin başlangıç süzgeçleri olarak, değerleri rastgele belirlenmiş

$$\mathbf{h}_0^{(0)} = \begin{bmatrix} -0.4146 \\ -0.3412 \\ -0.3459 \\ 0.1192 \\ -0.2615 \\ -0.3458 \\ -0.5565 \\ 0.2833 \end{bmatrix}, \quad \mathbf{h}_1^{(0)} = \begin{bmatrix} 0.2833 \\ 0.5565 \\ -0.3458 \\ 0.2615 \\ 0.1192 \\ 0.3459 \\ -0.3412 \\ 0.4146 \end{bmatrix} \quad (4.28)$$

süzgeçleri kullanıldığında, Çizelge 4.1’de kullanılan bütün μ değerleri için hata 5.9547 değerinin altına düşmemektedir. $\mu = 10^{-3}$ için elde edilen hata grafiği Şekil 4.2’de verilmiştir. Grafikte hatanın başlangıçta çok hızlı azaldığı fakat sonrasında sabit giderek hiç değişmediği gözlenmektedir.



Şekil 4.2. Nümerik yöntemde Daub-4 dönüşüm matrisini elde etmek için rastgele bir süzgeçle başlandığında hata grafiği

4.2. Kafes Parametre Belirleme Yöntemi

Bölüm 4.1’de anlatılan nümerik yöntemde süzgeç katsayıları üzerinde sadece (3.17)’de verilen diklik koşulu kısıtı kullanılmıştı. QMF bankalarındaki süzgeçlerde ise kısıtlar şöyledir:

1. Süzgeçlerin normları 1 olmalı:

$$\sum_k \mathbf{h}_0^2(k) = 1, \quad \sum_k \mathbf{h}_1^2(k) = 1. \quad (4.29)$$

2. Süzgeçler birbirlerine ve her birinin ikinin katları kadar ötelenmiş hallerine dik olmalı:

$$\sum_k \mathbf{h}_0(k)\mathbf{h}_0(k - 2n) = 0, \quad \forall n \neq 0. \quad (4.30)$$

3. Alçak-geçiren süzgeç, düşük frekansları geçirip yüksek frekansları süzmeli; yani *alçak-geçiren koşulunu* sağlamalı:

$$\sum_k (-1)^k \mathbf{h}_0(k) = 0. \quad (4.31)$$

Nümerik yöntemde yüksek-geçiren süzgeç, alçak-geçiren süzgecin elemanlarının tersten sıralanıp çift indisli terimlerin işaretlerinin değiştirilmesiyle elde edildiğinden ikinci koşul otomatik olarak sağlanmaktadır. Süzgeçlerin hataya göre güncellenmeleri sırasında normların 1 olması gerektiği göz ardı edilmiş, ancak süzgeçler güncellendikten sonra normalize edilerek bu koşul gerçeklenmeye çalışılmıştır. Üçüncü koşul ise nümerik yöntemde göz önünde bulundurulmamıştır.

Denklem (4.29), (4.30) ve (4.31)’de verilen QMF koşulları kullanılarak süzgeçlerdeki serbest parametre sayısı azaltılabilir.

QMF koşullarını sağlayan 2-uzunluklu alçak-geçiren süzgeç sadece $\mathbf{h}_0 = \pm \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]^T$ Haar süzgeci olabilir.

Koşulları sağlayan 4-uzunluklu alçak-geçiren süzgeç ise tek bir α parametresine bağlı olarak şu şekilde yazılabilir [30]:

$$\begin{aligned} \mathbf{h}_0(1) &= \frac{1 - \cos \alpha + \sin \alpha}{2\sqrt{2}} \\ \mathbf{h}_0(2) &= \frac{1 + \cos \alpha + \sin \alpha}{2\sqrt{2}} \end{aligned} \quad (4.32)$$

$$\mathbf{h}_0(3) = \frac{1 + \cos \alpha - \sin \alpha}{2\sqrt{2}}$$

$$\mathbf{h}_0(4) = \frac{1 - \cos \alpha - \sin \alpha}{2\sqrt{2}}$$

Daha uzun süzgeçler için *kafes parametre belirleme* yöntemi kullanılabilir [37]. Bu yöntem aşağıda verilmiştir.

Alçak- ve yüksek-geçiren süzgeçler sırasıyla $H_0(z)$ ve $H_1(z)$ olsun. $H_0(z)$ 'nin çok-fazlı bileşenleri $H_{00}(z)$ ve $H_{01}(z)$; $H_1(z)$ 'nininkiler de $H_{10}(z)$ ve $H_{11}(z)$ olsun. Çok-fazlı bileşenler cinsinden $H_0(z)$ ve $H_1(z)$ şöyle yazılır:

$$\begin{aligned} H_0(z) &= H_{00}(z^2) + z^{-1}H_{01}(z^2) \\ H_1(z) &= H_{10}(z^2) + z^{-1}H_{11}(z^2). \end{aligned} \quad (4.33)$$

Bu durumda 2-kanallı süzgeç bankasının çok-fazlı matrisi şu şekilde tanımlanır:

$$H_p(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{bmatrix}. \quad (4.34)$$

Tekil süzgeç bankaları için çok-fazlı matris şu şekilde yazılabilir [38]:

$$H_p(z) = \begin{bmatrix} 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \cdot R(\theta_K) \cdot \Lambda(z) \cdot R(\theta_{K-1}) \cdot \Lambda(z) \cdots \Lambda(z) \cdot R(\theta_0). \quad (4.35)$$

Buradaki $R(\theta)$ ve $\Lambda(z)$ matrisleri şu şekilde tanımlanmıştır:

$$\begin{aligned} R(\theta) &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \\ \Lambda(z) &= \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}. \end{aligned} \quad (4.36)$$

Denklem (4.34)'le ifade edilen kafes parametre belirleme yöntemi, $H_p(z)$ matrisini, açılarının bir fonksiyonu şeklinde verir. $H_0(z = -1) = 0$ koşulu sebebiyle açılarının toplamı $\pi/4$ olmalıdır. Böylece N uzunluğundaki süzgeç bankaları, $(N - 2)/2$ açı parametresine bağlı olarak yazılabilir.

Selesnick [37] tarafından hazırlanan Maple kodu $N = 8$ için çalıştırıldığında 3 aşamalı blok dalgacık sisteminde kullanılan alçak-geçiren süzgecin elemanları, t_0 , t_1 ve t_2 açılarına bağlı olarak şu şekilde bulunmaktadır:

$$\begin{aligned} t_3 &= \frac{\pi}{4} - t_0 - t_1 - t_2, \\ h_0(1) &= \cos(t_3) \cos(t_2) \cos(t_1) \cos(t_0), \\ h_0(2) &= \cos(t_3) \cos(t_2) \cos(t_1) \sin(t_0), \end{aligned} \quad (4.37)$$

$$\begin{aligned}
h_0(3) &= -\cos(t_3) \cos(t_2) \sin(t_1) \sin(t_0) \\
&\quad - \cos(t_3) \sin(t_2) \sin(t_1) \cos(t_0) \\
&\quad - \sin(t_3) \sin(t_2) \cos(t_1) \cos(t_0), \\
h_0(4) &= \cos(t_3) \cos(t_2) \sin(t_1) \cos(t_0) \\
&\quad - \cos(t_3) \sin(t_2) \sin(t_1) \sin(t_0) \\
&\quad - \sin(t_3) \sin(t_2) \cos(t_1) \sin(t_0), \\
h_0(5) &= -\cos(t_3) \sin(t_2) \cos(t_1) \sin(t_0) \\
&\quad + \sin(t_3) \sin(t_2) \sin(t_1) \sin(t_0) \\
&\quad - \sin(t_3) \cos(t_2) \sin(t_1) \cos(t_0), \\
h_0(6) &= \cos(t_3) \sin(t_2) \cos(t_1) \cos(t_0) \\
&\quad - \sin(t_3) \sin(t_2) \sin(t_1) \cos(t_0) \\
&\quad - \sin(t_3) \cos(t_2) \sin(t_1) \sin(t_0), \\
h_0(7) &= -\sin(t_3) \cos(t_2) \cos(t_1) \sin(t_0), \\
h_0(8) &= \sin(t_3) \cos(t_2) \cos(t_1) \cos(t_0).
\end{aligned}$$

4.2.1. Kafes Parametre Belirleme ile Nümerik Yöntem

Kafes parametre belirleme yöntemiyle serbest parametre sayısı azaltıldıktan sonra en dik iniş algoritması daha etkili bir şekilde uygulanabilir. Bu amaçla BWT matrisi, kafes parametre belirlemeyle elde edilen daha az sayıdaki serbest parametreyle ifade edilir, hatanın bu parametrelere göre türevleri hesaplanır ve seçilen bir başlangıç noktasından gradientin tersi yönünde ilerlenerek hatayı minimum yapan noktaya yaklaşılr.

4.2.1.1. Kafes Parametre Belirleme ile En Dik İniş Algoritması

Alçak- ve yüksek-geçiren süzgeçler

$$\begin{aligned}
\mathbf{h}_0 &= [w_1 \quad w_2 \quad w_3 \quad w_4]^T \\
\mathbf{h}_1 &= [w_4 \quad -w_3 \quad w_2 \quad -w_1]^T
\end{aligned} \tag{4.38}$$

olsun. Buradaki $w_i, i = 1, 2, 3, 4$ katsayıları (4.32) ile α cinsinden yazılabilirler. Denklem (4.8)'de verilen \mathbf{H}_0 ve \mathbf{H}_1 matrisleri iki aşamalı dalgacık sisteminde şu şekli alırlar:

$$\mathbf{H}_0 = \begin{bmatrix} w_1 & w_4 & w_3 & w_2 \\ w_3 & w_2 & w_1 & w_4 \\ w_1 & w_4 & w_3 & w_2 \\ w_3 & w_2 & w_1 & w_4 \end{bmatrix} \quad (4.39)$$

$$\mathbf{H}_1 = \begin{bmatrix} w_4 & -w_1 & w_2 & -w_3 \\ w_2 & -w_3 & w_4 & -w_1 \\ w_4 & -w_1 & w_2 & -w_3 \\ w_2 & -w_3 & w_4 & -w_1 \end{bmatrix}.$$

Denklem (4.14)'te verilen $\mathbf{Y}_i, i = 1, 2, 3, 4$ matrisleri iki aşamada şöyle hesaplanır:

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{H}_0 \mathbf{H}_0 \\ \mathbf{Y}_2 &= \mathbf{H}_1 \mathbf{H}_0 \\ \mathbf{Y}_3 &= \mathbf{H}_0 \mathbf{H}_1 \\ \mathbf{Y}_4 &= \mathbf{H}_1 \mathbf{H}_1. \end{aligned} \quad (4.40)$$

Denklem (4.13)'teki dönüşüm matrisi, \mathbf{Y} , şimdiki durumda şöyle hesaplanır:

$$\mathbf{Y} = \sum_{i=1}^4 \mathbf{C}_i \odot \mathbf{Y}_i. \quad (4.41)$$

Buradaki \mathbf{C}_i matrisleri, i 'inci satırı 1, diğer satırları 0 olan 4×4 boyutlu matrislerdir.

$$\mathbf{E} = \mathbf{Y} - \mathbf{A} \quad (4.42)$$

olmak üzere hata fonksiyonunu,

$$e = \text{topla}\{\mathbf{E} \odot \mathbf{E}\} \quad (4.43)$$

şeklinde tanımlarız.

Bölüm 4.1.2'de hata fonksiyonunun w_i katsayılarına göre türevlerini hesaplamıştık. Şimdi ise serbest parametre α olduğu için hatanın α 'ya göre türevini hesaplamamız gerekir. Denklem (4.42), (4.43), ve \mathbf{A} matrisinin sabit olmasından,

$$\frac{de}{d\alpha} = 2 \times \text{topla}\left\{\frac{d\mathbf{Y}}{d\alpha} \odot \mathbf{E}\right\} \quad (4.44)$$

yazılabilir. Bu denklemdeki $d\mathbf{Y}/d\alpha$ terimleri, zincir kuralı kullanılarak şöyle hesaplanır:

$$\frac{d\mathbf{Y}}{d\alpha} = \frac{dw_1}{d\alpha} \frac{d\mathbf{Y}}{dw_1} + \frac{dw_2}{d\alpha} \frac{d\mathbf{Y}}{dw_2} + \frac{dw_3}{d\alpha} \frac{d\mathbf{Y}}{dw_3} + \frac{dw_4}{d\alpha} \frac{d\mathbf{Y}}{dw_4}. \quad (4.45)$$

Bu denklemdeki w_i katsayılarının α 'ya göre türevleri, $dw_i/d\alpha$, (4.32)'den şöyle bulunur:

$$\begin{aligned}
\frac{dw_1}{d\alpha} &= \frac{\cos \alpha + \sin \alpha}{2\sqrt{2}} \\
\frac{dw_2}{d\alpha} &= \frac{\cos \alpha - \sin \alpha}{2\sqrt{2}} \\
\frac{dw_3}{d\alpha} &= \frac{-\cos \alpha - \sin \alpha}{2\sqrt{2}} \\
\frac{dw_4}{d\alpha} &= \frac{-\cos \alpha + \sin \alpha}{2\sqrt{2}}.
\end{aligned} \tag{4.46}$$

\mathbf{Y} matrisinin w_n katsayılarına göre türevleri ise şöyle bulunur:

$$\begin{aligned}
\frac{\partial \mathbf{Y}}{\partial w_n} &= \sum_{i=1}^4 \mathbf{C}_i \odot \frac{\partial \mathbf{Y}_i}{\partial w_n} \\
&= \sum_{i=1}^8 \mathbf{C}_i \odot \left\{ \frac{\partial \mathbf{H}_{k(i)}}{\partial w_n} \mathbf{H}_{l(i)} + \mathbf{H}_{k(i)} \frac{\partial \mathbf{H}_{l(i)}}{\partial w_n} \right\}
\end{aligned} \tag{4.47}$$

Buradaki $k(i)$ ve $l(i)$ fonksiyonları, ya 0 ya da 1 değerini alan ve (4.15) ile tanımlanmış fonksiyonlardır.

Bundan sonra (4.45), (4.46) ve (4.47)'deki ifadeler (4.44)'te yerlerine yazıldığında gradient hesaplanabilir ve α katsayısı

$$\alpha^{(n+1)} = \alpha^{(n)} - \mu \frac{de}{d\alpha} \tag{4.48}$$

formülüyle iteratif olarak güncellenebilir.

