

**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**DALGACIK DÖNÜŞÜMLERİ VE HİYERARŞİK**  
**AĞAÇLARDA KÜME BÖLÜMLEME YÖNTEMİ İLE**  
**GÖRÜNTÜ SIKIŞTIRMA**

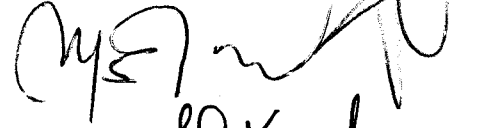

- 139807 -

Elektronik ve Haberleşme Müh. Gökhan BİLGİN

FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Haberleşme Programında  
Hazırlanan

**YÜKSEK LİSANS TEZİ**

Tez Danışmanı : Yrd. Doç. Dr. Ünal KÜÇÜK

Prof. Metin YÜCEL (MY)   
Prof. Dr. Ahmet H. Kayran 

**İSTANBUL, 2003**

# İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ.....	vii
ÖNSÖZ.....	viii
ÖZET.....	ix
ABSTRACT .....	x
1. GİRİŞ .....	11
1.1 Tarihi Gelişim.....	11
1.2 Tezin Gelişimi .....	13
2. GÖRÜNTÜ SIKIŞTIRMA.....	15
2.1 Yakınlık Ölçütleri .....	17
2.2 Sıkıştırma Sistem Modeli .....	18
2.3 Kayıpsız Sıkıştırma Yöntemleri .....	20
2.3.1 Temel Teknikler.....	21
2.4 Kayıplı Sıkıştırma Yöntemleri.....	22
2.4.1 Uzamsal (Spatial) Domen Kodlamaları .....	22
2.4.2 Dönüşüm Domeni Kodlamaları.....	23
3. SÜREKLİ DALGACIK DÖNÜŞÜMÜ .....	26
3.1 İki Boyutlu Sürekli Dalgacık Dönüşümü .....	27
3.2 Süzgeç Bankası Yorumu .....	27
3.3 Dalgacık Dizisi Yayılımı.....	29
3.3.1 Diyalik Dalgacıklar .....	29
3.3.2 Tanımlama.....	29
3.3.3 Sıkı (Compact) Diyalik Dalgacıklar .....	30
4. AYRIK DALGACIK DÖNÜŞÜMÜ VE GÖRÜNTÜ SIKIŞTIRMA .....	32
4.1 Ortalama ve Ayrıntılar .....	32
4.2 Ölçekleme Fonksiyonları ve Dalgacık Fonksiyonları.....	34
4.3 Çokluçözünürlük Analizi (Multiresolutional Analysis).....	37
4.3.1 İç Çarpım, Ortogonalite, Ortonormalite .....	38
4.3.2 Normalizasyon.....	40
4.4 Dalgacık Dönüşümü.....	40
4.5 Ters Dalgacık Dönüşümü.....	42

4.6	Daubechies Dalgacıkları .....	43
4.6.1	Ağırlıklı Ortalamalar ve Farklar .....	43
4.6.2	Alçak Geçiren ve Yüksek Geçiren Süzgeçleme .....	43
4.6.3	Matrisel Biçimde Gösterim.....	44
4.6.4	Dalgacık Dönüşümü.....	45
4.6.5	Ölçekleme ve Dalgacık Fonksiyonları.....	46
4.6.6	Daubechies Dalgacığ Katsayıları.....	47
4.7	Diğer Dalgacık Sistemleri .....	49
5.	AĞAÇ İÇERİKLİ GÖRÜNTÜ KODLAMASI .....	51
5.1	HAKB Algoritmasının Temel Geçişleri.....	53
5.2	Algoritma.....	56
6.	SONUÇLAR VE ÖNERİLER.....	59
6.1	Sonuçlar ve Uygulamalar .....	59
6.1.1	Haar Dalgacıklarıyla Geri Dönüşüm .....	61
6.1.2	Daubechies (D4) Dalgacıklarıyla Geri Dönüşüm.....	63
6.1.3	Biortogonal Dalgacıklarla (Bior 2.2) Geri Dönüşüm .....	65
6.1.4	İlerleyen İletim .....	67
6.2	Öneriler .....	68
KAYNAKLAR .....		70
INTERNET KAYNAKLARI .....		70
ÖZGEÇMİŞ.....		71

## SİMGE LİSTESİ

$\psi(x)$	Tek boyutlu temel dalgacık fonksiyonu
$\psi(x,y)$	İki boyutlu dalgacık fonksiyonu
$\phi(x)$	Tek boyutlu temel ölçek fonksiyonu
$\phi(x,y)$	İki boyutlu temel ölçek fonksiyonu
$\psi_{j,k}(t)$	Ölçeklenmiş ve ötelenmiş dalgacık fonksiyonu
$\phi_{j,k}(t)$	Ölçeklenmiş ve ötelenmiş ölçek fonksiyonu
$\tilde{\psi}(x)$	Tek boyutlu filtre bankası
$\tilde{\psi}(x,y)$	İki boyutlu filtre bankası
$\delta_{j,k}$	Kronecker delta
$L^2(\mathbb{R})$	Bir boyutlu Hilbert uzayı
$a_{a,b}$	Ortalama katsayıları
$c_{i,j}$	Dönüşüm katsayısı
$d_{a,b}$	Fark katsayıları
$A_n$	Dalgacık ortalama matrisi
$D_n$	Dalgacık fark matrisi
$H_n$	Yüksek geçiren süzgeç operatörü
$L_n$	Alçak geçiren süzgeç operatörü
$V^n$	Vektör uzayı
$W_f(a,b)$	Tek boyutlu $f(x)$ fonksiyonunun sürekli dalgacık dönüşümü
$W_f(a,b_x,b_y)$	İki boyutlu $f(x)$ fonksiyonunun sürekli dalgacık dönüşümü

## **KISALTMA LİSTESİ**

<b>ADD</b>	Ayrık Dalgacık Dönüşümü
<b>BKK</b>	Blok Kısaltma Kodlaması
<b>DCT</b>	Discrete Cosine Transform
<b>DKL</b>	Değerli Kümeler Listesi
<b>DPL</b>	Değerli Pikseller Listesi
<b>DzPL</b>	Değersiz Pikseller Listesi
<b>FKK</b>	Farksal Kestirim Kodlaması
<b>HAKB</b>	Hiyerarşik Ağaçlarda Küme Bölümleme
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>JPEG</b>	Joint Photographic Experts Group
<b>KUK</b>	Koşu Uzunluğu Kodlaması
<b>PSNR</b>	Peak Signal -to- Noise Ratio
<b>SDD</b>	Sürekli Dalgacık Dönüşümü
<b>SNR</b>	Signal -to- Noise Ratio
<b>SPIHT</b>	Set Partitioning In Hierarchical Trees



## ŞEKİL LİSTESİ

Şekil 2.1 Sıkıştırma Sistem Modeli (a) Sıkıştırma (b) Açma .....	18
Şekil 2.2 Sıkıştırıcı .....	19
Şekil 2.3 Açıcı .....	19
Şekil 3.1 Temel dalgacık .....	26
Şekil 3.2 İşaretin dalgacık dönüşümün süzgeç bankası yorumu .....	28
Şekil 3.3 Dalgacığın ikili ölçeklenmesi ve diyadik ötelenmesi [Castleman, 1996] .....	29
Şekil 4.1 Dalgacık Analizi ve Dönüşüm arasındaki ilişki .....	32
Şekil 4.2 Parçalı fonksiyon yaklaşımı .....	34
Şekil 4.3 Temel dalgacık ve ölçeklenmiş ve ötelenmiş türevleri .....	36
Şekil 4.5 128x128 Lenna görüntüsünün iki boyutlu Haar dalgacık dönüşümü .....	42
Şekil 4.6 Alçak ve yüksek geçiren süzgeçlerle ağaç yapısındaki dalgacık dönüşümü .....	45
Şekil 4.7 Ters dalgacık dönüşümünün ağaç yapısındaki gösterimi .....	46
Şekil 4.8 Daubechies D4 (a) ve D6 (b) dalgacık fonksiyonları [Misis, Oppenheim, Poggi, 2001] .....	48
Şekil 5.1 Piramitte alt bantlar arası ilişki .....	51
Şekil 5.2 Pikseller ve uzamsal yönelimli ağaçlar arasındaki ilişki .....	53
Şekil 5.3 Genel Ağaç İçerikli Küme Bölümleme Akış Diyagramı .....	56
Şekil 5.4 DzPL Test Algoritması .....	57
Şekil 5.5 DKL Test Algoritması .....	58
Şekil 6.1 Haar dalgacığıyla artan eşik değerinde (a-h) geri dönüşüm görüntüleri (bkz. Çizelge 6.1) .....	61
Şekil 6.2 Haar dalgacığı eşik değeri=20, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.2) .....	62
Şekil 6.3 Daubechies D4 dalgacığı ile artan eşik değerinde (a-h) geri dönüşüm görüntüleri. (bkz.Çizelge 6.3) .....	63
Şekil 6.4 Daubechies D4 dalgacığı eşik değeri=20, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.4) .....	64
Şekil 6.5 Biortogonal 2.2 dalgacığı ile artan eşik değerinde (a-h) geri dönüşüm görüntüleri. (bkz.Çizelge 6.5) .....	65
Şekil 6.6 Biortogonal 2.2 dalgacığı eşik değeri=40, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.6) .....	66
Şekil 6.7 İlerleyen İletim (a-l) HAKB kodlama minimum seviyeleri verilmiştir .....	67

## ÇİZELGE LİSTESİ

Çizelge 2.1 Çokluortam veri türleri ve sıkıştırılmamış depolama alanı, iletim bant genişliği ve gerekli iletim zamanı [Saha, 2000] .....	15
Çizelge 6.1 Haar dalgacığı geri dönüşüm değerleri.....	61
Çizelge 6.2 Haar dalgacığı tabanlı HAKB geri dönüşüm sonuçları .....	62
Çizelge 6.3 Daubechies D4 dalgacığı geri dönüşüm değerleri.....	63
Çizelge 6.4 Daubechies D4 dalgacığı tabanlı HAKB geri dönüşüm sonuçları .....	64
Çizelge 6.5 Biortogonal 2.2 dalgacığı geri dönüşüm değerleri. ....	65
Çizelge 6.6 Biortogonal 2.2 dalgacığı tabanlı HAKB geri dönüşüm sonuçları .....	66
Çizelge 6.7 İlerleyen iletim HAKB kodlama ve geri dönüşüm sonuçları.....	68



## ÖNSÖZ

Günümüzde giderek artan bir öneme sahip olan, dalgacık dönüşümleri ve hiyerarşik ağaçlarda küme bölümlene yöntemi ile görüntü sıkıştırma konulu bu tezi hazırlarken gerek işaret işleme gerekse görüntü işlem konularında kendimi geliştirme fırsatı buldum. Umarım bu çalışma daha sonra bu alanda çalışanlar için de yararlı bir kaynak olur.

Bu tezin gerçekleştirilmesinde her türlü desteğini esirgemeyen değerli danışmanım Yrd. Doç. Dr. Ünal Küçük'e teşekkür ederim.