4.2.1.2. db2 Dönüşüm Matrisini Üreten Süzgeç Bankasının Bulunması

Yukarıda anlatılan algoritmanın ilk uygulaması olarak Daubechies-2 süzgeç bankasının ürettiği 4×4 boyutlu BWT matrisini üreten süzgeç bankasını kafes parametre belirleme yöntemi ile bulmaya çalışalım. Çözüm, Daubechies-2 süzgeç bankasının kendisi olmalıdır:

$$\begin{aligned}
\mathbf{h}_0^{(db2)} &= [0.4830 \quad 0.8365 \quad 0.2241 \quad -0.1294]^T \\
\mathbf{h}_1^{(db2)} &= [-0.1294 \quad -0.2241 \quad 0.8365 \quad -0.4830]^T.
\end{aligned} \tag{4.49}$$

Bu süzgeçlerin ürettiği BWT matrisi, yani hedef dönüşüm matrisi şudur:

$$\mathbf{A} = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.1830 & -0.6830 & -0.1830 & 0.6830 \\ 0.5000 & -0.5000 & 0.5000 & -0.5000 \\ -0.6830 & -0.1830 & 0.6830 & 0.1830 \end{bmatrix}. \tag{4.50}$$

En dik iniş algoritmasının adım büyüklüğü $\mu = 10^{-2}$, kabul edilebilir hata miktarı $\varepsilon = 10^{-6}$ ve α 'nın başlangıç değeri 0 alındığında 183 iterasyon sonunda çözüme ulaşılır. Matlab'da yazılan kodun Pentium Core 2 Duo 2.20 GHz işlemcili bir bilgisayarda çalışması 300 ms. civarında süre almıştır. İterasyona bağlı olarak hatanın azalma grafiği Şekil 4.3'te verilmiştir.

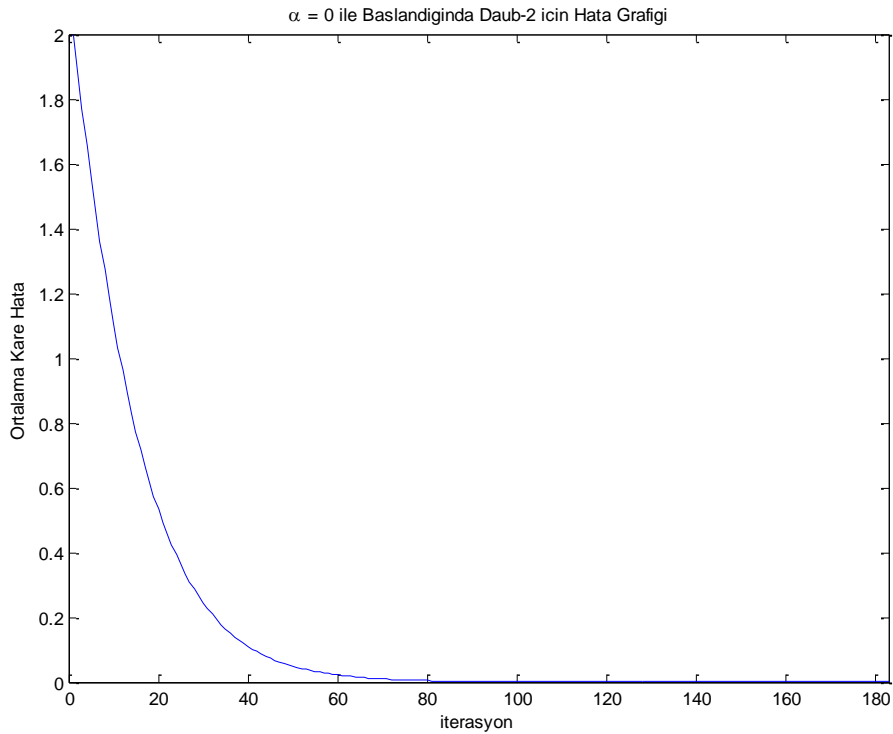
183 iterasyon sonunda hata 10^{-6} 'nın altına düştüğünde elde edilen süzgeçler şu şekildedir:

$$\begin{aligned} \mathbf{h}_0^{(183)} &= [0.4826 \quad 0.8366 \quad 0.2245 \quad -0.1295]^T \\ \mathbf{h}_1^{(183)} &= [-0.1295 \quad -0.2245 \quad 0.8366 \quad -0.4826]^T. \end{aligned} \quad (4.51)$$

Bu süzgeçler, (4.48)'de verilen çözümün hemen hemen aynıdır ve bu süzgeçlerle üretilen BWT matrisi,

$$\mathbf{Y} = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.1825 & -0.6831 & -0.1825 & 0.6831 \\ 0.5000 & -0.5000 & 0.5000 & -0.5000 \\ -0.6831 & -0.1825 & 0.6831 & 0.1825 \end{bmatrix}$$

(4.50)'de verilen \mathbf{A} matrisine çok yakındır.



Şekil 4.3. db2'nin BWT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği

Çizelge 4.2. db2'nin dönüşüm matrisinin $\alpha = 0$ ile başladığında kafes parametre belirlemeli nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-6}$ için μ adım büyüklüğüne karşı iterasyon sayıları

Adım büyüklüğü (μ)	İterasyon sayısı
> 0.600	> 10000 (Çözüm yok)
0.500	> 10000 (Çözümeye yaklaşıyor)
0.400	14
0.300	4
0.200	6
0.100	16
0.050	35
0.010	183
0.005	368
0.001	1847

Seçilen adım büyüklüğüne göre çözüme ulaşmak için kullanılan iterasyon sayıları Çizelge 4.2'de verilmiştir.

4.2.1.3. db4 Dönüşüm Matrisini Üreten Süzgeç Bankasının Bulunması

İkinci bir uygulama olarak Bölüm 4.2.1.2'de kullanılan algoritmayı Daubeschies-2 yerine Daubeschies-4 süzgeç bankasının ürettiği 8×8 boyutundaki dönüşüm matrisini üreten süzgeç bankasını bulmaya çalışalım. Bu durumda kullanılacak kafes parametre belirleme açıları 1 yerine 3 tane olacak ve matrisler de 4×4 yerine 8×8 boyutlu olacaktır. Denklemler benzer olduğundan algoritmanın detayı burada anlatılmayacaktır.

Başlangıç noktası olarak $t_0 = t_1 = t_2 = 0$ ve adım büyüklüğü $\mu = 10^{-3}$ alındığında sonucu ulaşılamamaktadır. Bu yüzden başlangıç noktasını rastgele seçmek yerinde olacaktır. Başlangıç açıları radyan cinsinden $t_0 = 4.6790$, $t_1 = 1.1872$, $t_2 = 4.3151$ alındığında elde edilen iterasyon-hata grafiği Şekil 4.4'te verilmiştir. Bu deneyde kabul edilebilir hata oranı $\varepsilon = 10^{-5}$ alınmış ve 805 iterasyon sonunda hata bu değer altına düşmüştür. Matlab'da yazılan kodun

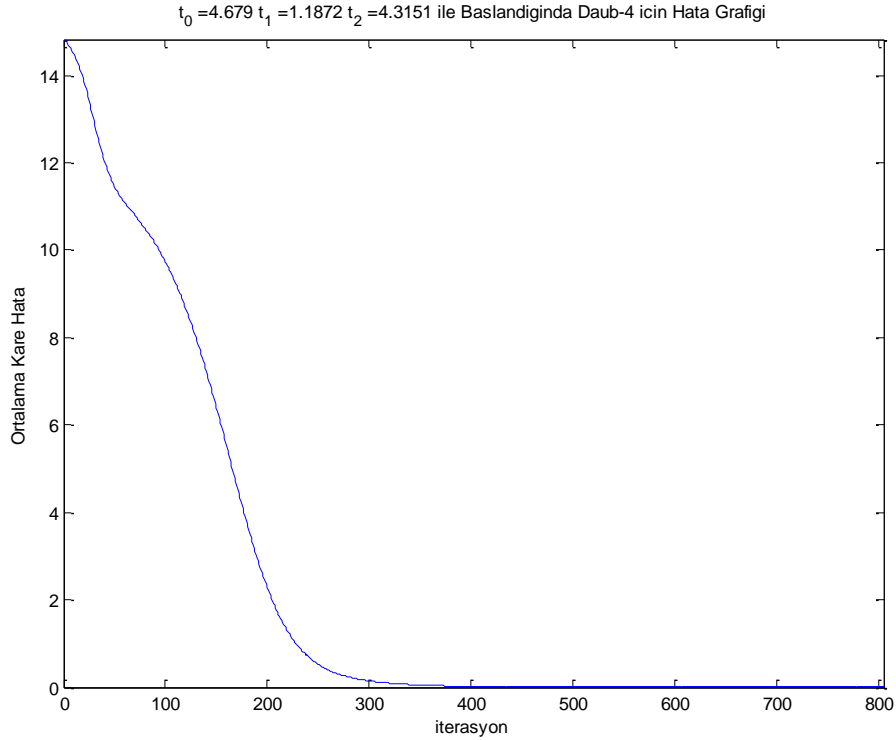
çalışması Pentium Core 2 Duo 2.20 GHz işlemcili bir bilgisayarda 1 saniye civarında sürmüştür. Sonuçta elde edilen süzgeçler şu şekildedir:

$$\mathbf{h}_0^{(805)} = \begin{bmatrix} 0.2311 \\ 0.7145 \\ 0.6308 \\ -0.0279 \\ -0.1877 \\ 0.0312 \\ 0.0330 \\ -0.0107 \end{bmatrix}, \quad \mathbf{h}_1^{(805)} = \begin{bmatrix} -0.0107 \\ -0.0330 \\ 0.0312 \\ 0.1877 \\ -0.0279 \\ -0.6308 \\ 0.7145 \\ -0.2311 \end{bmatrix}. \quad (4.52)$$

Bu değerler, (4.23)'te verilen gerçek sonuçla hemen hemen aynıdır. Bu süzgeçlerle elde edilen BWT matrisi,

$$\mathbf{Y} = \begin{bmatrix} 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 & 0.35 \\ -0.52 & -0.39 & 0.03 & 0.29 & 0.52 & 0.39 & -0.03 & -0.29 \\ -0.31 & -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 \\ 0.03 & 0.29 & 0.52 & 0.39 & -0.03 & -0.29 & -0.52 & -0.39 \\ 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 & 0.35 & -0.35 \\ 0.37 & -0.15 & -0.31 & 0.49 & -0.37 & 0.15 & 0.31 & -0.49 \\ -0.39 & 0.31 & 0.39 & -0.31 & -0.39 & 0.31 & 0.39 & -0.31 \\ -0.31 & 0.49 & -0.37 & 0.15 & 0.31 & -0.49 & 0.37 & -0.15 \end{bmatrix} \quad (4.53)$$

bulunmuştur ve bu matris (4.24)'te verilen dönüşüm matrisinden üçüncü ondalık basamaktan itibaren farklılık göstermektedir.



Şekil 4.4. db4'ün BWT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği

Çizelge 4.3. db4'ün dönüşüm matrisinin $t_0 = 4.6790$, $t_1 = 1.1872$, $t_2 = 4.3151$ ile başladığında kafes parametre belirlemeli nümerik yöntemle elde edilmesinde $\varepsilon = 10^{-5}$ için μ adım büyüklüğüne karşı iterasyon sayıları

Adım büyüklüğü (μ)	İterasyon sayısı
> 0.018	(Çözüm yok)
0.017	50
0.016	52
0.015	56
0.014	60
0.013	64
0.012	69
0.011	75
0.010	83
0.005	163
0.001	805

Aynı başlangıç noktası seçildiğinde çözüme ulaşmak için gereken iterasyon sayıları Çizelge 4.3'te gösterilmiştir.

4.2.1.4. Lena'nın KLT Matrislerini Üreten Süzgeç Bankasının Bulunması

Bölüm 2.1.2'de anlatıldığı üzere resimler üzerinde KLT dönüşümü için satırlar ve sütunlar için ayrı ayrı iki dönüşüm matrisi hesaplamak gerekmektedir.

Lena resminin satırları üzerinden hesaplanan KLT matrisi şöyledir:

$$A_{Lena}^{(sattır)} = \begin{bmatrix} 0.3501 & 0.3534 & 0.3557 & 0.3566 \\ 0.4770 & 0.4198 & 0.2896 & 0.1030 \\ 0.4648 & 0.1915 & -0.1848 & -0.4526 \\ 0.4493 & -0.1079 & -0.4931 & -0.2703 \\ 0.3511 & -0.3831 & -0.3100 & 0.3806 \\ 0.2637 & -0.4816 & 0.1178 & 0.4093 \\ 0.1616 & -0.4030 & 0.4254 & -0.1685 \\ -0.1144 & 0.3308 & -0.4705 & 0.4935 \end{bmatrix} \quad (4.54)$$

$$\begin{bmatrix} 0.3558 & 0.3542 & 0.3525 & 0.3500 \\ -0.1043 & -0.2854 & -0.4164 & -0.4859 \\ -0.4585 & -0.2011 & 0.1875 & 0.4716 \\ 0.2880 & 0.4603 & 0.0866 & -0.4098 \\ \dots & 0.3226 & -0.3564 & -0.3601 & 0.3582 \\ -0.4483 & -0.0638 & 0.4831 & -0.2805 \\ -0.2371 & 0.5180 & -0.4896 & 0.1945 \\ -0.4561 & 0.3759 & -0.2386 & 0.0793 \end{bmatrix}$$

Sütunlar üzerinden hesaplanan KLT matrisi ise şöyledir:

$$A_{Lena}^{(sütun)} = \begin{bmatrix} 0.3516 & 0.3533 & 0.3544 & 0.3548 \\ 0.4819 & 0.4210 & 0.2873 & 0.1027 \\ 0.4638 & 0.1832 & -0.1886 & -0.4668 \\ 0.4400 & -0.1006 & -0.4925 & -0.2557 \\ 0.3439 & -0.3654 & -0.3253 & 0.3563 \dots \\ 0.2290 & -0.3875 & 0.0435 & 0.3466 \\ 0.2182 & -0.4765 & 0.3278 & 0.1219 \\ -0.1307 & 0.3788 & -0.5464 & 0.5643 \end{bmatrix}$$

(4.55)

$$\begin{bmatrix} 0.3549 & 0.3547 & 0.3532 & 0.3515 \\ -0.1080 & -0.2912 & -0.4198 & -0.4740 \\ -0.4509 & -0.1941 & 0.1955 & 0.4676 \\ 0.2722 & 0.4800 & 0.0778 & -0.4217 \\ \dots & 0.3399 & -0.3382 & -0.3820 & 0.3737 \\ -0.2170 & -0.3300 & 0.6424 & -0.3267 \\ -0.4883 & 0.5108 & -0.3079 & 0.0941 \\ -0.4289 & 0.1928 & -0.0355 & 0.0054 \end{bmatrix}$$

Bu matrislerden birinci ve beşinci satırlar dikkat çekmektedir. Birinci satırın elemanları $0.3536 = \frac{1}{2\sqrt{2}}$ 'ye yakın değerler alırken, beşinci satırın elemanları aynı değerlerin artılı ve eksili değerlerini almaktadır. Kafes parametre belirleme ile elde edilen süzgeçlerin ürettiği BWT matrisinde ise ilk satırın bütün elemanları 0.3536, beşinci satırın bütün elemanları da bu değerlerin artılı veya eksili değerini almaktadır. Buradan çıkan sonuç, kafes parametre belirleme yöntemi kullanıldığında KLT matrislerinin sıfır hata ile elde edilemeyeceği, fakat bir yaklaşımını elde etmenin mümkün olduğudur.

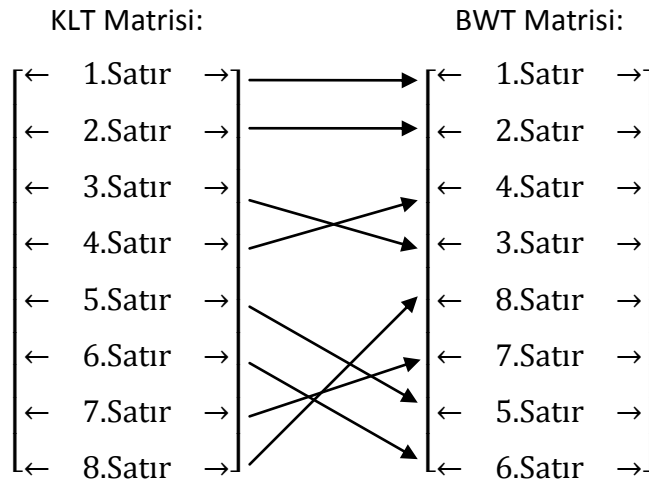
Denklem (4.54) ve (4.55)'teki matrislerde görülen ikinci bir özellik de satırların *ardışıklık* değerleridir. Bir satır için ardışıklık, o satırdaki işaret değişiklikleri sayısının yarısı olarak tanımlanır [3]. Buna göre (4.54) ve (4.55)'teki matrislerin ikisinin de satırlarının ardışıklık değerleri, yukarıdan aşağıya doğru sırasıyla 0, 1/2, 1, 3/2, 2, 5/2, 3 ve 7/2 olmaktadır. BWT matrisinde ise ardışıklık sıralaması kullanılan süzgece göre daha farklı olabilmektedir. Fakat

hangi süzgeç kullanılırsa kullanılsın, BWT matrisinin ilk satırının ardışıklık değeri 0, beşinci satırınki ise 7/2 olmaktadır.

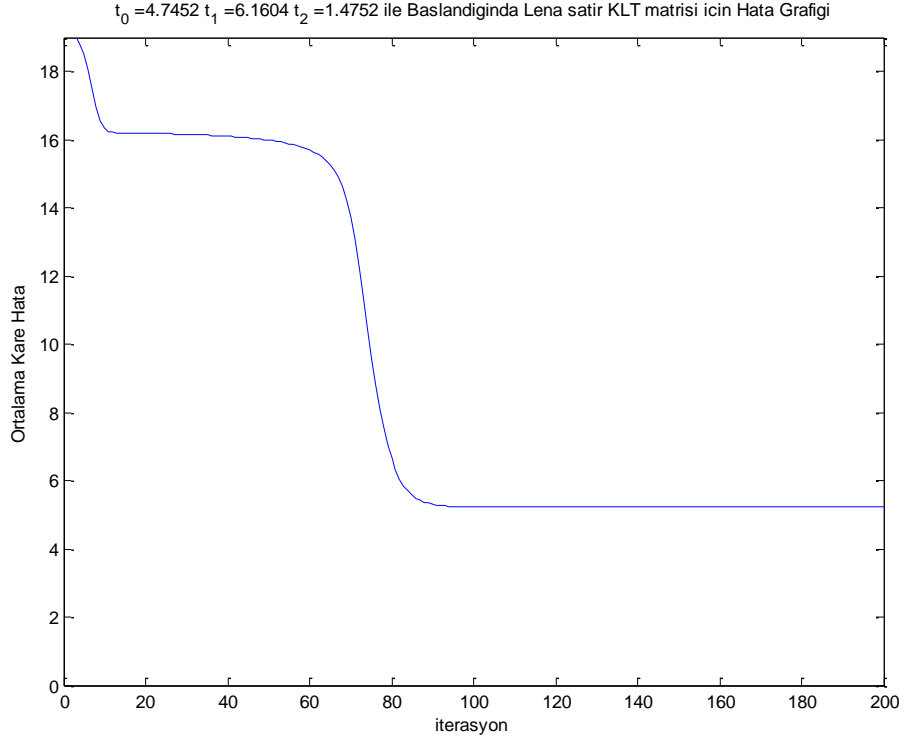
KLT ve BWT matrislerinin satırlarının ardışıklık değerleri birbirinden farklı olduğuna göre, bu iki matrisin satırlarının ardışıklık değerlerini eşleştirerek çözüm aramak daha doğru olacaktır. Bu yüzden kafes parametre belirlemeli en dik iniş yönteminde KLT matrisinin satırlarının sıralaması değiştirilmelidir.

Gerek, Gürcan ve Çetin [39], altbantların enerjilerine göre sıralamasının Gray koduna göre yapılması gerektiğini ifade etmektedirler. Bu, 2-aşamalı BWT sistemi için geçerli olsa da 3-aşamalı BWT sistemi için doğru gözükmemektedir. Dönüşüm matrisleri üzerinde yapılan gözlemler de bunu doğrulamaktadır. Sonuç olarak, satırların sıralamasının Şekil 4.5'teki gibi değiştirilmesinin daha doğru olduğu ve bu durumda BWT ile KLT matrisleri arasındaki en küçük ortalama kare hatanın elde edildiği görülmüştür.

En dik iniş algoritmasında adım büyüklüğü $\mu = 10^{-2}$, kabul edilebilir hata oranı $\varepsilon = 10^{-5}$ alınıp Şekil 4.5'teki satır eşleştirmesi ile birlikte başlangıç açıları rastgele seçilen $t_0 = 4.7452$, $t_1 = 6.1604$ ve $t_2 = 1.4752$ değerleri alındığında ilk 200 iterasyonda elde edilen hata grafiği Şekil 4.6'da gösterilmiştir. Matlab'da yazılan kodun 1000 iterasyon için çalışması, Pentium Core 2 Duo 2.20 GHz işlemcili bir bilgisayarda 1.1 saniye sürmüştür.



Şekil 4.5. KLT ile BWT matrisleri arasındaki satır eşleştirmesi



Şekil 4.6. Lena resminin satırlarına ait KLT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği

Bu deney sonunda kafes parametre belirleme açı değerleri radyan cinsinden şöyle bulunmuştur:

$$\begin{aligned}
 t_0 &= 6.2865 (= 0.0033) \\
 t_1 &= 6.2796 (= -0.0036) \\
 t_2 &= 0.0026.
 \end{aligned} \tag{4.56}$$

Bu açı değerlerine karşılık gelen süzgeçler şu şekildedir:

$$\mathbf{h}_0 = \begin{bmatrix} 0.7087 \\ 0.0023 \\ -0.0018 \\ -0.0025 \\ 0.0025 \\ 0.0019 \\ -0.0023 \\ 0.7055 \end{bmatrix}, \quad \mathbf{h}_1 = \begin{bmatrix} 0.7055 \\ 0.0023 \\ 0.0019 \\ -0.0025 \\ -0.0025 \\ 0.0018 \\ 0.0023 \\ -0.7087 \end{bmatrix}. \tag{4.57}$$

Bu süzgeçlerin ürettiği BWT matrisi, Şekil 4.5'teki satır değişimi geri alındıktan sonra şöyledir:

$$\mathbf{Y} = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.3574 & 0.3584 & 0.3487 & 0.3496 \\ 0.3487 & 0.3496 & -0.3574 & -0.3584 \\ 0.3577 & 0.3494 & -0.3577 & -0.3494 \\ 0.3494 & -0.3577 & -0.3494 & 0.3577 \\ 0.3501 & -0.3492 & -0.3579 & 0.3570 \\ 0.3579 & -0.3570 & 0.3501 & -0.3492 \\ 0.3536 & -0.3536 & 0.3536 & -0.3536 \end{bmatrix} \dots$$

(4.58)

$$\begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ -0.3574 & -0.3584 & -0.3487 & -0.3496 \\ -0.3487 & -0.3496 & 0.3574 & 0.3584 \\ 0.3577 & 0.3494 & -0.3577 & -0.3494 \\ \dots & 0.3494 & -0.3577 & -0.3494 & 0.3577 \\ -0.3501 & 0.3492 & 0.3579 & -0.3570 \\ -0.3579 & 0.3570 & -0.3501 & 0.3492 \\ 0.3536 & -0.3536 & 0.3536 & -0.3536 \end{bmatrix}$$

Denklem (4.54)'teki KLT matrisiyle (4.58)'deki BWT matrisleri karşılaştırıldığında, matris elemanlarının birbirine pek yakın olmadığı görünmektedir. Fakat algoritma farklı başlangıç noktalarıyla çalıştırıldığında ya aynı sonuca ulaşılmakta, ya da daha fazla hata veren sonuçlar bulunmaktadır. Yukarıdaki sonuçları veren başlangıç noktaları seçildiğinde hata her durumda

$$e = 5.2200 \quad (4.59)$$

çıkılmaktadır. Buna göre (4.54)'te verilen satır KLT matrisini üreten BWT süzgeçlerini bulmak mümkün değildir. En küçük ortalama kare hata bağlamında bulunabilecek en yakın matris (4.58)'deki matristir ve bu matrisi üreten süzgeçler (4.57)'deki süzgeçlerdir.

Aynı işlem (4.55)'te verilen ve Lena resminin sütunları üzerinden hesaplanan KLT matrisi üzerinde $t_0 = 2.2936$, $t_1 = 1.9424$ ve $t_2 = 0.7597$ başlangıç açıları için tekrarlandığında optimum noktadaki hata

$$e = 5.2653 \quad (4.60)$$

olarak bulunurken kafes parametre belirleme açıları

$$\begin{aligned} t_0 &= 3.0707 \\ t_1 &= 3.0631 \end{aligned} \quad (4.61)$$

$$t_2 = 6.4186 (= 0.1354)$$

olur.

Bu açılara karşılık gelen BWT süzgeçleri,

$$\mathbf{h}_0 = \begin{bmatrix} 0.6870 \\ -0.0488 \\ -0.0927 \\ -0.0477 \\ 0.0627 \\ 0.0972 \\ 0.0501 \\ 0.7063 \end{bmatrix}, \quad \mathbf{h}_1 = \begin{bmatrix} 0.7063 \\ -0.0501 \\ 0.0972 \\ -0.0627 \\ -0.0477 \\ 0.0927 \\ -0.0488 \\ -0.6870 \end{bmatrix} \quad (4.62)$$

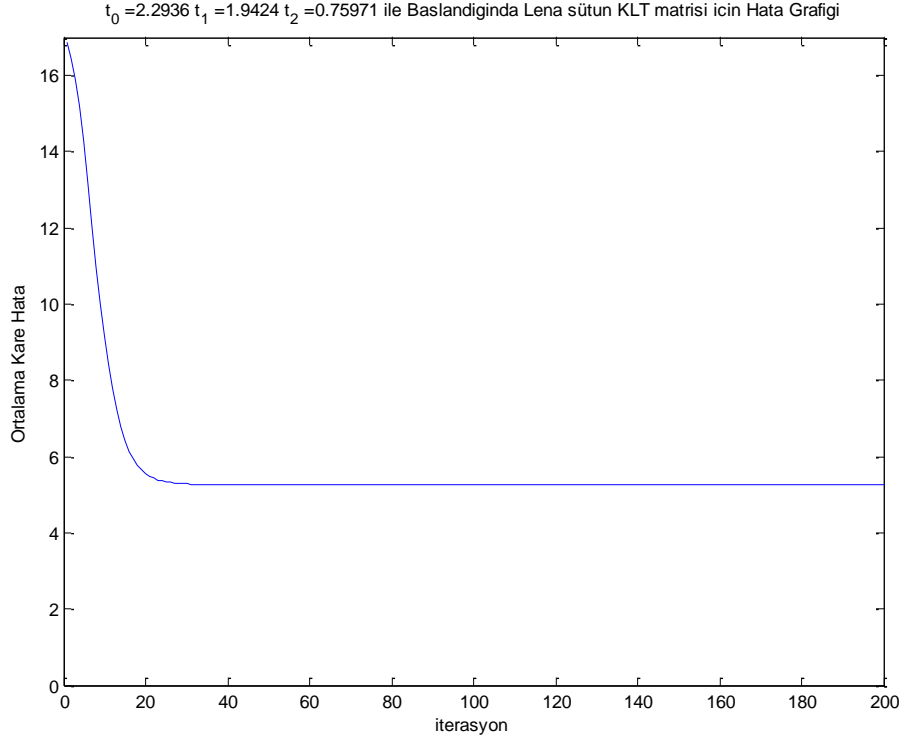
şeklinde ve bu süzgeçlerin ürettiği BWT matrisi, Şekil 4.5'teki satır değişiklikleri geri alındıktan sonra,

$$\mathbf{Y} = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.2881 & 0.3594 & 0.3494 & 0.4071 \\ 0.3494 & 0.4071 & -0.2881 & -0.3594 \\ 0.3961 & 0.3051 & -0.3961 & -0.3051 \\ 0.3051 & -0.3961 & -0.3051 & 0.3961 \\ 0.2435 & -0.1893 & -0.4854 & 0.4114 \\ 0.4854 & -0.4114 & 0.2435 & -0.1893 \\ 0.3536 & -0.3536 & 0.3536 & -0.3536 \end{bmatrix} \dots \quad (4.63)$$

$$\begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ -0.2881 & -0.3594 & -0.3494 & -0.4071 \\ -0.3494 & -0.4071 & 0.2881 & 0.3594 \\ 0.3961 & 0.3051 & -0.3961 & -0.3051 \\ \dots & 0.3051 & -0.3961 & -0.3051 & 0.3961 \\ -0.2435 & 0.1893 & 0.4854 & -0.4114 \\ -0.4854 & 0.4114 & -0.2435 & 0.1893 \\ 0.3536 & -0.3536 & 0.3536 & -0.3536 \end{bmatrix}$$

şeklinde hesaplanır. Bu deneyin ilk 200 iterasyonundaki hata grafiği Şekil 4.7'de gösterilmiştir.

Denklem (4.63)'teki BWT matrisi, (4.55)'te verilen, Lena resminin sütunlarından elde edilen KLT matrisine, kafes parametre belirleme yöntemiyle en küçük ortalama kare hata bağlamında elde edilebilecek en yakın dönüşüm matrisidir. Farklı başlangıç noktaları kullanıldığında da daha yakın bir matris bulunamamaktadır.