## ÖZET

Sayısal görüntüler giderek artan bir oranda enformasyon dünyası içindeki payını artırmaktadır. İnternetin gelişimi; daha güçlü, satın alınabilir düzeylerdeki bilgisayarların yaygınlaşması, sayısal kamera, tarayıcı, yazıcı teknolojisindeki giderek artan ilerlemeler sayısal görsel imgelerin daha geniş kullanımına imkan sağlamıştır. Son zamanlarda görüntü verilerinin sıkıştırılması için gelişmiş algoritma çalışmalarına yeniden ilgi uyanmaya başlamıştır.. Sıkıştırma, gerek iletim hızında, gerek bilgilerin verimli bir şekilde depolanması için önemli bir kavramdır.

Bu çalışmada; görüntü sıkıştırmak için dalgacık dönüşümü yöntemi seçilmiştir. Herhangi bir dönüşüm kodlama tekniğinin başarısı temel fonksiyonların, işaretin özelliklerini ne kadar iyi temsil edebildiğine bağlıdır. Ayrık dalgacık dönüşümü (ADD) işaretin çokluçözünürlük analizini gerçekleştirir; bu özellikle ortalama ve ayrıntı işaret bölgeleri verimli bir şekilde sunulabilir. Bu sebeplerden dolayı JPEG2000 gibi günümüz görüntü sıkıştırma standartları dalgacık dönüşümünü kullanmaktadır.

Dalgacık dönüşümü ile görüntü ortalamalar ve farklar şeklinde alt bantlara ayrıştırılır. Elde edilen ayrıntı alt bantları görüntü enerjisinin yüzde olarak çok az bir bölümünü oluşturan küçük değerli katsayılara sahiptir. Bu küçük değerli katsayıların seçilecek eşik değeri ile sıfır değerine nicemlenmesi görüntüde büyük bir kayba neden olmamaktadır. Bu işleme oluşturulan görüntü büyük oranda sıfır değerine sahip piksellerden oluşmaktadır. Dalgacık dönüşümünün sağladığı bu özellik hiyerarşik ağaçlarda küme bölümlene algoritmasına temel teşkil etmektedir. Bu algoritmada ayrık dalgacık dönüşümü katsayıları uzamsal yönelimli ağaçlar şeklinde organize edilir. Bu ağaçların oluşturulmasının sebebi birbiriyle yüksek derecede ilişkili olan piksellerin bir küme altında toplanmasıdır. Bu şekildeki küme yapılandırma ile bir seviyedeki katsayı ile diğer seviyedeki benzerlik oranı artırılmış olur. Seçilen algoritmada da temel amaç birbiriyle yüksek derecede ilişkili olan, sıfır değerli ağaçları tek bir değerle kodlamaktır. Dalgacık dönüşümünün doğasındaki bu özellik sayesinde hiyerarşik ağaçlarda küme bölümlene algoritması ile birlikte görüntülerde yüksek sıkıştırma oranı elde edilmesi amaçlanmıştır.

**Anahtar kelimeler:** Dalgacıklar, dalgacık dönüşümü, görüntü sıkıştırma, hiyerarşik ağaçlarda küme bölümlene (HAKB), ortogonal dalgacıklar

## ABSTRACT

Digital images are claiming an increasingly larger portion of the information world. The growth of internet, along with more powerful and affordable computers and continuing advances in the technology of digital cameras, scanners and printers has led to the widespread use of digital imagery. Nowadays, there is a renewed interest in improving algorithms for the compression of image data. Compression is important both for the speed of transmission and the efficiency of storage.

In this study, wavelet transform is chosen for image compression. The success of any transform coding technique depends on how well the basis functions represent the signal features. The discrete wavelet transform (DWT) performs a multiresolutional analysis of a signal; this enables an efficient representation of averaged and detailed signal regions. For these reasons, recent image compression standards such as JPEG2000 use the wavelet transform.

Image is decomposed into subbands of averages and details with wavelet transform. Obtained detail subbands have small valued coefficients that also constitute a small percentage of image energy. If these small valued coefficients are quantized to zero with a chosen threshold, there will be no great loss in the image. The image which is formed with this computation has a big portion of zero valued pixels. This feature provided by wavelet transform is a basis for set partitioning in hierarchical trees algorithm. With this algorithm discrete wavelet transform coefficients are organized in spatial orientation trees. The reason to have formed these trees is gathering all pixels that are highly correlated with each other. With this kind of set structure, the similarity among the coefficients in a set from one level to the next is increased. The main aim in this algorithm is to code the trees with highly correlated, zero valued coefficients with a single code word. With these features in the nature of wavelet transform and set partitioning in hierarchical trees algorithm, it is intended to get high compression ratios in images.

**Keywords:** Wavelets, wavelet transform, image compression, set partitioning in hierarchical trees (SPIHT), orthogonal wavelets.

## 1. GİRİŞ

Dalgacıklar belirli matematiksel gereklilikleri yerine getiren, veri ve diğer fonksiyonların gösteriminde kullanılan fonksiyonlardır. 1800’lerde Joseph Fourier fonksiyonları, sinüs ve kosinüs fonksiyonların toplamı şeklinde ifade ettiğinden beri “yaklaşıklık toplamsallığı-approximation superposition-” kavramı kullanılagelmiştir. Bununla birlikte dalgacık analizinde veriye, ölçek (scale) yönünden bakışımız özel bir rol oynamaktadır. Dalgacık analizinde veri değişik ölçek veya çözünürlüklerde incelenir. Eğer işarete veya fonksiyona geniş pencereden bakılırsa büyük özellikler gözlemlenirken; küçük pencereden bakılırsa bu defa da ayrıntılar gözlemlenir. Bir başka anlatımla dalgacık analizinde orman ve ormanı oluşturan ağaçlar beraber gözlemlenir.

Dalgacık fonksiyonları bu kullanışlılığı dolayısıyla araştırmacıların ilgisini çekmiştir. Yıllarca bilim insanları Fourier analizinde kullanılan sinüs ve kosinüs fonksiyonlarından daha uygun fonksiyonları kullanmak istemişlerdir. Onların tanımıyla bu fonksiyonlar, yerel olmayan (nonlocal), sonsuza uzanan fonksiyonlardır. Dalgacık fonksiyonları ise sonlu uzunluktaki alanda yaklaşıklık fonksiyonları olarak kullanılır. Özellikle görüntü işleme yönünden olaya bakarsak, dalgacık fonksiyonunun kullanımı keskin geçişe sahip veri veya fonksiyonlara yaklaşıklık açısından oldukça önemli bir adımdır.

Dalgacık fonksiyonları bilim alanında kendine oldukça geniş uygulama sahası bulmuştur: astronomi, akustik, nükleer mühendislik, alt bant kodlaması, işaret ve görüntü işleme, nörofizyoloji, müzik, manyetik rezonans görüntüleme, optik, fraktallar, türbülans ve deprem tahmini, insan görüşü üzerine çalışmalar ve matematikte parçalı farksal denklem çözümlene.

### 1.1 Tarihi Gelişim

Dalgacık analizi matematik tarihi içerisinde çeşitli köklere sahiptir. Fakat bu kökler yoğun olarak 1930’lardan sonraki dönemlere ait ayrı çalışmalara uzanmaktadır.

#### 1930 Öncesi

1930 Öncesi dalgacıklara öncülük eden gelişmeler, Joseph Fourier’in “Fourier sentezi” olarak bilinen matematiksel çalışmalarına dayanmaktadır. Bu çalışmada  $2\pi$ ’lik periyodik bir fonksiyon,  $f(x)$ , şu şekilde ifade edilmiştir:

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx))$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx, \quad a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx \quad (1.1)$$

Fourier'in bu katkısı fonksiyonlar hakkında matematikçilere temel bir fikir kapısı açmıştır. Ardından Fourier serisi açılımı ve ortogonal sistemlerle birlikte matematikçiler frekans analizi tanımından ölçek analizi tanımına geçmeye başlamıştır.

Dalgacıklar tanım olarak literatürde, bu alanda öncü kabul edilebilecek olan Haar'ın tezinin ek kısmında yer almıştır (1909). Haar'ın kullandığı dalgacığının bir özelliği de sıkı desteğe (compact support) sahip olmasıdır. Fakat Haar dalgacığının sürekli türevinin olmaması uygulamalarda sınırlamalar getirmektedir.

### 1930'lar

1930'lardan itibaren birçok grup bağımsız olarak ölçekle değişen fonksiyonların gösterimi üzerine çalışmıştır. Temel fonksiyonların ve ölçekle değişen temel fonksiyonları anlamak dalgacıkları anlamak için anahtar kavramlardır.

Paul Levy (1930, fizikçi) ölçekle değişen Haar temel fonksiyonlarını inceleyerek rasgele bir işaretin "Brown Hareketini" araştırmıştır. Çalışmasında, Brown hareketinin az karışıklıktaki ayrıntılarının bulunmasında Haar temel fonksiyonlarının, Fourier temel fonksiyonlarından elde edilen sonuçlardan daha yararlı olduğunu tespit etmiştir.

Diğer bir çalışma Littlewood, Paley, Stern tarafından bir fonksiyonun enerjisinin hesaplanması üzerine yapılmıştır.

$$E = \frac{1}{2} \int_0^{2\pi} |f(x)|^2 dx \quad (1.2)$$

Hesaplama geniş bir aralıkta dağılmış veya birkaç noktada yoğunlaşmış enerji dağılımında değişik sonuçlar elde edilmiştir. Bu sonuçlar enerjinin muhafaza edilemeyeceğini gösterdiğinden birçok bilim adamını rahatsız etmiştir. Araştırmalar ölçekle değişen fonksiyonların kullanımıyla, sadece hesaplamada, fonksiyonun enerjisini koruyabileceğini ortaya koymuştur. Bu çalışma David Marr'ın 1980'lerde dalgacıklarla nümerik görüntü işleme algoritması geliştirmesinde dayanak olmuştur.

### 1960-1980

Bu yıllar arasında Guido Weiss ve Ronald R. Coifman "atom" olarak adlandırılan, bir fonksiyon uzayının en basit elemanı üzerine çalıştılar. Ortak bir fonksiyonun atomunu bulmak

ve birleşim kurallarını tespitteki amaç, bu atomlarla fonksiyon uzayının elemanlarının yeniden oluşturulmasıdır. 1980’de Grossman (fizikçi) ve Morlet (mühendis) kuantum fiziğinin kavramları arasına dalgacıkları da katmışlardır. Ayrıca bu iki araştırmacı dalgacıkların fiziksel sezgi üzerindeki bir düşünce yolu üzerinde çalışmışlardır.