Şekil 4.7. Lena resminin sütunlarına ait KLT matrisini üreten süzgeç bankasının kafes parametre belirlemeli en dik iniş yöntemiyle bulunmasındaki hata grafiği

4.3. BWT Ters Çevrimi

Bölüm 4.2’de anlatıldığı üzere, verilen bir tekil dönüşüm matrisini üretmek için kafes parametre belirleme yöntemiyle $N = 2^l$ uzunluklu süzgeçler için parametre sayısı $(N - 2)/2$ ’ye indirilebilir. Bu parametrelerle oluşturulan süzgeçlerin ürettiği BWT matrisiyle, verilen KLT matrisinin elemanları eşleştirilerek parametrelerin değeri analitik olarak hesaplanabilir. Bu yolla geliştirilen yöntem *BWT ters çevrimi* (*BWT Inversion*) adı verilmiştir.

4×4 boyutlu KLT matrisini üretmek için $N = 4$ uzunluklu süzgeçler kullandığımız durumu ele alalım. Bu durumda alçak-geçiren \mathbf{h}_0 süzgeci, bir α parametresine bağlı olarak (4.32)’deki formda ifade edilir. Buna karşılık gelen \mathbf{h}_1 yüksek-geçiren süzgeci ise şu şekilde olur:

$$\begin{aligned}
\mathbf{h}_1(1) &= \frac{1 - \cos \alpha - \sin \alpha}{2\sqrt{2}} \\
\mathbf{h}_1(2) &= \frac{-1 - \cos \alpha + \sin \alpha}{2\sqrt{2}} \\
\mathbf{h}_1(3) &= \frac{1 + \cos \alpha + \sin \alpha}{2\sqrt{2}} \\
\mathbf{h}_1(4) &= \frac{-1 + \cos \alpha - \sin \alpha}{2\sqrt{2}}.
\end{aligned} \tag{4.64}$$

Denklem (4.32) ve (4.64)'te verilen süzgeçleri BWT sistemine uygulandığında, dönüşüm matrisi şu şekilde bulunur:

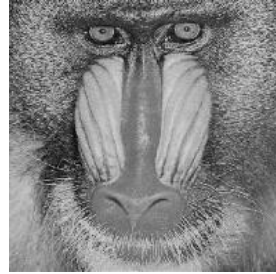
$$\mathbf{T} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ -B & -A & B & A \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -A & B & A & -B \end{bmatrix}, \tag{4.65}$$

$$A = \frac{\cos \alpha + \sin \alpha}{2}, B = \frac{\cos \alpha - \sin \alpha}{2}.$$

Bu matrisin KLT matrisleriyle eşleştirilmesi gerekmektedir. Bu amaçla Şekil 4.8'de gösterilen Lena, Mandril, Peppers ve Bridge test resimlerinin satır ve sütunlarından elde edilen 4×4 boyutundaki KLT matrislerini inceleyelim.



(a)



(b)



(c)



(d)

Şekil 4.8. Test resimleri (a) Lena (b) Mandril (c) Peppers (d) Bridge

Lena resminin satır ve sütunlarından oluşturulan KLT matrisleri şu şekildedir:

$$\mathbf{A}_{Lena}^{(satır)} = \begin{bmatrix} 0.4983 & 0.5011 & 0.5016 & 0.4990 \\ 0.6525 & 0.2699 & -0.2648 & -0.6567 \\ -0.5060 & 0.5014 & 0.4958 & -0.4966 \\ 0.2642 & -0.6516 & 0.6576 & -0.2705 \end{bmatrix}$$

$$\mathbf{A}_{Lena}^{(sütun)} = \begin{bmatrix} 0.4991 & 0.5012 & 0.5009 & 0.4989 \\ 0.6525 & 0.2720 & -0.2720 & -0.6529 \\ -0.4998 & 0.4957 & 0.5023 & -0.5022 \\ 0.2745 & -0.6551 & 0.6503 & -0.2694 \end{bmatrix}. \quad (4.66)$$

Mandril resminin satır ve sütunlarından oluşturulan KLT matrisleri şu şekildedir:

$$\mathbf{A}_{Mandril}^{(satır)} = \begin{bmatrix} 0.4999 & 0.5003 & 0.5023 & 0.4975 \\ 0.7287 & 0.1924 & -0.4016 & -0.5202 \\ -0.2565 & 0.3470 & 0.5876 & -0.6845 \\ 0.3915 & -0.7696 & 0.4911 & -0.1152 \end{bmatrix}$$

$$\mathbf{A}_{Mandril}^{(sütun)} = \begin{bmatrix} 0.4985 & 0.5010 & 0.5015 & 0.4990 \\ 0.6175 & 0.3228 & -0.2787 & -0.6609 \\ -0.5371 & 0.5031 & 0.4930 & -0.4641 \\ 0.2858 & -0.6258 & 0.6540 & -0.3145 \end{bmatrix}. \quad (4.67)$$

Peppers resminin satır ve sütunlarından oluşturulan KLT matrisleri şu şekildedir:

$$\mathbf{A}_{Peppers}^{(satır)} = \begin{bmatrix} 0.4994 & 0.5008 & 0.5006 & 0.4992 \\ 0.6629 & 0.2502 & -0.2557 & -0.6577 \\ -0.4998 & 0.5076 & 0.4909 & -0.5015 \\ 0.2477 & -0.6549 & 0.6656 & -0.2582 \end{bmatrix}$$

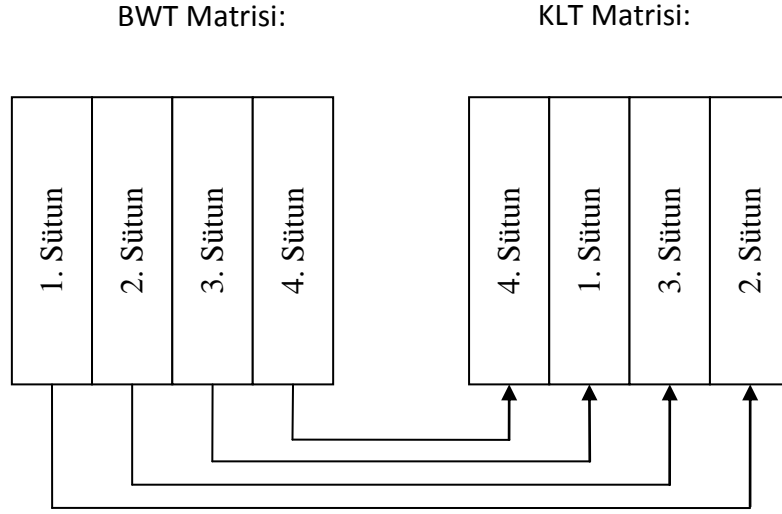
$$\mathbf{A}_{Peppers}^{(sütun)} = \begin{bmatrix} 0.4995 & 0.5004 & 0.5004 & 0.4997 \\ 0.6438 & 0.2725 & -0.2440 & -0.6721 \\ -0.5138 & 0.4793 & 0.5193 & -0.4864 \\ 0.2683 & -0.6676 & 0.6483 & -0.2490 \end{bmatrix}. \quad (4.68)$$

Bridge resminin satır ve sütunlarından oluşturulan KLT matrisleri şu şekildedir:

$$\mathbf{A}_{Bridge}^{(satır)} = \begin{bmatrix} 0.5000 & 0.5016 & 0.5007 & 0.4977 \\ 0.6240 & 0.3190 & -0.2990 & -0.6476 \\ -0.5186 & 0.4996 & 0.4978 & -0.4834 \\ 0.3028 & -0.6301 & 0.6420 & -0.3150 \end{bmatrix}$$

$$\mathbf{A}_{Bridge}^{(sütun)} = \begin{bmatrix} 0.4999 & 0.5015 & 0.5008 & 0.4978 \\ 0.6594 & 0.2762 & -0.3133 & -0.6252 \\ -0.4890 & 0.5334 & 0.4629 & -0.5120 \\ 0.2762 & -0.6227 & 0.6609 & -0.3149 \end{bmatrix}. \quad (4.69)$$

Bu KLT matrislerinin hepsinin satırlarındaki ardışıklıklar yukarıdan aşağıya şu sıradadır: 0, 1/2, 1, 3/2. Ayrıca KLT matrislerinin birinci ve üçüncü satırlarındaki elemanlar ± 0.5 değerlerine yakındır.



Şekil 4.9. Satırların ardışıklıklarının eşleştirilmesi için BWT ile KLT matrisleri arasında yapılması gereken sütun değişiklikleri

Denklem (4.65)'te verilen BWT matrisi \mathbf{T} 'nin birinci ve üçüncü satırlarının değerleri ± 0.5 değerlerini almasına rağmen ardışıklıkları 0 ve $3/2$ olmaktadır. O halde \mathbf{T} matrisi ile KLT matrislerinin satırlarının ardışıklıklarını eşleştirmek için \mathbf{T} matrisinin sütunları değiştirilmelidir. Sütunların yeni sıralamasının belirlenmesi için \mathbf{T} matrisinin üçüncü satırındaki değerler ile ikinci ve üçüncü satırdaki değerler işaretleriyle birlikte kullanılabilir. Buna göre \mathbf{T} matrisinin sütunları 4-1-3-2 şeklinde sıralanmalıdır. \mathbf{T} matrisi ile KLT matrisleri arasındaki sütun eşleştirmesi, Şekil 4.9'da gösterilmiştir.

Denklem (4.65)'teki \mathbf{T} matrisinin sütunları bu şekilde değiştirildiğinde şu şekli alır:

$$\mathbf{T}^{(\text{sıralı})} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ A & -B & B & -A \\ -0.5 & 0.5 & 0.5 & -0.5 \\ -B & -A & A & B \end{bmatrix}, \quad (4.70)$$

$$A = \frac{\cos \alpha + \sin \alpha}{2}, B = \frac{\cos \alpha - \sin \alpha}{2}.$$

BWT matrisinin birinci ve üçüncü satırları, α parametresinin değeri ne olursa olsun hep ± 0.5 olduğundan bu satırlardaki değerlerin KLT matrisindeki karşılık gelen değerlere tam olarak uydurulması mümkün olmayacaktır. Fakat

ikinci ve dördüncü satırlardaki değerler kullanılarak ortalama kare hatayı minimum yapacak şekilde α katsayısının bulunması mümkündür. Bu amaçla BWT matrisi $\mathbf{T}^{(\text{sıralı})}$ ile hedeflenen bir KLT matrisi \mathbf{A} arasındaki toplam kare hata e 'yi şu şekilde yazalım:

$$\begin{aligned}
e &= (A - a_{21})^2 + (-B - a_{22})^2 + (B - a_{23})^2 \\
&\quad + (-A - a_{24})^2 + (-B - a_{41})^2 \\
&\quad + (-A - a_{42})^2 + (A - a_{43})^2 + (B - a_{44})^2 \\
&= (A - a_{21})^2 + (B + a_{22})^2 + (B - a_{23})^2 + (A + a_{24})^2 \\
&\quad + (B + a_{41})^2 + (A + a_{42})^2 + (A - a_{43})^2 \\
&\quad + (B - a_{44})^2.
\end{aligned} \tag{4.71}$$

Buradaki a_{ij} , $i, j = 1, 2, 3, 4$ değerleri, A matrisinin i 'inci satır ve j 'inci sütundaki elemanlarını göstermektedir.

Denklem (4.70)'te verilen A ve B ifadelerinin α 'ya göre türevleri $\frac{dA}{d\alpha} = B$ ve $\frac{dB}{d\alpha} = -A$ olduğundan, (4.72)'de verilen hatanın α 'ya göre türevi,

$$\begin{aligned}
\frac{de}{d\alpha} &= 2B(A - a_{21}) - 2A(B + a_{22}) - 2A(B - a_{23}) \\
&\quad + 2B(A + a_{24}) - 2A(B + a_{41}) + 2B(A + a_{42}) \\
&\quad + 2B(A - a_{43}) - 2A(B - a_{44}) \\
&= 2A(-a_{22} + a_{23} - a_{41} + a_{44}) + 2B(-a_{21} + a_{24} + a_{42} - a_{43})
\end{aligned} \tag{4.72}$$

şeklinde hesaplanır. Hatanın minimum olduğu yerde türev sıfır olduğundan,

$$A(a_{23} + a_{44} - a_{22} - a_{41}) + B(a_{24} + a_{42} - a_{21} - a_{43}) = 0 \tag{4.73}$$

olur. Denklem (4.70)'deki A ve B değerleri yerine yazıldığında,

$$\alpha = \tan^{-1} \left(\frac{c_1 + c_2}{c_2 - c_1} \right), \tag{4.74}$$

$$c_1 = a_{23} + a_{44} - a_{22} - a_{41}, \quad c_2 = a_{24} + a_{42} - a_{21} - a_{43}$$

bulunur. Böylece verilen bir KLT matrisine en küçük kare hata bağlamında en yakın BWT matrisini üreten α katsayısı hesaplanmış olur. \square

\tan^{-1} fonksiyonu $-\pi$ ile $+\pi$ arasında bir değer verir. α değerinin 0 ile π arasında çıkmasını istediğimizde fonksiyonun döndürdüğü değere π eklememiz gerekir. Buna göre (4.74)'teki formülü şu şekilde güncelleyebiliriz:

$$\alpha^{(\text{opt})} = \begin{cases} \alpha & , \quad \alpha \geq 0 \\ \pi - \alpha & , \quad \alpha < 0 \end{cases} \tag{4.75}$$

4.3.1. Performans Analizi

Denklem (4.75) ile hesaplanan α açısı, (4.32) ve (4.64)'te yerine yazılarak alçak- ve yüksek-geçiren süzgeçler hesaplanır. Bu süzgeçlerin bir resim üzerindeki kodlama performansını ölçmek için resmin bütün satırları iki aşamalı BWT sisteminden geçirilir ve 4 altsinyal elde edilir. Bu kanalların her birinin varyansı kullanılarak kodlama kazancı şu şekilde hesaplanır [3]:

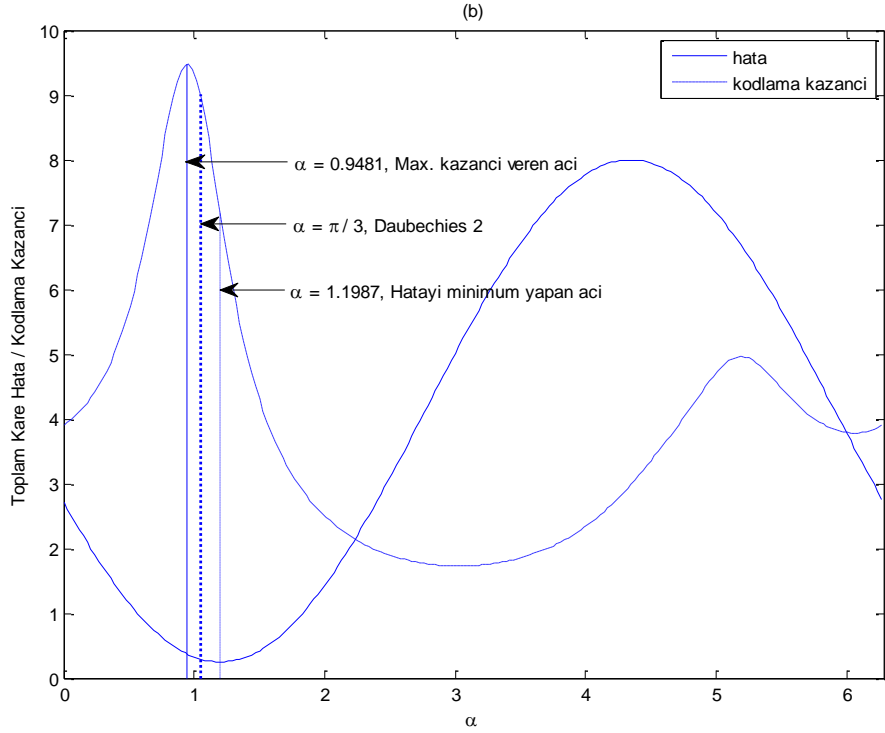
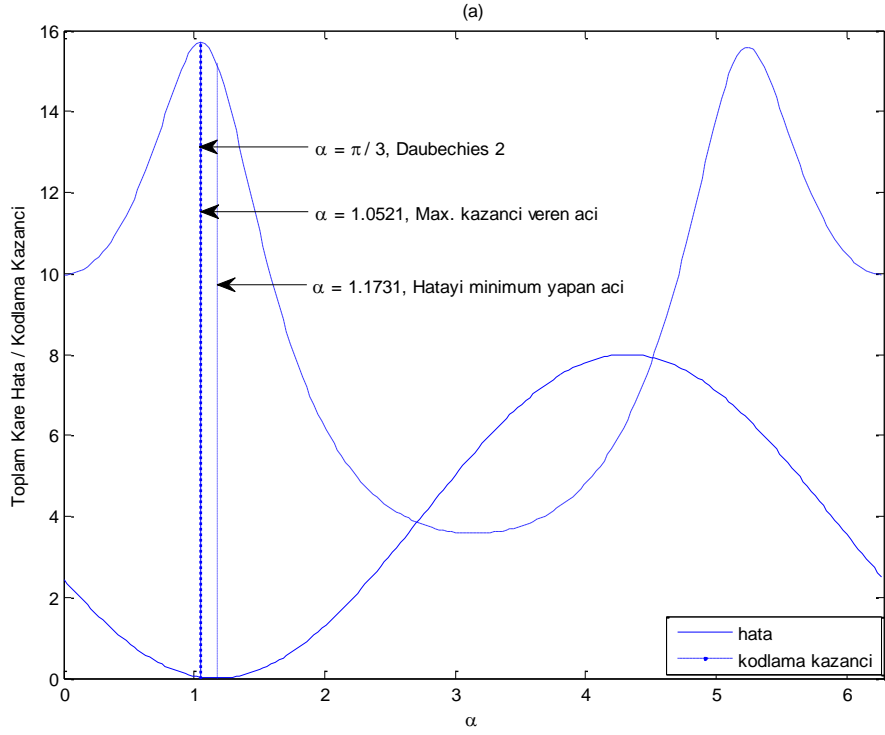
$$G_{TC} = \frac{\frac{1}{4} \sum_{k=1}^4 \sigma_k^2}{(\prod_{k=1}^4 \sigma_k)^{1/4}}. \quad (4.76)$$

Buradaki $\sigma_k, k = 1, 2, 3, 4$ değerleri, elde edilen altsinyallerin varyanslarıdır ve kodlama kazancı bu varyansların aritmetik ortalamasının geometrik ortalamasına bölünmesiyle hesaplanır. Kodlama kazancının büyük olması, orijinal sinyaldeki enerjinin büyük kısmının ilk altsinyalde toplandığını; kalan enerjinin de diğer üç altsinyale paylaştırıldığını gösterir. Böyle bir ayrıştırma sonucunda altsinyaller daha fazla sıkıştırılabilirler.

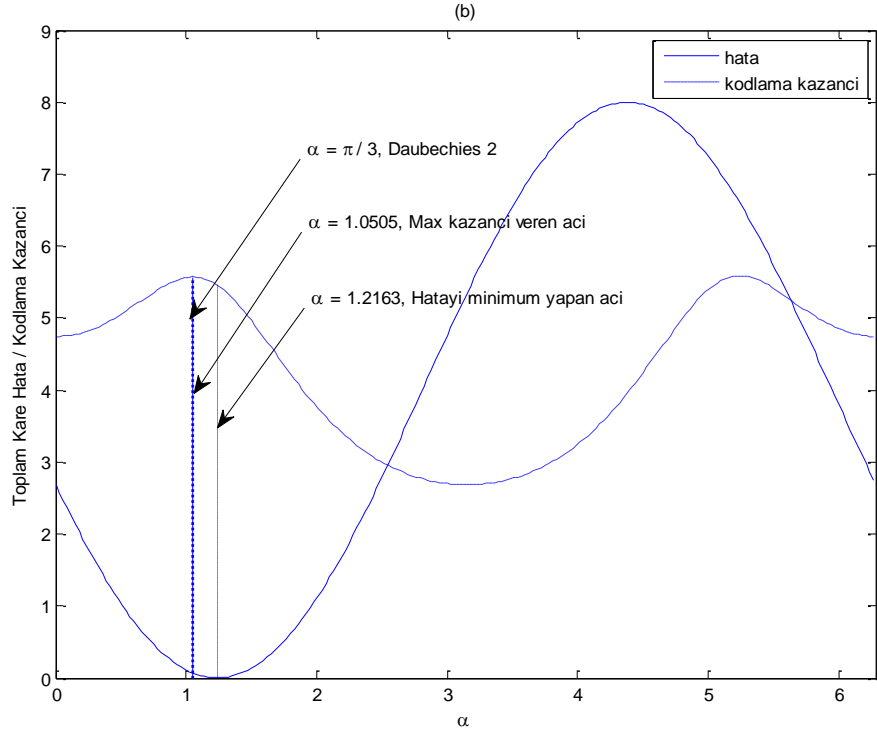
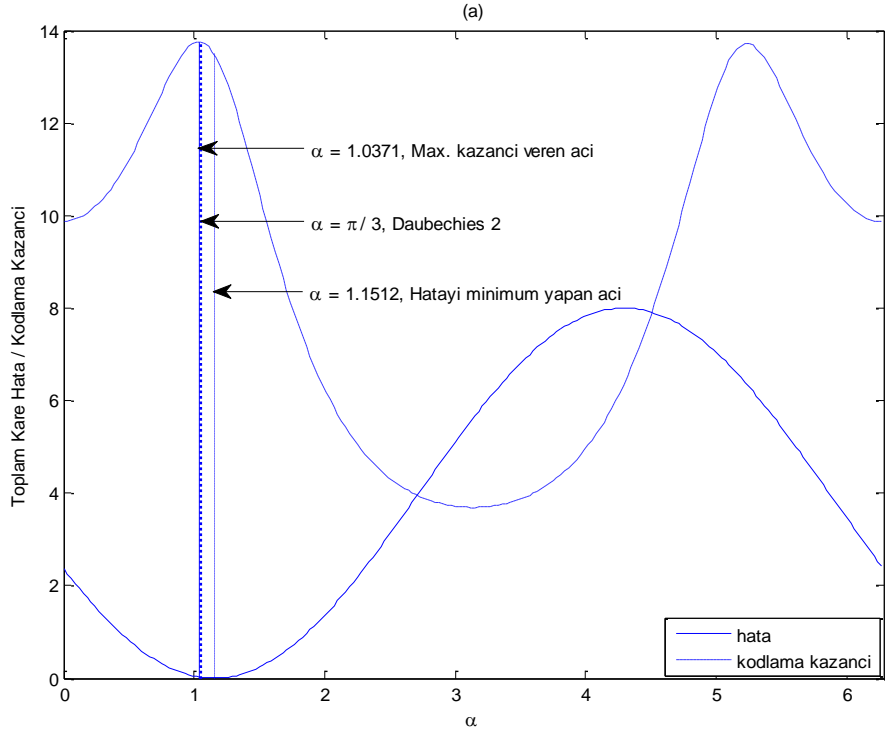
Test resimlerinin satırları üzerinden elde edilen KLT matrisi ile, sütunları 4-1-3-2 sırasıyla sıralanmış BWT matrisi arasındaki toplam kare hatanın grafiği, $0..2\pi$ aralığındaki bütün α açılara bağlı olarak Şekil 4.10 ve Şekil 4.11'de düz çizgilerle; kodlama kazançları da kesikli çizgilerle gösterilmiştir. Grafiklerden, toplam kare hatanın minimum olduğu yerde kodlama kazancının maksimuma yakın olduğu görülmektedir. Yüksek frekanslar açısından zengin Mandril resmi dışında Daubechies-2 süzgeçleri maksimuma yakın kodlama kazancı vermektedir.

Çizelge 4.4'te test resimleri için toplam kare hatayı minimum yapan açılarla kodlama kazancını maksimum yapan açılar ve bu açılara karşılık gelen kodlama kazançları listelenmiştir. Toplam kare hatayı minimum yapan açılar (4.75) ile hesaplanmıştır. BWT ters çevrimi ile bulunan kodlama kazançları, elde edilebilecek maksimum kodlama kazancına yakın değerler almaktadır.

Çizelge 4.5'te ise test resimleri için Daubechies-2 süzgeçleriyle elde edilen kodlama kazançları verilmiştir. Bu değerlerin de maksimum kodlama kazancına yakın olduğu görülmektedir.



Şekil 4.10. Lena ve Mandril test resimlerinin satırlarından elde edilen kodlama kazancı ve toplam kare hata grafikleri (a) Lena (b) Mandril



Şekil 4.11. Peppers ve Bridge test resimlerinin satırlarından elde edilen kodlama kazancı ve toplam kare hata grafikleri (a) Peppers (b) Bridge

Çizelge 4.4. Test resimlerinde toplam kare hatanın minimum, kodlama kazancının maksimum olduğu açı değerleri ve karşılık gelen kodlama kazançları

Resim	Toplam kare hatayı minimum yapan açı (üstte) ve kodlama kazancı (altta)	Kodlama kazancını maksimum yapan açı (üstte) ve kodlama kazancı (altta)
Lena (satır)	1.1731 15.1908	1.0521 15.7026
Lena (sütun)	1.1802 21.0076	1.0524 21.8264
Mandril (satır)	1.1987 7.1926	0.9481 9.4769
Mandril (sütun)	1.2246 4.3168	1.0359 4.6043
Peppers (satır)	1.1512 13.5035	1.0371 13.7553
Peppers (sütun)	1.1597 13.0679	1.0372 13.2861
Bridge (satır)	1.2377 5.4588	1.0505 5.5726
Bridge (sütun)	1.2163 5.1669	1.0547 5.2838

Çizelge 4.5. Test resimleri üzerinde Daubechies-2 süzgeçleriyle elde edilen kodlama kazançları

Resim	db2 süzgeçleriyle elde edilen kodlama kazançları
Lena (satır)	15.7017
Lena (sütun)	21.8251
Mandril (satır)	9.0181
Mandril (sütun)	4.6032
Peppers (satır)	13.7534
Peppers (sütun)	13.2847
Bridge (satır)	5.5726
Bridge (sütun)	5.2835

Çizelge 4.6. Test resimlerinin satır ve sütunlarından oluşturulan KLT matrisleri için elde edilen alçak-geçiren süzgeç katsayıları. İlk satır: Daubechies-2 süzgecinin katsayıları; Diğer satırlar: BWT ters çevrimi ile elde edilen ve kodlama kazancının maksimum olduğu noktada elde edilen süzgeç katsayıları.

Resim	$h_0(1)$	$h_0(2)$	$h_0(3)$	$h_0(4)$
Daub-2	0.4830	0.8365	0.2241	-0.1294
Lena (satır) (BWT Ters Çev.)	0.5426	0.8164	0.1645	-0.1093
Lena (satır) (Max. Kod. Kaz.)	0.4853	0.8359	0.2218	-0.1288
Lena (sütun) (BWT Ters Çev.)	0.5549	0.8151	0.1612	-0.1080
Lena (sütun) (Max. Kod. Kaz.)	0.4855	0.8358	0.2216	-0.1287
Mandrill (satır) (BWT Ters Çev.)	0.5544	0.8114	0.1527	-0.1043
Mandrill (satır) (Max. Kod. Kaz.)	0.4345	0.8470	0.2726	-0.1398
Mandrill (sütun) (BWT Ters Çev.)	0.5661	0.8061	0.1410	-0.0990
Mandrill (sütun) (Max. Kod. Kaz.)	0.4775	0.8379	0.2296	-0.1308
Peppers (satır) (BWT Ters Çev.)	0.5324	0.8205	0.1747	-0.1134
Peppers (satır) (Max. Kod. Kaz.)	0.4781	0.8378	0.2290	-0.1307
Peppers (sütun) (BWT Ters Çev.)	0.5364	0.8189	0.1707	-0.1118
Peppers (sütun) (Max. Kod. Kaz.)	0.4781	0.8378	0.2290	-0.1307
Bridge (satır) (BWT Ters Çev.)	0.5721	0.8033	0.1350	-0.0962
Bridge (satır) (Max. Kod. Kaz.)	0.4846	0.8361	0.2225	-0.1290
Bridge (sütun) (BWT Ters Çev.)	0.5624	0.8079	0.1447	-0.1007
Bridge (sütun) (Max. Kod. Kaz.)	0.4866	0.8355	0.2205	-0.1284

Sonuç olarak BWT ters çevrimiyle elde edilen süzgeçlerin, Daubechies-2 ve maksimum kodlama kazancı veren süzgeçlere kabul edilebilir seviyede yakın kodlama kazancı verdiği söylenebilir.

Çizelge 4.6'da, test resimlerinin satırlarından ve sütunlarından elde edilen KLT matrisleri için BWT ters çevrimi ile elde edilen alçak-geçiren süzgecin katsayıları ile birlikte maksimum kodlama kazancı veren açının ürettiği alçak-geçiren süzgeç katsayıları verilmiştir. Daubechies-2 süzgecinin katsayıları sabit olduğundan bu katsayılar karşılaştırma yapabilmek için tablonun başına ayrı bir satır olarak eklenmiştir. Bu çizelgede, farklı yöntemlerle elde edilen süzgeç katsayılarının birbirlerine çok yakın oldukları görülmektedir.

Çizelge 4.7’de Daubechies-2, BWT ters çevrimi ve maksimum kodlama kazancı veren nokta kullanılarak bulunan süzgeçlerle altbant ayrıştırması yapıldığında elde edilen kanal varyansları verilmiştir. Bu kanal varyanslarına bakıldığında, üç durumda da kanalların hemen hemen aynı miktarlarda bilgi saklayacak şekilde ayrıştırıldığı görülmektedir. Bu da kodlama sonunda elde edilecek kazançların birbirine yakın olacağına işaret etmektedir.

Sonuç olarak, BWT ters çevrimi yöntemiyle, maksimum kodlama kazancına yakın bir sonucun elde edildiği ve altbant ayrıştırması sonunda bulunan kanalların enerjilerinin çoğunluğunun ilk kanalda toplanmasıyla yüksek kodlama oranlarına ulaşılabileceği söylenebilir. BWT ters çevrimi, Daubechies yaklaşımından farklı bir yöntemle süzgeçler üretmekte ve üretilen süzgeçler yüksek kodlama kazancı vermektedir.

4.4. 8×8 Boyutunda BWT Ters Çevrimi

4×4 boyutunda yapılan işlemler, benzer şekilde 8×8 boyutlu KLT matrisine en yakın BWT matrisini bulmak için tekrarlanabilir. 8×8 durumuna geçildiğinde serbest kafes parametre sayısı 1’den 3’e çıkar ve alçak-geçiren süzgecin katsayıları, t_0 , t_1 ve t_2 kafes parametreleri cinsinden (4.37)’deki gibi yazılır. Yüksek-geçiren süzgeç katsayıları ise, alçak-geçiren süzgecin katsayılarının tersten sıralanıp çift indisli katsayılarının işaretlerinin değiştirilmesiyle elde edilir. Parametrik olarak ifade edilen bu süzgeç bankası, 3-aşamalı blok dalgacık sisteminden geçirildiğinde 8×8 boyutlu BWT matrisi, kafes parametreleri cinsinden ifade edilir. Bu matris, burada ifade edilemeyecek kadar uzun olduğundan burada gösterilmeyecektir. Fakat teorik olarak bu parametrik matrisin KLT matrisiyle eşleştirilip iki matris arasındaki toplam kare farkın hesaplanması ve bu farkın kafes parametrelerine göre türevlerinin sıfıra eşitlenerek denklemlerin çözülmesi mümkündür. Pratikte ise bu iş oldukça zor görünmektedir. Çünkü serbest parametre sayısı 1’den 3’e çıktığında, hem matris elemanlarının sayısı 16’dan 64’e çıkmakta, hem de her matris elemanının içerisindeki terimlerin sayısı üstel biçimde artmaktadır.

Çizelge 4.7. BWT ters çevrimi, Daubechies-2 ve maksimum kodlama kazanç noktası yöntemleri ile elde edilen kanal varyansları

Resim	Kanal 1	Kanal 2	Kanal 3	Kanal 4
Lena (satır) (BWT Ters Çev.)	0.1576	0.0022	0.0003	0.0005
Lena (satır) (Daub-2)	0.1576	0.0022	0.0002	0.0005
Lena (satır) (Max. Kod. Kaz.)	0.1576	0.0022	0.0002	0.0005
Lena (sütun) (BWT Ters Çev.)	0.1579	0.0015	0.0002	0.0003
Lena (sütun) (Daub-2)	0.1579	0.0015	0.0001	0.0003
Lena (sütun) (Max. Kod. Kaz.)	0.1579	0.0015	0.0001	0.0003
Mandrill (satır) (BWT Ters Çev.)	0.0890	0.0076	0.0003	0.0007
Mandrill (satır) (Daub-2)	0.0892	0.0078	0.0002	0.0004
Mandrill (satır) (Max. Kod. Kaz.)	0.0892	0.0080	0.0002	0.0003
Mandrill (sütun) (BWT Ters Çev.)	0.0867	0.0084	0.0005	0.0029
Mandrill (sütun) (Daub-2)	0.0868	0.0085	0.0004	0.0028
Mandrill (sütun) (Max. Kod. Kaz.)	0.0868	0.0085	0.0004	0.0028
Peppers (satır) (BWT Ters Çev.)	0.1738	0.0020	0.0005	0.0006
Peppers (satır) (Daub-2)	0.1739	0.0020	0.0005	0.0006
Peppers (satır) (Max. Kod. Kaz.)	0.1739	0.0020	0.0005	0.0006
Peppers (sütun) (BWT Ters Çev.)	0.1748	0.0019	0.0006	0.0008
Peppers (sütun) (Daub-2)	0.1749	0.0018	0.0005	0.0007
Peppers (sütun) (Max. Kod. Kaz.)	0.1749	0.0018	0.0005	0.0007
Bridge (satır) (BWT Ters Çev.)	0.1698	0.0062	0.0015	0.0029
Bridge (satır) (Daub-2)	0.1701	0.0061	0.0015	0.0029
Bridge (satır) (Max. Kod. Kaz.)	0.1702	0.0061	0.0015	0.0029
Bridge (sütun) (BWT Ters Çev.)	0.1674	0.0081	0.0012	0.0034
Bridge (sütun) (Daub-2)	0.1676	0.0082	0.0012	0.0033
Bridge (sütun) (Max. Kod. Kaz.)	0.1676	0.0082	0.0012	0.0022

8×8 boyutunda analitik çözüm bulunmasının zorluğu sebebiyle burada sadece Bölüm 4.2.1’de bahsedilen nümerik yöntemler kullanılacaktır. Analitik çözümün bulunması ise ileriki bir çalışmaya bırakılacaktır.

Bölüm 4.3’te anlatıldığı üzere, KLT ve BWT matrislerinin uygun bir şekilde eşleştirilmesi büyük önem taşımaktadır. O yüzden burada ileriki çalışmalara ışık tutması amacıyla 8×8 boyutunda KLT ve BWT matrislerinin nasıl eşleştirilmesi gerektiği üzerine bazı açıklamalar yapılacaktır.

KLT ve BWT matrislerinin eşleştirilmesi için Bölüm 4.2.1.4’te satırların enerjilerine göre sıralanacak şekilde yerlerinin değiştirilmesi önerilmişti. Burada ise, QMF süzgeç bankalarıyla elde edilebilecek en büyük kodlama kazancının konumuna göre yeni bir eşleştirme önerilecektir. Bu eşleştirmede satırların yanında sütunlar da değiştirilecek, ayrıca daha iyi bir eşleşme sağlamak için BWT matrisinin satırlarının işaretleri de değiştirilecektir.

Denklem (4.37) kullanılarak alçak- ve yüksek-geçiren süzgeçler parametrik olarak ifade edilip 3-aşamalı sistem kullanıldığında, BWT matrisi şu şekilde ifade edilir:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ A & B & -C & -D & -A & -B & C & D \\ -E & -F & E & F & -E & -F & E & F \\ -C & -D & -A & -B & C & D & A & B \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ G & -H & -I & J & -G & H & I & -J \\ -F & E & F & -E & -F & E & F & -E \\ -I & J & -G & H & I & -J & G & -H \end{bmatrix} \quad (4.77)$$

Burada $A, B, C, D, E, F, G, H, I,$ ve J ile gösterilen matris elemanları, her biri t_0, t_1 ve t_2 kafes parametreleri cinsinden ifade edilen fakat burada yazılamayacak kadar uzun terimlerdir.

Denklem (4.54) ve (4.55)’te verilen Lena test resminin satırları ve sütunları üzerinden elde edilen KLT matrisleri incelendiğinde, iki KLT matrisinin de ikinci satırlarının büyükten-küçüğe sıralandığı görülmektedir. Bu durum diğer test resimleri için de geçerli olsa da, Mandrill resminin sütun-KLT, Bridge resminin ise hem satır-KLT hem de sütun-KLT matrislerinin ikinci satırlarının büyükten-küçüğe sıralanmadığı fark edilmiştir. Bahsedilen bu resimler, içlerinde çok miktarda yüksek frekans barındırdıklarından, KLT matrisinin bu resimler için

düşük-frekans bilgisi yerine yüksek-frekans bilgisini (veya şu an kestiremediğimiz başka bir bilgiyi) ön sıralara taşıdığını söyleyebiliriz.

Test resimlerinin çoğunluğunda geçerli olan sıralama göz önüne alınarak, (4.77)'deki BWT matrisinin de ikinci satırındaki elemanların büyükten-küçüğe sıralanacak şekilde değiştirilmesi gerektiği sonucuna varabiliriz. İkinci satırdaki elemanların büyükten-küçüğe sıralanması, ancak **A** matrisinin sütunlarının uygun biçimde değiştirilmesiyle elde edilebilir. **A** matrisinin ikinci satırında *A*, *B*, *C* ve *D* ile gösterilen terimlerin hem artı işaretlileri hem de eksi işaretlileri olduğundan, satırdaki pozitif sayılar büyükten küçüğe sıralanıp ilk dört sütuna yerleştirildiğinde sonda kalan dört sütun da otomatik olarak belirlenmiş olur. Örneğin *A*, *B*, *C* ve *D* terimlerinin dördünün de pozitif olduğunu ve bu terimleri arasındaki büyükten küçüğe sıralamanın $A > B > C > D$ şeklinde olduğunu varsayarsak, bu durumda ikinci satırdaki elemanların büyükten küçüğe sıralaması $A, B, C, D, -D, -C, -B, -A$ şeklinde olur.

BWT matrisinin ikinci satırındaki elemanların büyükten-küçüğe sıralaması, kafes parametrelerine bağlı olduğundan, uygun sütun sıralamasına çabucak karar vermek mümkün değildir. Uygun satır sıralamasını bulmak için, QMF süzgeç bankalarıyla elde edilebilecek en yüksek kodlama kazancını veren durumlar incelenmelidir. Yani, en yüksek kodlama kazancının elde edildiği bölgedeki kafes parametreleriyle elde edilen BWT matrislerinin ikinci satırlarının sıralaması bize doğru sütun sıralamasının nasıl olması gerektiği yönünde ipuçları verebilir.

En yüksek kodlama kazancının hangi kafes parametreleriyle elde edildiğini bulmak amacıyla kafes parametrelerine 0 ile 2π aralığında rastgele değerler atanarak bu kafes parametreleriyle elde edilen süzgeçlerin test resimleri üzerinde oluşturdukları kodlama kazançları incelenmiştir. Yüksek kodlama kazancı veren süzgeçlerle oluşturulan BWT matrislerinin ikinci satırları hangi sıralamayla dizildiğinde satırın elemanlarının büyükten-küçüğe sıralandığı araştırılmıştır. Elde edilen sütun sıralamaları Çizelge 4.8'de gösterilmiştir.

Çizelge 4.8. Test resimleri üzerinde en büyük kodlama kazancı veren BWT matrisinin ikinci satırının büyükten-küçüğe sıralamasını veren sütun sıralaması

Resim	Sütun sıralaması
Lena (Satır)	1-8-2-7-3-6-4-5
Lena (Sütun)	1-8-2-7-3-6-4-5
Mandrill (Satır)	1-8-2-7-3-6-4-5
Mandrill (Sütun)	8-1-7-2-6-3-5-4
Peppers (Satır)	1-8-2-7-3-6-4-5
Peppers (Sütun)	1-8-2-7-3-6-4-5
Bridge (Satır)	1-8-2-7-3-6-4-5
Bridge (Sütun)	1-8-2-7-3-6-4-5

Çizelge 4.8'deki verilerden, Mandrill test resminin sütunları üzerinde yapılan altbant ayrıştırması dışındaki bütün durumlarda, BWT matrisinin sütunları 1-8-2-7-3-6-4-5 sırasıyla dizildiğinde ikinci satırın elemanlarının KLT matrisinde olduğu gibi büyükten-küçüğe sıralandığı görülmüştür. O halde, resimlerin büyük çoğunluğu için geçerli olan 1-8-2-7-3-6-4-5 sıralaması, BWT matrisi ile KLT matrisini eşleştirmek için kullanılabilir. Denklem (4.77)'deki BWT matrisinin sütunları bu sıralamaya göre dizildiğinde şu şekle dönüşür:

$$\mathbf{A}^{(ord)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ A & D & B & C & -C & -B & -D & -A \\ -E & F & -F & E & E & -F & F & -E \\ -C & B & -D & A & -A & D & -B & C \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ G & -J & -H & I & -I & H & J & -G \\ -F & -E & E & F & F & E & -E & -F \\ -I & -H & J & G & -G & -J & H & I \end{bmatrix} \quad (4.78)$$

KLT ve BWT matrisleri arasındaki en uygun eşleştirmeyi bulabilmek için şimdi de KLT matrislerinin ve sütunları sıralanmış BWT matrisinin satırlarının ardışıklıklarını inceleyelim.

Test resimlerine ait KLT matrisleri incelendiğinde, Mandrill ve Peppers resimlerinin satır-KLT matrisleri haricinde kalan bütün KLT matrislerinin satırlarının ardışıklık değerlerinin 0, 1/2, 2/2, 3/2, 4/2, 5/2, 6/2 ve 7/2 olduğu

görülmüştür. Mandrill ve Peppers resimlerinin satır-KLT matrislerinin ise sadece son satırlarının ardışıklıkları farklılık göstermektedir. Mandrill satır-KLT matrisinin son satırının ardışıklık değeri $5/2$ olurken, Peppers satır-KLT matrisinin son satırının ardışıklık değeri $6/2$ olmaktadır. Buna göre, BWT matrisinin satırlarının, ardışıklık değerlerine göre küçükten-büyüğe sıralanması gerektiği sonucuna varılabilir.

Denklem (4.78)'deki sütun-sıralı BWT matrisinin sadece birinci ve beşinci satırların ardışıklık değerleri kesin olarak söylenebilir. Bu satırların ardışıklıkları sırasıyla $0/2$ ve $4/2$ olduğundan, satırlar ardışıklıklarına göre sıralandığında bu satırların yerleri değişmeyecektir. İkinci satır da sütun-sıralaması esnasında büyükten-küçüğe sıralandığından bu satırın da yeri değişmemelidir. İkinci satırdaki elemanlar, dördüncü satırda da bulduklarından, dördüncü satırın da ardışıklık değeri hesaplanabilir. A , D , B , C terimlerinin hepsinin pozitif olduğu durum göz önüne alındığında, dördüncü satırın ardışıklığının $7/2$ olduğu fark edilmektedir.