### **1980 Sonrası**

1985’te Stephen Mallat sayısal işaret işleme çalışmaları doğrultusunda dalgacıklara yeni bir yorum getirmiştir. Dörtgenel ayna süzgeçler (Quadrature Mirror Filters), piramit algoritmaları ve ortonormal dalgacık temelleri konusunda bazı ilişkileri keşfetmiştir. Y.Meyer bu çalışmalardan esinlenerek ilk, sıradan olmayan (nontrivial) dalgacıkları oluşturdu. Haar dalgacıklarından farklı olarak Meyer dalgacıklarının sürekli türevleri alınabilir fakat bu dalgacıklar sıkı destek özelliğine sahip değildirler. Birkaç yıl sonra Ingrid Daubechies yine bu alanda çalışan Mallat’ın çalışmalarını kullanarak daha sık ve bugünkü dalgacık uygulamalarının temeltaşı olan, temel ortonormal dalgacıklar üzerine olan çalışmalarını bilim dünyasına sundu.(Graps, 1995)

## **1.2 Tezin Gelişimi**

İkinci bölümde görüntü sıkıştırmanın gerekliliği ve öneminden bahsedildikten sonra görüntü sıkıştırma teknikleri anlatılacaktır. Önce nesnel olarak yakınlık ölçütleri belirlenecek ardından sıkıştırma sistem modeli incelenecektir. Kayıplı ve kayıpsız sıkıştırma tekniklerinin özellikleri ve teknikleri genel olarak aktarılacaktır.

Üçüncü bölümde dalgacıklar tanıtılacak, matematiksel özellikleri incelenecektir. Bu bölümde ilk önce tek boyutlu dalgacıklar daha sonra iki boyutlu dalgacıklar üzerinde durulacaktır. Ardından sinyal işleme açısından dalgacıkların süzgeç bankası olarak nasıl kullanıldığı açıklanacaktır. Bölüm kapsamında dalgacık dizisi yayılımı aktarıldıktan sonra diyardik dalgacık kavramı ve özellikleri incelenecektir.

Dördüncü bölümde tezde kullanılan ayrık verilerin ayrık dalgacıklar yardımıyla nasıl işleneceği ayrıntılı olarak anlatılacaktır. Bu bölümde ayrık dalgacık dönüşümün görüntü sıkıştırma işleminde nasıl kullanılacağı; Haar dalgacığı temel alınarak, daha sonra konu diğer dalgacıklar da göz önünde bulundurularak, genişletilerek anlatılacaktır. Ayrıca dalgacık dönüşümlerinin geri dönüşüm işleminden ve özelliklerinden bahsedilip, sıkıştırılan görüntülerin tekrar yapılandırılması matematiksel olarak açıklanacaktır. Görüntü sıkıştırmada kullanılan ve dalgacık analizinde önemli bir yeri olan Daubechies dalgacıklarının üzerinde durulduktan sonra diğer dalgacık sistemlerine de genel bir yaklaşımda bulunulacaktır.

Beşinci bölümde dalgacık dönüşümüyle elde edilen, alt bantlara ayrılmış görüntü; alt bantlar arasındaki ilişkiyi kullanan, Said ve Pearlman tarafından geliştirilen hiyerarşik ağaçlarda küme bölümlene (HAKB) sıkıştırma algoritması incelenecektir. Bu algoritma dalgacık dönüşümleri göz önüne alınarak geliştirilmiş sıfır ağacı algoritmasının geliştirilmiş bir türevidir. Bu bölümün sonunda HAKB algoritmasının akış diyagramı sunulacaktır.

Altıncı bölümde dalgacık dönüşümleri ve HAKB algoritması ile elde edilen sıkıştırılmış görüntü örnekleri, dalgacık çeşitleri ve sıkıştırma oranları açısından karşılaştırmalı olarak sunulacaktır. Ardından sonuç değerlendirmesi ve öneriler açıklanarak tez sonuçlandırılmış olacaktır.



## 2. GÖRÜNTÜ SIKIŞTIRMA

Görüntü sıkıştırma, internet alanındaki büyük gelişim, bilgisayarların günlük kullanımındaki payının artması, çokluortam pazarındaki hızlı yükselişe beraber görüntü işleme konuları arasında öne çıkmıştır.

Çizelge 2.1 basit yazıdan video görüntüsüne kadar, disk alanı, iletim bant genişliği ve iletim zamanı gibi iletim ve depolama ihtiyaçlarına göre sıkıştırılmamış veri için nitel bir karşılaştırmayı göstermektedir. Bu çizelgeden veri işleme alanında sıkıştırmanın gerekliliği ve önemi daha iyi anlaşılabilir.

Çizelge 2.1 Çokluortam veri türleri ve sıkıştırılmamış depolama alanı, iletim bant genişliği ve gerekli iletim zamanı (Saha, 2000)

Çokluortam Verisi	Boyut/Süre	Bit/Piksel veya Bit/Örnek	Sıkıştırılmamış Boyut (B:Bayt)	İletim Bant genişliği (b:bit)	İletim Zamanı (28K modem)
Yazı Sayfası	11" x 8.5"	Değişen çözünürlükte	4-8 KB	32-64 Kb/sayfa	1.1 - 2.2 sn
Telefon kalitesinde konuşma	10 sn	8 Kbps	78.125 KB	64 Kb/sn	22.2 sn
Siyah Beyaz Resim	512 x 512	8 bpp	256 KB	2.1 Mb/resim	1 dk 13 sn
Renkli Resim	512 x 512	24 bpp	768 KB	6.29 Mb/resim	3 dk 39 sn
Tıbbi Görüntü	2048 x 1680	12 bpp	4.92 MB	41.3 Mb/resim	23 dk 54 sn
Video	640 x 480, 1dk (30 çerçeve/sn)	24 bpp	1.545 GB	221 Mb/sn	5 gün 8 saat

Görüntü sıkıştırma konu olarak görüntü verilerindeki gerekli enformasyonun saklanıp, sunulmasıyla ilgilendir. İşlenerek boyutu azaltılmış dosyalara "sıkıştırılmış dosya" denir. Sıkıştırılmamış orijinal dosya ile sıkıştırılmış dosya boyutu arasındaki orana "sıkıştırma oranı" olarak tanımlanır.

Sıkıştırma Oranı = Orijinal dosya boyutu / Sıkıştırılmış dosya boyutu

Bu oran genel olarak Boyut<sub>0</sub>:Boyut<sub>s</sub> olarak ifade edilir; örneğin 10:1, 20:1 sıkıştırma oranları. Sıkıştırma miktarının belirlenmesinin diğer bir yolu da piksel başına düşen bit miktarını

belirlenmesidir. Görüntü işleme terminolojisinde bu miktar “piksel başına düşen bit -bit per pixel-” olarak adlandırılır. NxN boyutlarındaki görüntü için ;

$$\text{Piksel başına düşen bit} = \text{Toplam bit sayısı} / \text{Piksel sayısı} = (8 \times \text{bayt sayısı}) / \text{NxN}$$

Dosya boyutlarındaki azaltma işlemi, birçok iletişim sistemi için ve bilgisayar veritabanlarındaki depolama için gereklidir. Özellikle sayısal görüntülerdeki veri miktarı oldukça büyüktür. Bu, internet uygulamalarında büyük bir dezavantaj oluşturur.

Başarılı bir sıkıştırma planı için yapılması gerekli ikinci tanımlama, “tutulan gerekli enformasyon (retaining necessary information)” olacaktır. Bu kavramı anlamak için “veri” ve “enformasyon” arasındaki farkın anlaşılması gerekir. Sayısal görüntülerde veri piksel gri-seviye değerlerinin diğer bir deyişle parlaklığının bir göstergesidir. Enformasyon ise verinin anlamlı bir tarzda yorumudur. Verinin enformasyonu taşınmasını, tıpkı alfabenin kelimeler yardımıyla bilgiyi taşınmasına benzetilebilir.

Görüntü sıkıştırma yöntemleri iki öncül tipte incelenir. İlk tip “kayıplı yöntemler -lossless methods-” olarak adlandırılır. Bu yöntemde sıkıştırılmış dosyadan tekrar elde edilen dosya, orijinal dosya ile birebir aynıdır; herhangi bir kayıp yoktur. Kayıpsız yöntemde görüntünün karmaşıklığına bağlı olarak sıkıştırma miktarı değişiklik gösterir. Örneğin sadece yazıdan oluşan görüntülerde yüksek sıkıştırma oranlarına ulaşılır.

Sıkıştırma yöntemlerinin ikinci tipi ise “kayıplı yöntemler -lossy methods-” olarak adlandırılır. Bu yöntemde ise sıkıştırmada, belirlenen koşullara göre yeniden elde edilen görüntü orijinal haliyle birebir aynı değildir. Bu yöntemle karmaşık görüntüler için daha yüksek sıkıştırma oranlarına ulaşılabilir.

Sıkıştırma algoritmaları görüntü verisindeki gereksizliklerin (redundancies) çeşitli yöntemlerle azaltılması veya yok edilmesi esasına dayanır. Görüntülerde üç temel gereksizlik tipi bulunur:

- 1.) Kodlama (Coding)
- 2.) Pikseller-arası (Interpixels)
- 3.) Psikogörsel (Psychovisual)

Kodlama gereksizliğinde, görüntüyü temsil eden veri optimal bir tarzda ifade edilmemiştir. Örneğin 16 gri-seviyeye sahip görüntülerde piksel başına 8 bit/piksel oranında bir kodlama yapılması.



Pikseller arası gereksizlik, ardışıl pikseller arasındaki yüksek dereceli ilişkinin hesaba katılmamasından oluşur. Görüntüdeki parlaklık seviyeleri çok hızlı bir değişim göstermeyip, dereceli geçişlere sahiptir. Piksel değerleri doğal olarak birbirine yakın değerlerden oluşur.

Psikogörsel gereksizlik, insan görme sistemindeki (Human Visual System) özellikler dolayısıyla oluşur. Görme olayında uzay frekanslarının 50 çevrim/derece değerinin altındaki değişimler algılanır. Daha yüksek frekanslı değişimler göz tarafından algılanmaz.

## 2.1 Yakınlık Ölçütleri

Görüntü sıkıştırmadaki algoritmaların geliştirilmesi, gerekli enformasyon için ihtiyaç duyulan minimal verinin belirlenmesi esasına dayanır. Hangi enformasyonun önemli olduğu ise “görüntüye yakınlık ölçütleri” olarak belirlenir. Yakınlık ölçütleri iki sınıfa ayrılır; nesnel yakınlık ölçütleri ve öznel yakınlık değerleri.