Test resimleri üzerinde yapılan incelemeler sonucunda, ardışıklıkların eşleştirilmesi için sadece 4. ve 8. satırların yerlerinin değiştirilmesinin yeterli olduğu görülmüştür. Fakat KLT matrisinin resme has hangi özelliği öne çıkardığına bağlı olarak ardışıklıkları eşleştirmek için uygulanması gereken satır sıralaması farklı olabilir. Üzerinde çalışılan resimler için 4. ve 8. satırların değiştirilmesi yeterli olsa da, genel konuşmak gerektiğinde satırların ardışıklıkları küçükten-büyüğe sıralanacak şekilde satırların yerlerinin değiştirilmesi gerektiği söylenebilir.

BWT matrisinin ikinci satırı büyükten-küçüğe sıralanacak şekilde sütunları dizildikten ve satırların ardışıklıkları küçükten-büyüğe sıralanacak şekilde satırları dizildikten sonra, KLT ve BWT matrislerinin eşleşmelerinde çözülmesi gereken son bir problem daha kalmaktadır. O da satırların işaretlerinin eşleştirilmesidir. Ardışıklıkların eşleştirilmesi, satırlardaki işaret değişikliklerinin sayısını eşleştiriyor olsa da, artı ve eksi işaretli sayıların dizilimlerini eşleştirememektedir. Bu yüzden önce sütunlar, sonra satırlar değiştirilerek oluşturulan BWT matrisinin satırlarının işaretlerinin de KLT matrisinde karşılık gelen satırlara göre

değiştirilmesi gerekmektedir. Bu değişiklik yapıldıktan sonra BWT ve KLT matrisleri tam olarak eşleştirilmiş olur.

Bu noktada; satırların, sütunların ve işaretlerin değiştirilmesinin geçerli bir işlem olup olmadığı ve yapılan bu işlemlerle BWT matrisinin tekilliğinin bozulup bozulmadığına dair bir soru akla gelebilir.

Hatırlanacağı üzere, KLT matrisi, sinyalin özilinti matrisinin özvektörlerinin özdeğerlere göre sıralanmasıyla elde edilmektedir. Yani sinyal içerisinde en ağır basan özellikler ön sıralara alınmakta, fazla bir ağırlığı olmayan özellikler de arka sıralara atılmaktadır. Her sinyalin istatistiksel özellikleri ve içerisinde barındırdığı özellikler farklı olduğundan, her sinyal için ağır basan tek bir özellikten bahsetmek mümkün olamaz. Örneğin Lena resminde düşük frekanslar ağır basmaktayken, Mandrill resminde yüksek frekanslar ağır basmaktadır.

BWT matrisi üretilirken, periyot içerisindeki elemanlarından biri 1, diğerleri 0 olan girdi sinyalleri altbant ayırıştırma sistemine uygulanıyordu. Kullanılan bu girdi vektörleri yan yana yazıldığında birim matris elde edilir. Başka bir deyişle BWT matrisi, birim matrisin süzgeçler tarafından belli bir dönüşüme tabi tutulmasıyla oluşturulur. Dolayısıyla, BWT matrisinin satırlarının veya sütunlarının değiştirilmesi, BWT matrisini oluşturmak için kullanılan birim matrisin satırlarının veya sütunlarının değiştirilmesi anlamına gelir. Birim matristeki 1'lerin yeri, sinyaldeki hangi özelliklerin ön plana çıkarılması gerektiğini göstermektedir. O halde KLT matrisi farklı bir özelliği ön plana çıkardığında, BWT matrisini oluşturan girdi matrisinde o özelliğe karşılık gelen 1'in yeri de değiştirilmelidir. BWT matrisindeki satırların işaretlerinin değiştirilmesi ise, girdi matrisindeki 1'lerin işaretlerinin değiştirilmesi anlamına gelir. Bu da KLT matrisinin bir özelliği arka plana atmasına denk gelmektedir. Buna göre BWT matrisinin satırlarının, sütunlarının ve işaretlerinin değiştirilmesi geçerli ve kabul edilebilir işlemlerdir.

BWT matrisinin tekilliğinin anlatıldığı Bölüm 3.4.2'de elde edilen sonuçlardan biri de, tekillik için gerekli koşulun, BWT matrisini oluşturmak için kullanılan girdi matrisinin de tekil olması gerektiği idi. Girdi matrisinin satırlarının ve sütunlarının yerlerinin veya işaretlerinin değiştirilmesi, girdi matrisinin tekilliğini bozmadığından, BWT matrisinin de tekilliği bozulmaz.

Son olarak, ortaya konan eşleştirmenin performans analizinin yapılması için test resimlerinden elde edilen KLT matrislerine en yakın BWT matrislerini üreten kafes parametreleri en-dik-iniş yöntemi kullanılarak bulunmaya çalışılmıştır. Bu amaçla çok sayıda rastgele başlangıç noktası seçilmiş ve bu noktalardan hatayı yerel minimum yapan noktaya doğru adım adım inilmiştir. Elde edilen her yerel minimumda kodlama kazancı hesaplanmış ve bu kodlama kazancı Daub-4 süzgeç bankasınıninkine karşılaştırılmıştır. Kodlama kazancını maksimum yapan yerel minimumlardaki kafes parametreleri, eşleştirme sonucunda elde edilen toplam kare hata miktarları, bu noktada elde edilen kodlama kazançları ve Daub-4 süzgeç bankasının verdiği kodlama kazançları Çizelge 4.9’da listelenmiştir. BWT matrisiyle KLT matrisinin eşleştirilmesiyle hesaplanan süzgeç bankaları ve Daub-4 süzgeç bankası kullanılarak altbant ayrıştırması yapıldığında elde edilen kanal varyansları da Çizelge 4.10’da gösterilmiştir.

Çizelge 4.9. 8×8 durumunda test resimleri üzerinde BWT ve KLT matrislerinin eşleştirilmeleriyle elde edilen kafes parametreleri, kodlama kazançları ve Daub-4 süzgeç bankasıyla elde edilen kodlama kazançları

Resim	t_0	t_1	t_2	Hata	Kodlama Kazancı	Daub-4 Kodlama Kazancı
Lena (Satır)	0.2050	1.7578	2.3681	1.5799	29.0332	28.1603
Lena (Sütun)	0.1815	1.7548	2.3792	1.9057	42.3720	40.9648
Mandrill (Satır)	0.1076	1.7306	2.4510	6.6210	19.4572	11.1199
Mandrill (Sütun)	0.2639	1.7654	2.3397	0.3251	10.8046	8.7974
Peppers (Satır)	0.1145	1.7522	2.4255	1.6738	21.7710	21.0646
Peppers (Sütun)	3.0985	1.7363	2.5385	1.0668	20.2952	20.0228
Bridge (Satır)	3.2223	1.7654	2.4497	1.2905	7.5605	7.3931
Bridge (Sütun)	1.7719	0.8179	2.7880	0.8227	7.2874	7.1124

Çizelge 4.10. 8×8 durumunda BWT ve KLT matrislerinin eşleştirilmesiyle elde edilen kanal varyanslarının Daub-4 süzgeç bankası ile elde edilen kanal varyanslarıyla karşılaştırılması

Resim	Yöntem	Kanal1	Kanal2	Kanal3	Kanal4	Kanal5	Kanal6	Kanal7	Kanal8
Lena (Satır)	BWT Ters Çevrimi	0.3077 (%95.93)	0.0085 (%2.64)	0.0009 (%0.27)	0.0027 (%0.84)	0.0001 (%0.04)	0.0001 (%0.05)	0.0005 (%0.16)	0.0003 (%0.08)
	Daub-4	0.3143 (%95.83)	0.0089 (%2.72)	0.0009 (%0.27)	0.0028 (%0.84)	0.0001 (%0.03)	0.0002 (%0.05)	0.0005 (%0.17)	0.0003 (%0.09)
Lena (Sütun)	BWT Ters Çevrimi	0.3049 (%97.21)	0.0057 (%1.81)	0.0006 (%0.18)	0.0019 (%0.60)	0.0001 (%0.02)	0.0001 (%0.04)	0.0003 (%0.10)	0.0002 (%0.05)
	Daub-4	0.3114 (%97.12)	0.0060 (%1.89)	0.0006 (%0.18)	0.0019 (%0.60)	0.0001 (%0.02)	0.0001 (%0.04)	0.0003 (%0.10)	0.0002 (%0.06)
Mandrill (Satır)	BWT Ters Çevrimi	0.1598 (%82.74)	0.0187 (%9.67)	0.0045 (%2.33)	0.0095 (%4.92)	0.0000 (%0.01)	0.0000 (%0.02)	0.0005 (%0.24)	0.0001 (%0.07)
	Daub-4	0.1574 (%82.19)	0.0189 (%9.88)	0.0038 (%1.97)	0.0092 (%4.81)	0.0000 (%0.02)	0.0001 (%0.08)	0.0014 (%0.72)	0.0006 (%0.33)
Mandrill (Sütun)	BWT Ters Çevrimi	0.1595 (%80.45)	0.0178 (%8.99)	0.0050 (%2.50)	0.0121 (%6.12)	0.0000 (%0.01)	0.0001 (%0.05)	0.0031 (%1.56)	0.0007 (%0.33)
	Daub-4	0.1610 (%80.06)	0.0182 (%9.06)	0.0062 (%3.07)	0.0125 (%6.21)	0.0001 (%0.04)	0.0001 (%0.07)	0.0023 (%1.13)	0.0007 (%0.37)
Peppers (Satır)	BWT Ters Çevrimi	0.3390 (%96.29)	0.0081 (%2.29)	0.0008 (%0.24)	0.0022 (%0.62)	0.0004 (%0.12)	0.0005 (%0.13)	0.0006 (%0.18)	0.0004 (%0.12)
	Daub-4	0.3363 (%96.06)	0.0086 (%2.45)	0.0009 (%0.26)	0.0023 (%0.66)	0.0005 (%0.13)	0.0004 (%0.12)	0.0006 (%0.17)	0.0005 (%0.13)
Peppers (Sütun)	BWT Ters Çevrimi	0.3450 (%96.64)	0.0064 (%1.79)	0.0010 (%0.29)	0.0022 (%0.62)	0.0005 (%0.13)	0.0005 (%0.14)	0.0009 (%0.24)	0.0005 (%0.15)
	Daub-4	0.3463 (%96.43)	0.0071 (%1.97)	0.0011 (%0.29)	0.0024 (%0.66)	0.0005 (%0.14)	0.0005 (%0.14)	0.0008 (%0.21)	0.0006 (%0.16)
Bridge (Satır)	BWT Ters Çevrimi	0.3209 (%90.37)	0.0150 (%4.22)	0.0041 (%1.16)	0.0074 (%2.09)	0.0011 (%0.31)	0.0014 (%0.40)	0.0033 (%0.92)	0.0018 (%0.52)
	Daub-4	0.3189 (%90.08)	0.0155 (%4.37)	0.0042 (%1.20)	0.0077 (%2.18)	0.0012 (%0.34)	0.0014 (%0.39)	0.0032 (%0.89)	0.0020 (%0.55)
Bridge (Sütun)	BWT Ters Çevrimi	0.3103 (%87.77)	0.0208 (%5.88)	0.0055 (%1.56)	0.0099 (%2.81)	0.0009 (%0.25)	0.0010 (%0.28)	0.0033 (%0.95)	0.0018 (%0.51)
	Daub-4	0.3036 (%87.51)	0.0211 (%6.08)	0.0050 (%1.44)	0.0101 (%2.90)	0.0008 (%0.24)	0.0011 (%0.33)	0.0033 (%0.95)	0.0019 (%0.56)

Çizelge 4.9'dan görüldüğü üzere, BWT ve KLT matrislerinin eşleştirilmesiyle Daub-4 süzgeç bankasıyla elde edilen kodlama kazançlarından daha yüksek kodlama kazançları elde edilmiştir. Çizelge 4.10'daki kanal varyanslarına bakıldığında ise, eşleştirme sonucunda sinyalin enerjisinin büyük bölümünün ilk alt-bantta toplandığı ve Daub-4'ten daha başarılı olduğu görülmektedir.

Eşleştirme sonucunda elde edilen hata miktarının da oldukça küçük olduğu görülmektedir. Buna göre, Bölüm 4.2.1.4'teki satır değişikliğinin yalnız başına kullanılması yerine, sütun ve işaret değişiklikleriyle birlikte kullanılması sayesinde daha küçük hatalar ve daha yüksek kodlama kazançları elde edilebildiği sonucuna varılmaktadır.

BWT ve KLT matrislerinin eşleştirilmesiyle elde edilen en büyük kodlama kazancı, hatanın global minimum olduğu yerde değil, yerel minimum olduğu noktalardan birinde elde edilmektedir. Önerilen BWT Ters Çevrimi yöntemi 8×8 boyutu için analitik olarak gerçekleştirildiğinde, hatanın kafes parametrelerine göre türevinin sıfır olduğu bütün noktaları bulabileceğinden, burada bulunan yerel minimumlar da analitik yöntemle hesaplanabilecektir. Öyleyse bütün yerel minimumlar bulunup, bütün bu yerel minimumlardaki kodlama kazançları hesaplanırsa en yüksek kodlama kazancını veren QMF süzgeç bankası bulunabilir.

Kodlama kazancının maksimum olduğu yerin, hatanın yerel minimumlarından birinde ortaya çıkması, KLT matrisinin sinyaldeki değişik özellikleri ön sıralara almasıyla ve en yüksek kodlama kazancının elde edilebilmesi için sinyaldeki özelliklere uygulanması gereken kısıtlamalarla açıklanabilir. Örneğin, en yüksek kodlama kazancının elde edilebilmesi için kafes parametrelerinin belirli bir aralıkta bulunmaları koşulu varsa, bu aralık dışındaki noktalarda hata daha küçük değer alıyor olsa bile en yüksek kodlama kazancını vermiyor olabilir. Fakat en yüksek kodlama kazancının, hatanın eğer noktalarında çıkmayıp yerel minimumlardan birinde çıkması, ileriki çalışmalar açısından önemli bir sonuç olarak değerlendirilebilir.

4×4 ve 8×8 durumlarında elde edilen sonuçlar karşılaştırıldığında, boyut arttıkça BWT Ters Çevrimi yöntemiyle elde edilen sonuçların daha başarılı olmaya başladığı görülmektedir. Çünkü boyut arttıkça serbest parametre sayısı da

artmakta ve sinyalin barındırdığı bilgilerden daha fazlası üzerinde işlem yapılabilmektedir. Buna göre BWT Ters Çevrimi yönteminin gerçek gücünü, daha yüksek boyutlarda daha net biçimde gösterebileceğini söyleyebiliriz. Bununla birlikte, boyut arttıkça analitik işlemler oldukça güçleşmekte ve daha fazla işlem gücü gerektirmektedir.

5. SONUÇLAR

Bu tezde KLT ve BWT matrislerinin tekiliğinden yola çıkılarak, verilen bir KLT matrisine en yakın BWT matrisini üreten süzgeç katsayıları bulunmaya çalışılmıştır.

Süzgeç bankası tasarımı için literatürde çok çeşitli tasarlama yöntemleri kullanılmaktadır. Örneğin Bernstein QMF-Wavelet süzgeç bankası, Bernstein polinomlarını kullanarak frekans tepkisindeki dalgalıklanmaları azaltmayı; Johnston QMF [40] süzgeç bankası ise dalgalıklanma enerjisiyle durdurma bandı enerjisinden türetilen bir fonksiyonu minimize etmeyi hedefler. Daubechies süzgeçleri ise frekans tepkisinin $\omega = 0$ ve $\omega = \pi$ frekanslarında maksimum izgesel düzlük özelliğine sahip olmasını amaçlar. Bu sebepten Daubechies süzgeçleriyle yapılan altbant ayrıştırmasında sinyalin düşük frekanslarının büyük bir kısmı ilk kanalda toplanır ve bu da kodlama kazancının yüksek çıkmasını sağlar. Daubechies gibi maksimum izgesel düzlük özelliğini taşıyan, Binomial [41] ve Bernstein QMF-Wavelet [42] gibi başka süzgeç bankası aileleri de vardır.

Bu çalışmada ise başlangıç noktası olarak optimum dönüşüm kodlama yöntemi KLT ile üretilen dönüşüm matrisi ele alınmış ve bu matrisi üreten BWT süzgeçleri bulunmaya çalışılmıştır. Bu bakımdan yukarıda anlatılan süzgeç tasarlama yöntemlerinden daha farklı bir yöntem denenmiş ve her sinyal için kullanılabilir sabit bir süzgeç bankası yerine, sadece kodlanmaya çalışılan sinyale has süzgeç bankalarının bulunması hedeflenmiştir.

Sinyale has süzgeç bankası tasarımında öncelikle en dik iniş yöntemiyle nümerik çözümler bulunmuş, arkasından kafes parametre belirleme yöntemiyle QMF süzgeçlerin serbest parametre sayısı azaltılarak hem nümerik hem de analitik yöntemlerle süzgeç bankaları oluşturulmuştur.

Kullanılan nümerik yöntemlerde *en dik iniş* algoritması tercih edilmiştir. Bu algoritmanın süzgeç tasarımında kullanılabilmesi için öncelikli olarak, belirlenen bir hata fonksiyonunun parametrelere bağlı olarak ifade edilmesi gerekmektedir. Bu amaçla, verilen bir tekil matrisle BWT matrisi arasındaki bir hata fonksiyonu, alçak-geçiren BWT süzgecinin katsayılarına bağlı olarak matris çarpımları ile ifade edilmiştir. Böylece problem, en dik iniş algoritmasının uygulanabileceği bir

şekle dönüştürülmüştür. Ardından yapılan deneylerde, işlemleri hızlandırmak için optimizasyon işlemleri yapılmadığı halde, kabul edilebilir sürelerde istenen sonuca ulaşılmıştır.

En dik iniş yönteminde, seçilen adım büyüklüğü ve başlangıç noktası büyük önem taşımaktadır. Adım büyüklüğü, olması gerekenden büyük seçildiğinde çözüm atlanabilmekte, küçük seçildiğinde çözüme ulaşılması çok uzun zaman alabilmektedir. Başlangıç noktası yanlış seçildiğinde ise global minimum yerine yerel minimumlarda takılıp kalınabilmektedir. Yerel minimumlara takılmamak için başlangıç noktası rastgele seçilmiş ve yerel minimuma rastlandığında program tekrar çalıştırılarak sonuca ulaşılmaya çalışılmıştır.

En dik iniş yöntemi kullanılarak problemin doğrusal olmayan yapısı analiz edildikten sonra optimizasyon problemindeki serbest parametreleri azaltmak gerektiği ortaya çıkmıştır. Bu amaçla kafes parametre belirleme yöntemi kullanılmış ve problemdeki serbest parametre sayısının azaltılmasının yanında nümerik yöntemlerdeki işlem süresi de azaltılabilmektedir.

Son olarak, kafes parametre belirleme yöntemine dayanan ve *BWT Ters Çevrimi* adı verilen yöntem geliştirilmiştir. Bu yöntemle, nümerik yöntemler kullanılmadan, verilen 4×4 boyutlu bir KLT matrisine en küçük kareler bağlamında en yakın BWT matrisini üreten QMF süzgeçleri analitik olarak basit bir formülle hesaplanabilmektedir. Daha büyük boyutlu KLT matrisleri için ise denklemlerdeki terim sayısı üstel biçimde arttığından bu işlemler sonraki bir çalışmaya bırakılmış, burada ise sadece BWT ve KLT matrislerinin nasıl eşleştirilmesi gerektiğinden bahsedilmiştir.

BWT ters çevrimi yönteminin başarısını ölçmek için bazı test resimlerinin satırları ve sütunları üzerinden hesaplanan 4×4 boyutlu KLT matrislerini üreten QMF süzgeçleri bulunmaya çalışılmış ve bulunan süzgeçler Daubechies-2 süzgeç bankasıyla ve maksimum kodlama kazancını veren QMF süzgeciyle karşılaştırılmıştır. Karşılaştırma ölçütü olarak, süzgeçler kullanılarak resimlerin satırları veya sütunları altbantlara ayrıldığında altbantlarda elde edilen *kodlama kazancı* ve varyans dağılımları kullanılmıştır. Daubechies-2 süzgeçleri, mümkün olan en büyük kodlama kazancına çok yakın kodlama kazançları üretirken, BWT ters çevrimiyle bulunan süzgeçlerin de Daubechies-2'ye yakın kodlama kazançları

verdikleri görülmüştür. Bunun yanında altsinyallerin varyanslarının, Daubechies-2 süzgeçlerinde olduğu gibi, maksimum kodlama kazancı veren süzgeçlerdeki kadar başarılı bir şekilde ayrıştırabildiği gözlemlenmiştir.

8×8 boyutlu KLT matrisleri için analitik yöntem geliştirilmemiş olsa da, geliştirilmiş olsaydı elde edilecek sonuçlar nümerik yöntemlerde hesaplanmış ve bulunan süzgeç bankaları Daub-4 süzgeç bankasıyla karşılaştırılmıştır. BWT Ters Çevrimi yöntemiyle elde edilen kodlama kazançlarının, Daub-4 süzgeç bankasıyla elde edilenden daha yüksek olduğu bulunmuştur. 4×4 boyutundan 8×8 boyutuna geçildiğinde daha başarılı sonuçlar elde edilmesi, BWT Ters Çevrimi yönteminin gerçek başarısının daha büyük boyutlarda daha net bir şekilde ortaya çıkacağını göstermektedir. Çünkü daha büyük boyutlarda sinyalin barındırdığı bilgi, daha fazla sayıdaki serbest parametre ile daha etkili biçimde işlenebilecektir.

İleriki çalışmalar olarak, BWT ters çevriminin 8×8, 16×16 ve daha büyük boyutlu KLT matrislerini üretecek şekilde genelleştirilmesi ve QMF koşulu olmayan daha genel dalgacık aileleri için de tasarım çözümleri elde edilmesi planlanmaktadır.

KAYNAKLAR

- [1] Oppenheim, A.V. ve Willsky, A.S., *Signals and Systems*, 2nd ed., Prentice Hall, New Jersey, A.B.D., 1997.
- [2] Vetterli, M. ve Herley, C., "Wavelets and Filter Banks: Theory and Design," *IEEE Transactions on Signal Processing*, **40**, no. 9, pp. 2207–2232, Eylül 1992.
- [3] Sayood, K., *Introduction to Data Compression*, 3rd ed., Morgan Kauffmann, San Francisco, A.B.D., 2006.
- [4] Gonzalez, R.C. ve Woods, R.E., *Digital Image Processing*, 2nd ed., Prentice Hall, New Jersey, A.B.D., 2001.
- [5] Saghri, J.A., Tescher, A.J. ve Reagan, J.T., "Terrain Adaptive Transform Coding of Multispectral Data," *Proceedings International Conference on Geosciences and Remote Sensing (IGARSS '94)*, 1994, pp. 313–316.
- [6] Akansu, A.N. ve Haddad, R.A., *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, 2nd ed., Academic Press, A.B.D., 2001.
- [7] Salomon, D. ve Motta, G., *Handbook of Data Compression*, 1st ed., Springer, New York, A.B.D., 2010.
- [8] Huang, J., Zhu, R. ve Li, J., "Spatial Combinative Lifting Algorithm-Based Block Wavelet Transform for Image Compression," *2nd IEEE Conference on Industrial Electronics and Applications, ICIEA 2007*, Mayıs 2007, pp. 2516–2521.
- [9] Shapiro, J.M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, **41**, no. 12, pp. 3445–3462, Aralık 1993.
- [10] Said, A. ve Pearlman, W.A., "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, **6**, no. 3, pp. 243–250, Haziran 1996.

- [11] Hsiang, S.T. ve Woods, J.W., "Embedded Image Coding Using ZeroBlocks of Subband/Wavelet Coefficients and Context Modeling," *IEEE Int. Conf. on Circuits and Systems (ISCAS2000)*, Cenevre, İsviçre, 2000, pp. 662–665.
- [12] Taubman, D., "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, **9**, no. 7, pp. 1158–1170, Haziran 2000.
- [13] Çetin, A.E., Gerek, Ö.N. ve Ulukuş, Ş., "Block Wavelet Transforms for Image Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, **3**, no. 6, pp. 433–435, Aralık 1993.
- [14] Doğan, M. ve Gerek, Ö.N., "Blok Dalgacık Dönüşümü'nün Dikliği Üzerine," *SIU 2006, 14. IEEE Sinyal İşleme ve İletişim Uygulamaları Konferansı*, Antalya, Nisan 2006.
- [15] Ogden, C.L., Fryar, C.D., Carroll, M.D. ve Flegal, K.M., *Mean Body Weight, Height, and Body Mass Index, United States 1960–2002*. Centers for Disease Control and Prevention (CDC), No: 347, 27 Ekim 2004.
- [16] Hotelling, H., "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, **24**, pp. 417–441, 498–520, 1933.
- [17] Karhunen, K., "Über Lineare Methoden in der Wahrscheinlichkeitsrechnung," *Annales Academiae Fennicae, Series A*, **37**, pp. 1–79, 1947.
- [18] Loève, M., "Fonctions Aléatoires de Seconde Ordre," *Processus Stochastiques et Mouvement Brownien*, (Ed: Levy, P.), Hermann, 1948.
- [19] Kramer, H.P. ve Mathews, M.V., "A Linear Encoding for Transmitting a Set of Correlated Signals," *IRE Transactions on Information Theory*, **2**, pp. 41–46, Eylül 1956.
- [20] Huang, J.Y. ve Schultheiss, P.M., "Block Quantization of Correlated Gaussian Random Variables," *IEEE Transactions on Communication Systems*, **11**, pp. 289–296, Eylül 1963.

- [21] Hayes, M.H., *Statistical Digital Signal Processing and Modeling*, John Wiley, New York, A.B.D., 1996.
- [22] Haar, A., "Zur Theorie Der Orthogonalen Funktionensysteme," *Mathematische Annalen*, **69**, pp. 331–371, 1910.
- [23] Goupillaud, P., Grossmann, A. ve Morlet, J., "Cycle-Octave and Related Transforms in Seismic Signal Analysis," *Geoexploration*, **23**, pp. 85–105, 1984.
- [24] Grossmann, A. ve Morlet, J., "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," *SIAM Journal on Math. Anal.*, **15**, no. 4, pp. 723–736, 1984.
- [25] Mallat, S.G., "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. on Anal. Mach. Intel.*, **11**, no. 7, pp. 674–693, 1989.
- [26] Meyer, Y., "Orthonormal Wavelets," *Wavelets, Time-Frequency Methods and Phase Space*, (Ed: Combers, J.M., Grossmann, A. ve Tachamitchian, P.), Berlin, Almanya, Springer-Verlag, 1989.
- [27] Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure & Applied Mathematics*, **41**, no. 7, pp. 909–996, 1988.
- [28] Yan, R. ve Gao, R.X., "Tutorial 21, Wavelet Transform: A Mathematical Tool for Non-Stationary Signal Processing in Measurement Science," *IEEE Instrumentation & Measurement Magazine*, pp. 35–44, Ekim 2009.
- [29] Burrus, C.S. ve Parks, T.W., *DFT/FFT and Convolution Algorithms*, John Wiley & Sons, New York, A.B.D., 1985.
- [30] Burrus, C.S., Gopinath, R.A. ve Guo, H., *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice-Hall, Upper Saddle River, New Jersey, A. B. D., 1998.
- [31] Vaidyanathan, P.P., *Multirate Systems and Filter Banks*, Prentice Hall, Upper Saddle River, New Jersey, A. B. D., 1993.

- [32] Williams, J.R. ve Amaratunga, K., *A Discrete Wavelet Transform Without Edge Effects Using Wavelet Extrapolation*. IESL Technical Report, No: 95-02, 28 Ocak 1995.
- [33] Eom, E.K., Kim, Y.S. ve Kim, J.H., "A Block Wavelet Transform for Sub-Image Coding/Decoding," *SPIE Electronic Imaging*, **2669**, pp. 169–177, Ocak 1996.
- [34] Christopoulos, C., *JPEG2000 Verification Model 4.0 (Technical Description)*. No: ISO/IEC JTC1/SC29 WG1 N1282,.
- [35] Chrysafis, C. ve Ortega, A., "Line-Based, Reduced Memory, Wavelet Image Compression," *IEEE Transactions on Image Processing*, **9**, no. 3, pp. 378–389, Mart 2000.
- [36] Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, A. B. D., 1989.
- [37] Selesnick, I.W., *Maple and Parameterization of Orthogonal Wavelet Bases*. ECE Dept. and Computational Mathematics Laboratory, Rice University, Houston, Tx., A. B. D., Technical Report, 24 Ekim 1997.
- [38] Vaidyanathan, P.P. ve Hoang, P.Q., "Lattice Structures for Optimal Design and Robust Implementation of Two-Channel Perfect Reconstruction QMF Banks," *IEEE Trans. on Signal Processing*, **36**, no. 1, pp. 81–93, Ocak 1988.
- [39] Gerek, Ö.N., Gürcan, M.N. ve Çetin, A.E., "Frequency Band Characteristics of Tree-Structured Filter Banks," *Electronics Letters*, **32**, no. 8, pp. 724–726, 11 Nisan 1996.
- [40] Johnston, J.D., "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," *Proc. ICASSP*, pp. 291–294, 1980.
- [41] Haddad, R.A., "A Class of Orthogonal Nonrecursive Binomial Filters," *IEEE Trans. on Audio and Electroacoustics*, **AU-19**, no. 4, pp. 196–304, Aralık 1971.
- [42] Lorentz, G.G., *Bernstein Polynomials*, University of Toronto Press, A. B. D., 1953.