Nesnel yakınlık değerleri sayısal işaret işleme ve enformasyon teorisinin sağladığı ölçüm eşitlikleri yardımıyla belirlenir. Genel olarak kullanılan nesnel ölçüm yöntemleri: ortalama karesel hata ( $e_{ms}$ ), işaret gürültü oranı (SNR) ve tepe işaret gürültü oranı (PSNR) olarak belirtilebilir. Hata ise orijinal görüntü ve yeniden yapılandırılan görüntü arasındaki değer farklılığı olarak tanımlanabilir. (Umbaugh, 1999)

$$hata(r, c) = I(r, c) - \hat{I}(r, c)$$

$$I(r, c) = \text{Orijinal görüntü} \quad \hat{I}(r, c) = \text{Yeniden yapılandırılan görüntü}$$

$N \times N$  sıkıştırılmış görüntüdeki toplam hata:

$$ToplamHata = \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} [I(r, c) - \hat{I}(r, c)] \quad (2.1)$$

ortalama karesel hata:

$$e_{ms} = \frac{1}{N^2} \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} [I(r, c) - \hat{I}(r, c)]^2 \quad (2.2)$$

Hata ölçümünde toplam hatanın veya ortalama karesel hatanın küçük değer alması iyi şekilde sıkıştırılmış görüntüyü temsil eder; buna karşın işaret gürültü oranı ölçütünde büyük değerler iyi sıkıştırmayı temsil eder.

$$SNR = 10 \log_{10} \left( \frac{\sum_{r=0}^{N-1} \sum_{c=0}^{N-1} [I(r,c)]^2}{\sum_{r=0}^{N-1} \sum_{c=0}^{N-1} [I(r,c) - \hat{I}(r,c)]^2} \right) \quad (2.3)$$

tepe işaret gürültü oranı ise:

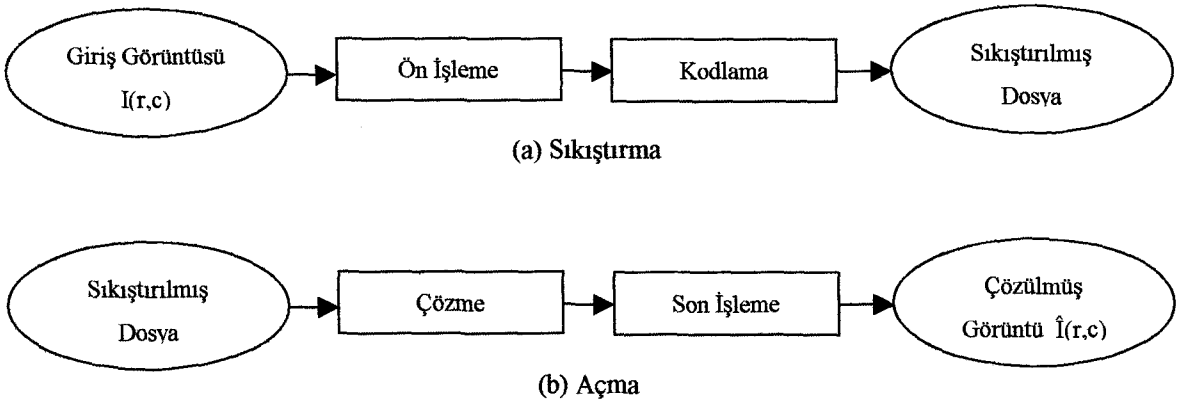
$$PSNR = 10 \log_{10} \frac{(L-1)^2}{\frac{1}{N^2} \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} [I(r,c) - \hat{I}(r,c)]^2} \quad L = \text{Toplam gri-seviye sayısı.} \quad (2.4)$$

Öznel yakınlık ölçütleri, görüntü kalitesini değerlendirmek için bir kalite ölçeğine ihtiyaç duyar. Bu ölçek insan test unsurları tarafından belirlenir. Yargısız sonuçlar elde edebilmek ve öznel ölçümlerin kalitesinin artırılması amacıyla öznel dikkatli seçimi ve iyi tasarlanmış değerlendirme deneyimleri gerekir. Sonuçlar istatistiksel olarak test edilerek ortalamalar ve standart sapma ölçütleri kullanılır.

## 2.2 Sıkıştırma Sistem Modeli

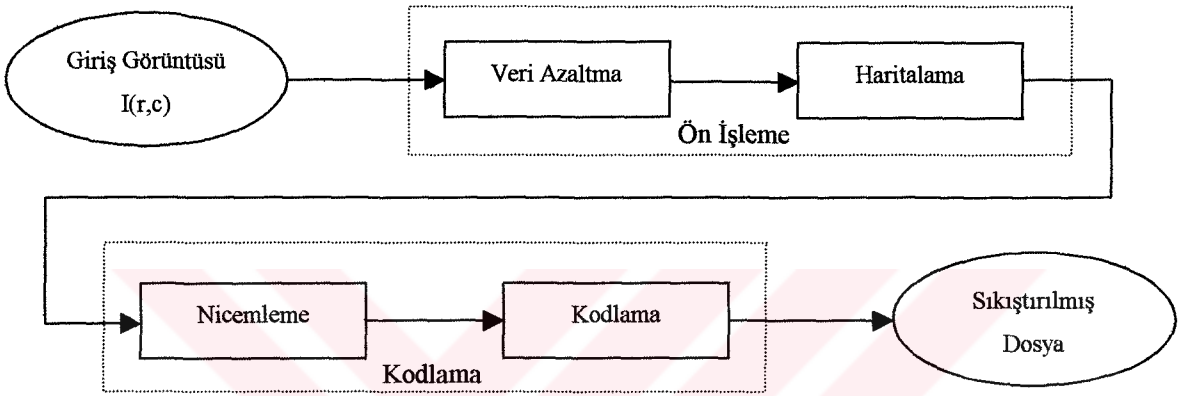
Sıkıştırma sistem modeli sıkıştırıcı ve açıcı olmak üzere iki ana bölümden oluşur. Sıkıştırıcı, ön işleme ve kodlama bölümlerinden; açıcı ise çözme ve son işleme bölümlerinden oluşur (Şekil 2.1).

Kodlamadan önceki ön işleme, görüntüyü uygulamaya bağlı olarak kodlamaya hazır hale getirir. Sıkıştırılan dosya açıldıktan sonraki son işleme ise potansiyel istenmeyen etkilerin giderilmesini ve görüntünün sunulacak biçime getirilmesiyle sorumludur. Çoğunlukla pratikteki sıkıştırma algoritmaları, çeşitli sıkıştırma tekniklerinin bir kombinasyonundan oluşur.



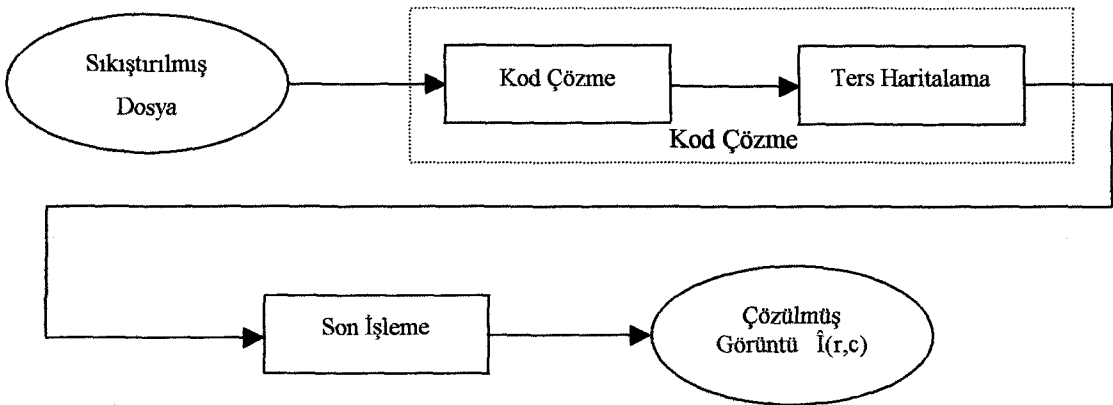
Şekil 2.1 Sıkıştırma Sistem Modeli (a) Sıkıştırma (b) Açma

Sıkıştırıcı şekil 2.2’de görüldüğü gibi alt bölümlere ayrılır. Ön işlemedeki ilk işlem veri miktarının azaltılmasıdır. Bu blokta veri gri-seviye ve/veya uzamsal nicemleme ile azaltılabilir; ayrıca istenmeyen etkiler görüntü zenginleştirme ve yeniden yapılandırma teknikleriyle giderilebilir (örneğin gürültü giderme). Ön işlemedeki ikinci adım haritalama işlemidir. Bu adımda orijinal görüntü verisi daha kolay bir biçimde sıkıştırma işleminin gerçekleştirilebileceği matematik uzayında ifade edilir. Kodlama aşamasının ilk adımı ise nicemleme işlemidir. Haritalama işlemindeki potansiyel sürekli veri ayrık biçimde temsil edilir. Kodlama aşamasındaki son bölüm ise sonuç verinin kodlanması işlemidir.



Şekil 2.2 Sıkıştırıcı

Açıcı da alt bölümlere ayrılır (Şekil 2.3). İlk kısım olan kod çözme aşaması iki alt bölüme; kodun çözülmesi ve ters haritalandırma bölümlerine ayrıştırılabilir. Kod çözme aşamasındaki ilk bölümde sıkıştırılmış dosyadan, orijinal kodlamadaki haritalanan nicemlenmiş değerler yeniden elde edilir. Bir sonraki adımda ters haritalamayla orijinal görüntü veya yaklaşıklığı elde edilir. Uygulamaya bağlı olarak son işleme aşamasıyla açma işlemi tamamlanır.



Şekil 2.3 Açıcı

Sıkıştırma algoritmalarının geliştirilmesi yüksek derecede uygulamaya bağlıdır. Algoritmaların kodlama bölümünün tasarlanması oldukça önemlidir; kodlayıcı birebir haritalamayı gerçekleştirir; her bir giriş için tek bir çıkış belirlenir. Kod geridönüşümü sağlanacak yapıda yapılandırılır. Kod uzunluğu eşit olabileceği gibi değişken uzunluklarda da olabilir. Bir çok durumda değişken uzunluklu kodlar veri sıkıştırması için en verimli planı oluştururlar. Buna karşın işleme uzunluğu, gerek kodlama gerekse çözme işlemleri için uzar.

### 2.3 Kayıpsız Sıkıştırma Yöntemleri

Kayıpsız sıkıştırma yöntemleri özellikle bazı uygulamalarda zorunludur. Örneğin tıbbi görüntülerin saklanması, kanunen, veri kaybı olmaksızın arşivlenmesi zorunludur. Aslında kayıpsız teknikleri çoğu görüntü dışı veriler için geliştirilmiş olup görüntü sıkıştırması için optimal bir çözüm sunmaz. Bununla birlikte kayıpsız teknikler görüntü sıkıştırma algoritmalarında ön işleme ve son işleme fazlarında kullanılır.

Kayıpsız sıkıştırmanın dayandığı ana teoriler, olasılık teorisinin matematiksel temeline dayanan haberleşme ve enformasyon teorisinden gelir. Burada kullanılan temel düşünce enformasyonun içeriği ve verideki rastgeleliktir. Enformasyon teorisine göre bir olayın oluşma ihtimali az ise oluşma ihtimali daha yüksek olaylardan daha fazla enformasyon taşır. Görüntü verisindeki ortalama enformasyonun, "entropi", ölçütü bu noktada önemli bir temel oluşturur.  $N \times N'$  lik bir görüntü için entropi:

$$Entropi = - \sum_{i=0}^{L-1} p_i \log_2(p_i) \text{ bpp} \quad (2.5)$$

$p_i = i$ . Gri-seviye olasılığı ( $n_k / N^2$ )

$n_k = k$ . gri-seviye toplam piksel sayısı

$L =$  Toplam gri-seviye sayısı ( 8 bit için 256 )

Entropi değeri, görüntünün kodlanmasında teorik olarak piksel başına düşen bit sayısının ortalamasının minimum değerini verir. Özellikle kodlama işlemi sırasında kodlama planının başarımını değerlendirebileceğimiz bir kriter oluşturur.

$$0 \leq Entropi \leq \log_2 L$$

Bu sonuç enformasyon teorisindeki, her bir değişkenin rastgeleliği olasılığının hesabından elde edilir ve bir pikselin diğer bir pikselle ilişkili olup olmadığı esasına dayanır. Görüntüdeki rastgelelik, gri-seviyelerindeki dağılımı arttırır; sonuçta verinin temsilinde her bir piksel

başına düşen bit miktarının artmasına yol açar. Tersine bir durumda birbirine olasılık olarak yakın değerlerden oluşmuş bir görüntüyü temsil etmek için piksel başına daha az bit kullanılır. Entropi, kodlayıcının performansını da değerlendirebileceğimiz bir ölçüttür. Ortalama piksel başına düşen bit sayısı ise:

$$L_{ort} = -\sum_{i=0}^{L-1} l_i \cdot p_i \quad (2.6)$$

$l_i$  = i. gri-seviyedeki bitlerin uzunluğu

$p_i$  = i. gri-seviyedeki histogram olasılığı

$L_{ort}$ , entropiye ne kadar yakın olursa bu kodlamanın iyi bir sonuca yaklaştığının göstergesidir; eşit olması durumunda ise “optimum kod” elde edilir.

### 2.3.1 Temel Teknikler

#### Huffman Kodlaması:

Huffman kodlamasında değişken kod uzunlukları elde edilir. Görüntü histogramdaki yüksek olasılıklı değerler en az sayıda bitle temsil edilir. Temel olarak algoritma adımları:

- 1.) Görüntü histogramından gri-seviye olasılıklarının bulunması.
- 2.) En küçükten büyüğe giriş olasılıklarının sıralanması.
- 3.) En az olasılık değerine sahip iki olasılığın toplanması.
- 4.) İki olasılık değeri klana kadar 2. adıma dallanılması.
- 5.) Geriye doğru giderek, 0 ve 1 değerleri verilerek seviyelere değişken uzunluklu kodun atanması.

#### Koşu Uzunluğu Kodlaması (Run Length Coding):

KUK ardışıl, benzer gri-seviye değerindeki piksellerin sayılması esasına dayanır. Genel olarak ikili görüntülerde veya eşik değeri yardımıyla gri-seviye değerleri azaltılmış görüntülerde kullanılır. Dikey ve yatay KUK yöntemleri vardır; bu yöntemlerde  $2^n$  piksellik satır veya sütundan oluşan görüntülerde n bitten oluşan bir kodlama kullanılır.

KUK'nın gelişmiş bir çok türevi de bulunur. “Bit düzeyi KUK” için her bir ikili rakam için yatay ve/veya dikey KUK uygulanır. Bu yöntemin dışında “Gray Koduna” dönüştürülerek azaltılmış görüntü verilerin haritalanmasına dayanan yöntemler de bulunur.

### **Lempel-Ziv-Welch Kodlaması:**

Bu algoritma “veri dizilerinin” kodlanması esasına dayanır. Görüntü alanında ise veri dizilerini piksel değerlerinin ardışıl katarları oluşturur. Bir dizi tablosu ve ona karşı düşen ilgili kodların oluşturulması ile gerçekleştirilir. Orijinal görüntü veri uzunluğundan daha uzun bir kod uzunluğu oluşturulur. Bu yöntemdeki kod yapılandırma işlemi ise sıkıştırma işlemi sırasında gerçek zamanlı olarak oluşturulur.

### **Aritmetik Kodlama:**

Aritmetik kodlamada giriş verisi küçük kayan noktalı ( floating point ) sayıya dönüştürülür. Elde edilen değer belirlenen gri-seviye değerli dizileri ifade etmek için kullanılır. Aritmetik kodlamada görüntü histogramından yararlanarak, entropiye teorik olarak en başarılı biçimde ulaşılmaya çalışılır.

## **2.4 Kayıplı Sıkıştırma Yöntemleri**

Karmaşık görüntülerde yüksek sıkıştırma oranları elde edilmek isteniyorsa kayıplı sıkıştırma yöntemleri uygulanır. Kayıplı sıkıştırmada görüntü kalitesi ve uygulamaya bağlı olarak sıkıştırmanın derecesi arasında bir denge aranır. Bazı ileri yöntemlerle görüntüler görsel açıdan enformasyon kayıpları olmaksızın 10-20 kat sıkıştırılmaktadır. Görüntü zenginleştirme ve yeniden yapılandırma teknikleri çeşitli sıkıştırma teknikleriyle birleştirilerek kayıplı sıkıştırma planları elde edilir. Birçok kayıplı yöntem ayarlanabilir parametrelerle kullanıcının, sıkıştırma oranı ve görüntü kalitesi arasında seçim yapabilmesine olanak sağlar.

Kayıplı sıkıştırma teknikleri, uzamsal (spatial) ve dönüşüm (transform) domeni kodlamaları olarak iki sınıfa ayrılır:

### **2.4.1 Uzamsal (Spatial) Dömen Kodlamaları**

#### **Gri-seviye Koşu Uzunluğu Kodlaması:**

Gri-seviyelerin eşik, nicemleme gibi çeşitli yöntemlerle azaltılması ve standart veya gelişmiş (örneğin dinamik pencere tabanlı KUK) KUK tekniklerinin kullanılması ile kayıplı bir sıkıştırma elde edilir.

#### **Blok Kısaltma Kodlaması (Block Truncating Coding):**

BKK kodlamasında görüntü alt görüntü bloklarına bölünür. Her bir alt bloktaki gri-seviyeler azaltılır. Azaltma işlemi bölgesel görüntü istatistiklerine uyarlanmış bir “nicemleyici” tarafından gerçekleştirilir. Temel blok kısaltma kodlaması 4x4'lük 16 pikselden oluşan yapılara ayrılır; ve her bir piksele iki seviyeli bir nicemleme uygulanır. Bu yöntemde bloğun

boyu arttıkça nicemleme adımları sabit kalırsa görüntü kalitesi düşer.

### **Vektör Nicemleme (Vector Quantization):**

Vektör nicemleme birçok değer alabilen vektörlerin, daha az sayıda (nicemlenmiş) değerler alabilen bir vektöre eşlenmesidir. Görüntü sıkıştırımda bu vektörler küçük alt görüntüler veya bloklar olabilir. Bu kodlamada amaç, örneğin her bir 4x4'lük blok için bir bayt ile temsil edilen ve kod kitabında 256 giriş için 16 bayt ile temsil edilen kelimeyi denk düşürmektir. Sıkıştırılmış dosya bu kod kitabını içerebilir; veya genelleştirilmiş sabit bir kod kitabından yararlanabilir. Doğal olarak kod kitabını içeren sıkıştırılmış dosya büyük oranda sıkıştırma oranı kazanımını kaybeder.

Bu noktada önemli bir sorun da hangi vektörlerin kod kitabında yer alacağıdır. Görüntü bloklarını en iyi temsil eden vektörleri bulan bir eğitime algoritması ile bu sorun giderilebilir. Algoritma kritik hata değerini optimize edecek bir biçimde tasarlanmalıdır.

### **Farksal Kestirim Kodlaması (Differential Prediction Coding):**

FKK kodlamasında, bir sonraki piksel değerinin önceki piksel değerleri yardımıyla kestirilmesi ve kestirilen değer ile orijinal değer arasındaki farkın kodlanması esasına dayanır. Analog işaretlerde bu yöntem farksal PCM olarak adlandırılır.

Ardışıl piksellerin yüksek derecede birbirleriyle ilişkili olması ve aradaki farkın çok küçük olması esasına dayalı olarak, fark az sayıda bitle temsil edilebilir. Sistemin başlangıç değerine getirilmesi sıkıştırma olmaksızın ilk kestirimin hesaplanması ile sağlanır. Formül olarak ifade edilecek olursa:

$$\tilde{I}(r, c+1) = \rho \hat{I}(r, c) + (1 - \rho) \bar{I}(r, c)$$

$\tilde{I}$  = Kestirilen bir sonraki piksel değeri

$\hat{I}$  = Yeniden yapılandırılmış piksel değeri

$\bar{I}$  = Görüntü için ortalama değer

$\rho$  = Normalize edilmiş ilişki katsayısı

Kestirim tek boyutta (sattır) olabileceği gibi, sattır ve sütunları beraber hesaba katarak, çok boyutlu biçimde de kurgulanabilir. FKK kodlayıcısının sonuçları optimal bir nicemleyici kullanarak kestirim hatasının özel bir dağılım şeklinde nicemlenmesi sağlanabilir.

### **2.4.2 Dönüşüm Domeni Kodlamaları**

Dönüşüm kodlamasında resim, vektörler veya bloklar şeklinde başka bir matematik uzayında ifade edilir. Dönüşüm için birçok matematiksel yapı kullanılabilir; örneğin Fourier, Walsh, Dalgacık dönüşümleri gibi. Günümüzde yoğun olarak kullanılan "ayrık kosinüs dönüşümü –

discrete cosine transform (DCT)- çoğu görüntü için optimal bir seçimdir. Görüntü bloklara veya alt görüntülere bölünerek her biri için dönüşüm hesabı yapılır. Dönüşüm hesaplandıktan sonra bulunan katsayılar nicemlenir ve kodlanır. Bu yöntemin verimli olmasının ana sebebi görüntü enformasyonunun göreceli olarak az sayıda katsayı ile ifade edilip, yüksek frekans bileşenlerinin sıfır olarak nicemlenmesidir. Bu tür dönüşüm işlemi haritalamanın özel bir çeşidi olup, uzamsal frekans düşüncesini temel olarak kabul eder. Görüntü sıkıştırmadaki ana fikir; orijinal verinin matematik uzayında bir başka biçimde haritalanması ve enformasyonun (veya bir başka bakış açısıyla enerjinin) az sayıda katsayı ile temsilidir.

Dönüşüm kodlamasının en basit ve temel yolu “süzgeçleme”dir. Bu yolla bazı yüksek frekans bileşen katsayıları yok edilebilir. Bu işlem tek başına büyük bir sıkıştırma sağlamasa da ek olarak niceme ve kodlama işlemleri ile sıkıştırma oranı artırılır.

Nicemeleme işlemi, “bit ataması” olarak adlandırılan ve her bir katsayının hangi koda karşı düşürüleceğine karar verilmesi esasına dayanır. Tipik olarak alçak frekans bileşenleri daha çok bit ile temsil edilir. ardından bir nicemeleme şeması, örneğin Lloyd-Max, uygulanır. Nicemeleme şeması seçilirken gerçek görüntülerde sıfır ve düşük frekanslı katsayılar görüntü enerjisinin büyük bir kısmını oluşturduğu göz önünde bulundurulur.

Basit süzgeçlemeye ek olarak iki özel tip dönüşüm kodlaması geniş olarak kullanılır: bölgesel (zonal) ve eşik (threshold) kodlama. Bölgesel kodlamada maksimal değişinti (variance) temelinde belirli katsayılar seçilir. Eşik kodlamada ise belirli bir değerin üzerindeki katsayılar seçilerek hesaplama gerçekleştirilir.

En yoğun olarak kullanılan görüntü sıkıştırma standartlarından biri olan JPEG (The Joint Photographic Experts Group) dönüşüm kodlama sınıfında yer alır. JPEG standardı sıkıştırmada temel olarak DCT ve 8x8 piksel bloklarını kullanır. DCT hesaplanmadan önce piksel değerlerinin seviyesi kaydırılarak sıfır merkezli hale getirilir; ardından dönüşüm katsayıları hesaplanır. Bu işlemden sonra dönüşüm katsayıları nicemeleme tablosundaki değerler göre bölünerek nicemeleme adımlarına eşlenir.

Renkli işaretler için JPEG dönüşümü RGB bileşenleri yerine YCrCb renk uzayını kullanır. Renk fark işaretleri olan Cr ve Cb alt örnekleme yapılarak görüntüde parlaklık bilgisinin ayrıntılı biçimde sunulması sağlanır. Bu işlemlerin ardından ileri düzeyde karmaşık entropi kodlaması uygulanır. Sıfır frekanslı katsayılar önceki bloklardaki işlemler de hesaba katılarak farksal tekniklerle kodlanır.(Salamon, 1997)



Uzamsal ve dönüşüm domenlerinin beraber kullanıldığı yöntemler ise “karma -hibrid-” sıkıştırma yöntemler olarak adlandırılır.

Tezde sunulan “dalgacık” fonksiyonları yardımıyla görüntü sıkıştırma da dönüşüm domeni sıkıştırma teknikleri arasında yer alır.



### 3. SÜREKLİ DALGACIK DÖNÜŞÜMÜ

$\psi(x)$ 'in temel dalgacık olabilmesi için gerçek değerli bir fonksiyon olması ve Fourier spektrumu  $\Psi(s)$  olmak üzere kabul edilebilirlik koşullarını sağlaması gerekir (3.1), (3.2).

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(s)|^2}{|s|^2} ds < \infty \quad (3.1)$$

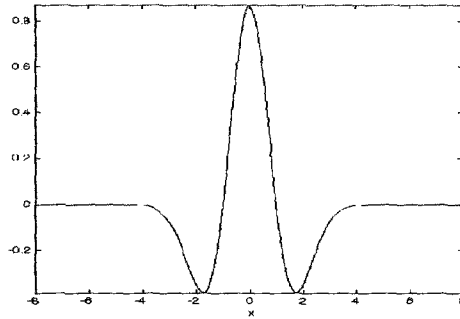
$$\Psi(0) = 0 \Rightarrow \int_{-\infty}^{\infty} \psi(x) dx = 0 \quad \text{ve} \quad \Psi(\infty) = 0 \quad (3.2)$$

Bu koşulları sağlayan dalgacığın genlik spektrumu bant geçiren süzgecin transfer fonksiyonuna benzer. Gerçekte artan frekansla birlikte hızla sifıra inen sıfır ortalamalı impuls cevabına sahip bant geçiren filtreler dönüşüm için temel dalgacık olarak kullanılabilir.

Temel dalgacık fonksiyon kümesi  $\{\psi_{a,b}(x)\}$ , temel dalgacığın ölçeklenmesi ve ötelenmesiyle elde edilir:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad a > 0, b \in \mathbb{R} \quad (3.3)$$

“a” değişkeni ölçekleme (genişlik); b ise x-eksenindeki ötelemeyi temsil eder. Normal gösterimde ise  $\psi(x)$  orijinde yer alır (şekil 3.1).



Şekil 3.1 Temel dalgacık

$f(x)$  fonksiyonunun sürekli dalgacık dönüşümü ise şu şekilde ifade edilir (  $\langle \rangle$  iç çarpım ):

$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x) \cdot \psi_{a,b}(x) dx \quad (3.4)$$

Dalgacık dönüşüm katsayıları dönüştürülen fonksiyonla her bir temel fonksiyonun iç çarpımı (inner product) olarak belirlenir.

Ters dönüşüm ise şu şekilde ifade edilir:

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_f(a,b) \cdot \psi_{a,b}(x) db \frac{da}{a^2} \quad (3.5)$$

(3.3) Eşitliğindeki ölçekleme faktörü temel dalgacık fonksiyonlarının normlarının aşağıdaki eşitlik sağlandığı sürece eşit kalmasını garanti eder.

$$\left\| f\left(\frac{x-b}{a}\right) \right\| = \sqrt{\int_{-\infty}^\infty \left| f\left(\frac{x-b}{a}\right) \right|^2 dx} = \sqrt{a} \cdot \|f(x)\| \quad (3.6)$$

(3.2) Eşitliğinde görüldüğü gibi temel dalgacık fonksiyonu sıfır ortalamaya sahipse, ölçeklenmiş ve ötelenmiş türevleri da aynı şekilde sıfır ortalamalı olacaktır. (Rao, Bopardikar, 1998)

### 3.1 İki Boyutlu Sürekli Dalgacık Dönüşümü

$f(x,y)$  iki boyutlu bir fonksiyon olmak üzere sürekli dalgacık dönüşümü şu şekilde ifade edilir:

$$W_f(a,b_x,b_y) = \int_{-\infty}^\infty \int_{-\infty}^\infty f(x,y) \cdot \psi_{a,b_x,b_y}(x,y) dx dy \quad (3.7)$$

İki boyutlu dönüşümde  $b_x, b_y$  ötelemeyi belirler. İki boyutta ters sürekli dalgacık dönüşümü ise şu şekilde ifade edilir:

$$f(x,y) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty W_f(a,b_x,b_y) \cdot \psi_{a,b_x,b_y}(x,y) db_x db_y \frac{da}{a^3} \quad (3.8)$$

$\psi(x,y)$ , iki boyutlu temel dalgacığı temsil etmektedir. Normalize edilmiş olarak ifade edilirse:

$$\psi_{a,b_x,b_y}(x,y) = \frac{1}{|a|} \psi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right) \quad (3.9)$$

### 3.2 Süzgeç Bankası Yorumu

Temel dalgacık fonksiyonu  $a$  değişkeni ile ölçeklenir ve  $a^{-1/2}$  değeriyle de normalize edilir.

$$\psi_a(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x}{a}\right) \quad (3.10)$$

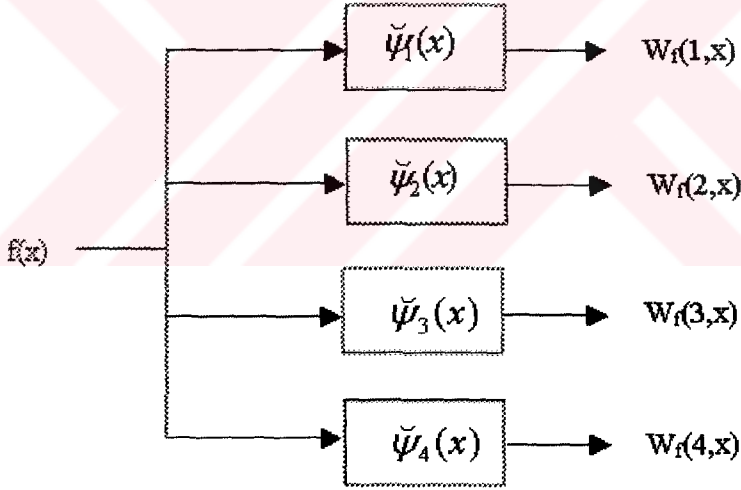
Genel eşitlik “a”nın artan değerleriyle genişleyen fonksiyonlar kümesini oluşturur. Ölçeklenmiş dalgacığın süzgeç bankası karmaşık eşleniği (complex conjugate) ise şu şekilde ifade edilir:

$$\check{\psi}_a(x) = \psi_a^* = \frac{1}{\sqrt{a}} \psi^*\left(-\frac{x}{a}\right) \quad (3.11)$$

Eğer  $\psi(x)$  gerçekte ve çift bir fonksiyonsa yansıma ve eşleniklik bir etkiye sahip değildir. Bu tanımlama yardımıyla sürekli dalgacık dönüşümünü tekrar yazacak olursak:

$$W_f(a, b) = \int_{-\infty}^{\infty} f(x) \cdot \check{\psi}_a(b-x) dx = f * \check{\psi}_a \quad (3.12)$$

Sabit bir a değeri için  $W_f(a, b)$ ,  $f(x)$  ile yansıtılmış eşlenik dalgacığın “a” değerli ölçekteki katlanmasıdır (convolution).



Şekil 3.2 İşaretin dalgacık dönüşümün süzgeç bankası yorumu

Şekil 3.2’de dalgacık dönüşümünün lineer süzgeç bankası yorumunu göstermektedir. “a” değişkeninin her bir değeri farklı bant geçiren süzgeçleri tanımlamakta ve hepsinin oluşturduğu bu banka da dalgacık dönüşümü oluşturmaktadır. 3.5 Eşitliği ise tekrar şu şekilde yazılabilir:

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty [f * \tilde{\psi}_a](b) \cdot \psi_a(b-x) db \frac{da}{a^2} = \frac{1}{C_\psi} \int_0^\infty [f * \tilde{\psi}_a * \psi_a](x) \frac{da}{a^2} \quad (3.13)$$

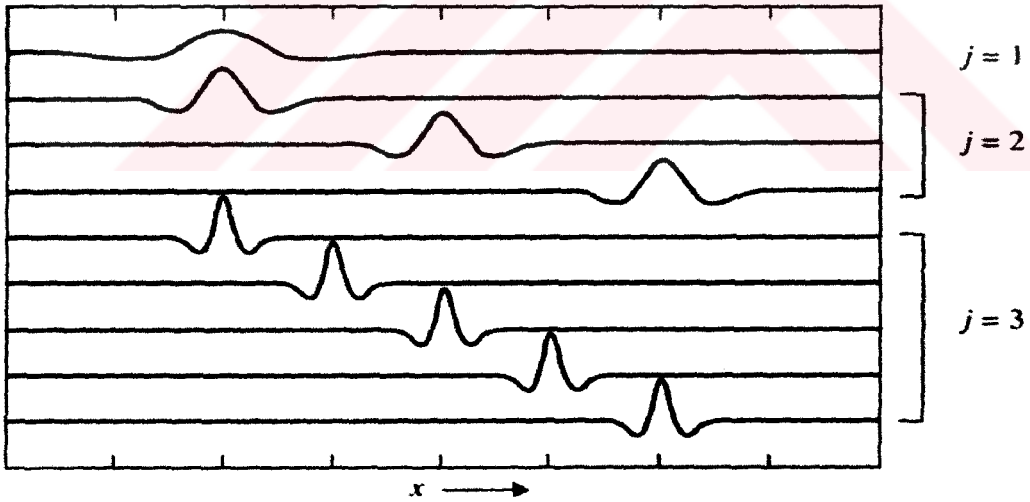
Bu eşitlik süzgeç çıkışlarının tekrar  $\psi_a(x)$  ile süzgeçlenmesi sonucu  $f(x)$  fonksiyonunun tekrar elde edilebileceğini (ters dönüşümü) gösterir. (Castleman, 1996)

### 3.3 Dalgacık Dizisi Yayılımı

#### 3.3.1 Diydik Dalgacıklar

Dalgacık dönüşümünün ikinci tipi ilkinde göre daha sınırlandırılmış olan ve yine aynı şekilde temel dalgacığın ölçeklenmiş ve ötelenmiş biçiminden oluşturulan diydik dalgacıklar kümesidir. Ayrıca ölçekleme ve öteleme gerçek sayılardan değil tamsayılar kümesinden seçilir.

Tanımlanan ikinci dalgacık biçiminde temel fonksiyonlar kümesi oluşturulurken ikili ölçeklemeler ve diydik ötelemeler kullanılır. Diydik ötelemeler  $k/2^j$ 'lik kaydırmalarla elde edilir (Şekil 3.3).



Şekil 3.3 Dalgacığın ikili ölçeklenmesi ve diydik ötelenmesi (Castleman, 1996)

#### 3.3.2 Tanımlama

$\psi_{j,k}(x)$  ortogonal dalgacık olmak üzere şu şekilde tanımlanır:

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad -\infty < j, k < \infty \text{ ve } j, k \in \mathbb{Z} \quad (3.14)$$

$\psi_{j,k}(x)$ ,  $L^2(\mathbb{R})$ 'nin ortonormal temel fonksiyonudur.

$$\langle \psi_{j,k}, \psi_{l,m} \rangle = \delta_{j,l} \cdot \delta_{k,m} \quad l, m \in \mathbb{Z} \quad (3.15)$$

eğer  $f(x) \in L^2(\mathbb{R})$  ise şu şekilde yazılabilir:

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \cdot \psi_{j,k}(x) \quad (3.16)$$

dönüşüm katsayıları ise iç çarpımlardan elde edilir:

$$c_{j,k} = \langle f(x), \psi_{j,k}(x) \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \cdot \psi(2^j x - k) dx \quad (3.17)$$

Eğer  $\psi(x)$  uygun bir biçimde seçilirse fonksiyon dizileri ciddi yaklaşıklık hatası olmaksızın kısaltılabilir.  $f(x)$  sonlu bir aralıkta tanımlıysa ve temel dalgacık iyi bir şekilde yerleştirilirse (orijinden uzaklaştıkça hızla sönümlenirse), katsayıların çoğu büyük  $|k|$  değeri için ihmal edilebilir. Benzer bir yaklaşımla büyük  $|j|$  değerleri için katsayı değerleri oldukça küçük değerler olacaktır.

### 3.3.3 Sıkı (Compact) Diyadik Dalgacıklar

Eğer  $f(x)$  fonksiyonu ve temel dalgacık fonksiyonu  $[0,1]$  aralığı dışında sıfır olacak şekilde sınırlandırılacak olursa ortonormal temel dalgacık fonksiyon ailesi tek bir indisle ( $n$ ) belirlenebilir.

$$\psi_n(x) = 2^{j/2} \psi(2^j x - k) \quad (3.18)$$

$j$  ve  $k$ ,  $n$ 'nin fonksiyonlarıdır:

$$n = 2^j + k \quad j = 0, 1, 2, \dots \quad k = 0, 1, 2, \dots, 2^j - 1 \quad (3.19)$$

Herhangi bir  $n$  değeri için  $j$  en büyük tamsayı değeri olmak üzere  $2^j \leq n$  ve  $k = n - 2^j$  dir.

Ters dönüşüm ise şu şekilde gösterilir:

$$f(x) = \sum_{n=0}^{\infty} c_n \cdot \psi_n(x) \quad \psi_0(x) = 1 \text{ kabul edilir.} \quad (3.20)$$

Dönüşüm katsayıları ise iç çarpım yoluyla şu şekilde tanımlanır:

$$c_n = \langle f(x), \psi_n(x) \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \cdot \psi(2^j x - k) dx \quad (3.21)$$

Bu yolla sürekli bir fonksiyon sonsuz bir dizi şeklinde, Fourier dizisinde olduğu gibi sunulabilir. Eğer bir veya birkaç  $\psi_n(x)$ ,  $f(x)$ 'in ana bileşenleriyle benzerlik gösteriyorsa kabul edilebilir bir hatayla dizi kısaltılıp birkaç terimle ifade edilebilir.

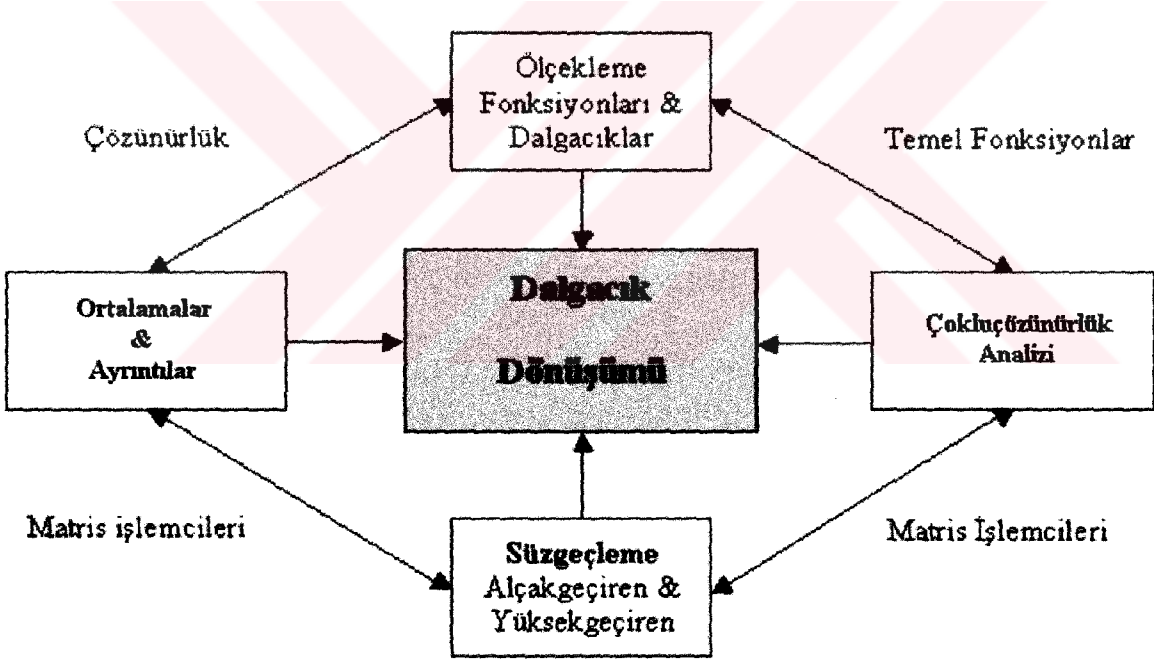


#### 4. AYRIK DALGACIK DÖNÜŞÜMÜ VE GÖRÜNTÜ SIKIŞTIRMA

Dalgacık fonksiyonları ile görüntü sıkıştırma ,dönüşüm sınıfı sıkıştırma yöntemleri arasında yer alır. Bu yöntem dalgacık dönüşüm alanında muhafaza edilen enformasyonun ölçekte fazlalıklarının azaltılması esasına dayanır.

Dalgacıklarla görüntü sıkıştırılması tekniği diğer dönüşüm sınıfı sıkıştırma teknikleriyle aynı esasa dayanır. İlk adımda görüntüye dalgacık dönüşümü uygulanır ve katsayıların bir kısmı dönüştürülmüş görüntüden atılır. Kodlama geriye kalan katsayılara uygulanır. Sıkıştırılmış görüntü çözülen katsayılardan tekrar oluşturulur ve ters dönüşüm uygulanarak orijinal görüntü yeniden elde edilir.(Rioul, Vetterli, 1991)

Bu alanda literatür dalgacıklar üzerine değişik yaklaşımlar sunar. Matematiksel referanslar sürekli ölçeklemenin belirlenmesi ve dalgacık fonksiyonları arasında çokluçözünürlük (multiresolution) analizi geliştirir. Bu bölümde ortalama ayrıntı fonksiyonlarının görüntü sıkıştırma konusundaki temel fikir incelenecektir (Şekil 4.1)



Şekil 4.1 Dalgacık Analizi ve Dönüşüm arasındaki ilişki

##### 4.1 Ortalama ve Ayrıntılar

Görüntü sıkıştırma konusunda en basit yaklaşım görüntünün piksellerinin değerlerinin ortalama alınarak yer değiştirilmesidir. Bu yaklaşımla önemli bir sıkıştırma elde edilebilir;



fakat görüntü kalitesi oldukça düşer. Aynı zamanda orijinal görüntü tekrar elde edilmek istenirse ne oranda bir ayrıntı kaybının olduğunun bilinmesi gerekir. Buna daha yakından bakıp konuyu bu noktadan daha ilerilere taşımak için iki pikselden oluşan  $\{x_1, x_2\}$  bir görüntüyü ele alalım. Bu iki değer ortalama ve ayrıntı şeklinde ifade edilebilir:

$$\begin{aligned} a &= (x_1 + x_2)/2 & d &= (x_1 - x_2)/2 \\ x_1 &= a + d & x_2 &= a - d \end{aligned} \quad (4.1)$$

Orijinal dizinin  $\{x_1, x_2\}$  dalgacık dönüşümü şimdi  $\{a, d\}$  olarak ifade edilebilir. Bu noktada hiçbir enformasyon kaybedilmiş veya kazanılmış değildir. Özel bir kazanımı görülmemekle beraber  $x_1$  ve  $x_2$  birbirine değer olarak yakınsa “d” oldukça küçük olacak ve resim sadece bir  $\{a\}$  yaklaşımıyla sunulabilir. Burada dikkat edilmesi gereken “sıkıştırılmış görüntünün” orijinal görüntüden daha az pikselle ifade edilmiş olmasıdır. Yeniden elde edilen görüntü  $\{a, a\}$  ; hata ise  $\{|x_1 - a|, |x_2 - a|\} = \{|d|, |d|\}$  olarak bulunur. Eğer “d” oldukça küçük ise hatanın da küçük olacağı açıktır.

Dalgacık analizinde temel olan ,değişik seviyelerdeki ayrıntılardan enformasyonun tekrar elde edilmesi fikridir. Bu ayrıntılar ölçek ve çözünürlük enformasyonu olarak düşünülebilir. Verilen örnek gerçek görüntüler için sınırlı bir kullanılılığa sahip olmakla beraber ayrıntı enformasyonunun tanımlanması ve ufak değerlerde ise atılması durumunda elde edilen sonuç görüntüsünün fazla değişmeyeceği konusunda açık bir fikir vermektedir.

Bir sonraki adım olarak biraz daha büyük bir görüntüyü  $\{x_1, x_2, x_3, x_4\}$  inceleyerek konuyu açalım. Bu durumda ortalamalar ve farklar şu şekilde ifade edilebilir:

$$\begin{aligned} a_{1,0} &= (x_1 + x_2)/2 & a_{1,1} &= (x_3 + x_4)/2 \\ d_{1,0} &= (x_1 - x_2)/2 & d_{1,1} &= (x_3 - x_4)/2 \end{aligned} \quad (4.2)$$

ortalama ve ayrıntılardaki çift indisler bir işlemdeki adım sayısını belirtmektedir. Orijinal resmin yeni sunumu  $\{a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1}\}$ 'dir. Sıkıştırma oranını daha da arttırmak amacıyla ortalamalara tekrar aynı işlemi uygularız:

$$a_{0,0} = (a_{1,0} + a_{1,1})/2 \quad \text{ve} \quad d_{0,0} = (a_{1,0} - a_{1,1})/2 \quad (4.3)$$

Bu durumda  $d_{0,0}$  değeri yeterince küçükse orijinal görüntü sadece  $\{a_{0,0}\}$  olarak ifade edilebilir. “ $a_{0,0}$ ” Orijinal görüntüyü oluşturan bütün piksellerin ortalamasıdır, aynı zamanda

görüntü enformasyonunun en kaba seviyesini temsil eder. “ $a_{1,0}$ ” ve “ $a_{1,0}$ ” birlikte enformasyonun bir yüksek çözünürlükteki veya daha küçük ölçekteki sunumudur ve de  $\{a_{0,0}, d_{0,0}\}$  değerlerine dönüştürülüp, tekrar bu değerlerden elde edilebilir. İlk baştaki orijinal pikseller de  $\{a_{1,0}, d_{1,0}\}$  ve  $\{a_{1,1}, d_{1,1}\}$  değerlerinden elde edilebilir. Sonuçta  $\{a_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}\}$  dizisinden orijinal pikseller tekrar elde edilebilir. Görüntüyü ne miktarda sıkıştırılacağına belirlenen kriterlere bağlı olarak ayrıntı değerlerini ihmal ederek karar verilir.

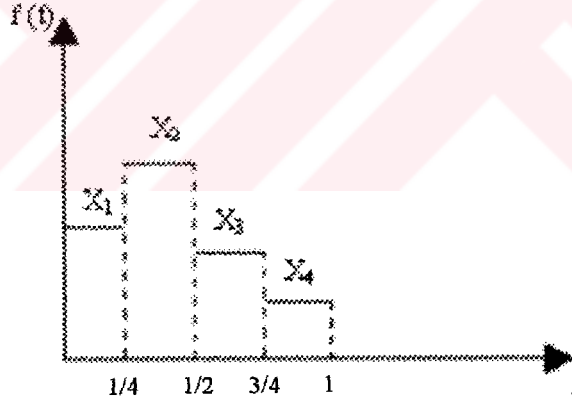
## 4.2 Ölçekleme Fonksiyonları ve Dalgacık Fonksiyonları

Görüntü dizimiz  $\{x_1, x_2, x_3, x_4\}$  birim aralıkta tanımlı bir fonksiyon olsun:

$$f(t) = x_1 X_{[0,1/4)}(t) + x_2 X_{[1/4,1/2)}(t) + x_3 X_{[1/2,3/4)}(t) + x_4 X_{[3/4,1)}(t) \quad (4.4)$$

$$X_{[a,b)}(t) = \{1, a \leq t < b; 0 \text{ diğer}\} \quad (4.5)$$

Şekil (4.2) keyfi seçilmiş  $x_n$  değerlerinden parçalı sabit fonksiyonlarla ifadesidir. Dikkat edilecek olursa  $f(t)$  sürekli fonksiyonun bir yaklaşık ifadesini vermektedir. Daha iyi bir yaklaşıklık ifadesi tanımlı aralıkları daha da daraltılarak elde edilebilir.



Şekil 4.2 Parçalı fonksiyon yaklaşımı

Bu sabit fonksiyonlar birbirlerinin ölçeklenmiş ve ötelenmiş türevleridir:

$$X_{[0,1/4)}(t) = X_{[0,1)}(2^2 t) \quad (4.6)$$

Şekil (4.2)'deki gibi karakteristik fonksiyonlar, bu parçalı, ölçeklenmiş ve ötelenmiş fonksiyonlar cinsinden ifade edilebilir:

$$\phi(t) = X_{[0,1)}(t)$$

$$\phi_{j,k}(t) = \phi(2^j t - k) \quad k = 0, \dots, 2^j - 1 \quad (4.7)$$

Bu tanımlama ile  $\phi(t)$  ölçekleme fonksiyonlarının ölçeklenmiş ve ötelenmiş türevlerini inceleyecek olursak:

$$\phi_{0,0}(t) = \phi(t)$$

$$\phi_{1,0}(t) = \phi(2t) = \{1, 0 \leq t < 1/2 ; 0 \text{ diğ}er\}$$

$$\phi_{1,1}(t) = \phi(2t - 1) = \{1, 1/2 \leq t < 1 ; 0 \text{ diğ}er\} \quad (4.8)$$

Bu tanımlamalar yardımıyla parçalı yaklaşıklık fonksiyonunu ifade edecek olursak:

$$f(t) = x_1 \phi_{2,0}(t) + x_2 \phi_{2,1}(t) + x_3 \phi_{2,2}(t) + x_4 \phi_{2,3}(t) \quad (4.9)$$

$\phi_{j,k}$  fonksiyonunun sıfır olmadığı aralığa fonksiyonun desteği denir; “j” değeri azaldıkça  $\phi_{j,k}$  fonksiyonunun genişliği dolayısıyla “desteği” artar. Diğ}er bir anlatımla  $\phi_{j+1,k}$ ’in destek genişliği  $\phi_{j,k}$  desteğinin yarısıdır. Sonuç olarak ölçek veya çözünürlük  $\phi_{j,k}$  fonksiyonundaki “j” değeri tarafından belirlenir.

Bu noktadan itibaren daha önce yaptığımız yordamı genişleterek  $f(t)$  fonksiyonunu ortalama ve fark fonksiyonları cinsinden ifade edelim. Ortalama, fonksiyonu düşük çözünürlükle ifade etmektir ve  $\phi_{j,k}$  fonksiyonundaki “j” değerinin azaltılmasıyla elde edilir.  $f(t)$  fonksiyonunu  $\phi_{1,0}$  ve  $\phi_{1,1}$  fonksiyonları cinsinden ifade edersek (4.10) eşitliği elde edilir:

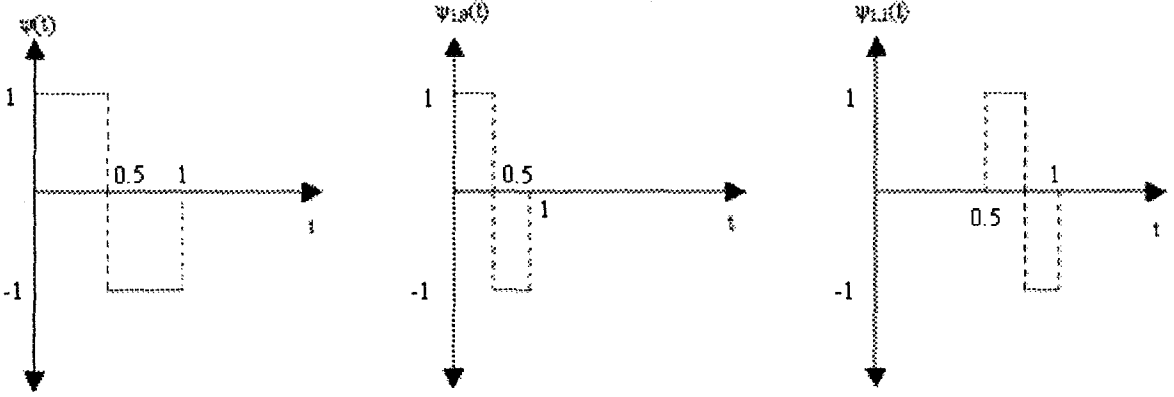
$$g_1(t) = a_{1,0} \phi_{1,0}(t) + a_{1,1} \phi_{1,1}(t) \quad (4.10)$$

Elde edilen fonksiyon  $f(x)$ ’e benzemektedir; fakat bu bir yaklaşıklık ( $f(1/8) = x_1$  fakat  $g_1(1/8) = a_{1,0} = (x_1 + x_2)/2$  değerindedir. ) ifade eden bir fonksiyondur. Eđer  $x_1, x_2$ ’ye eşit değılse enformasyon kaybı oluşur. Fakat herhangi bir kayıp olması istenmiyorsa bu farkı ifade edecek bir fonksiyon yani “dalgacık fonksiyonlarına” ihtiyaç duyulur.

$$\psi(t) = X_{[0,1/2)}(t) - X_{[1/2,1)}(t) = \begin{cases} 1 \Rightarrow 0 \leq t < 1/2 \\ -1 \Rightarrow 1/2 \leq t < 1 \\ 0 \Rightarrow \text{diğ}er \end{cases} \quad (4.11)$$

$$\psi_{j,k}(t) = \psi(2^j t - k) \quad k = 0, \dots, 2^j - 1 \quad (4.12)$$

Temel dalgacı ve ölçeklenmiş ve ötelenmiş dalgacıkları şekil 4.3'de şu şekilde gösterebiliriz:



Şekil 4.3 Temel dalgacık ve ölçeklenmiş ve ötelenmiş türevleri

$f(x)$  fonksiyonunu ortalama ve fark fonksiyonları cinsinden ifade edelim.

$[0, 1/4)$  aralığında:

$$f(t) - g_1(t) = x_1 - a_{1,0} = d_{1,0} \quad (4.13)$$

$[1/4, 1/2)$  aralığında:

$$f(t) - g_1(t) = x_2 - a_{1,0} = -d_{1,0} \quad (4.14)$$

Bu sonuçlardan yararlanarak benzer yaklaşımlarla sırasıyla  $[0, 1/2)$  aralığı için (4.15) ve  $[1/2, 1)$  aralığı için (4.16) ifadeleri yazılabilir:

$$f(t) = a_{1,0}\phi_{1,0}(t) + d_{1,0}\psi_{1,0}(t) \quad (4.15)$$

$$f(t) = a_{1,1}\phi_{1,1}(t) + d_{1,1}\psi_{1,1}(t) \quad (4.16)$$

$[0, 1)$  aralığında genel ifade ise şu şekilde yazılabilir:

$$f(t) = a_{1,0}\phi_{1,0}(t) + a_{1,1}\phi_{1,1}(t) + d_{1,0}\psi_{1,0}(t) + d_{1,1}\psi_{1,1}(t) \quad (4.17)$$

Bu sefer aynı adımları  $[0, 1)$  aralığında tüm ortalama ifadesi düzenleyip, (4.17)'deki ilk ki terimi  $a_{0,0}\phi_{0,0}(t)$  ile değiştirip fark terimlerini bırakacak olursak:

$$g_0(t) = a_{0,0}\phi_{0,0}(t) + d_{1,0}\psi_{1,0}(t) + d_{1,1}\psi_{1,1}(t) \quad (4.18)$$

$[0, 1/4)$  aralığında  $f(t)$  ve  $g_0(t)$ 'nin farkı:

$$f(t)-g_0(t)=x_1-a_{0,0}-d_{1,0}=d_{0,0} \quad (4.19)$$

[0,1/4) aralığında:

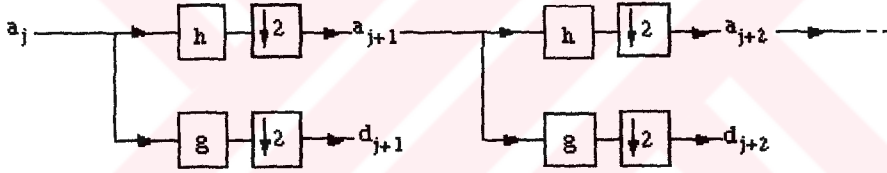
$$f(t)-g_0(t)=x_2-a_{0,0}+d_{1,0}=d_{0,0} \quad (4.20)$$

Benzer şekilde [1/2,1) aralığında  $f(t)-g_0(t)=-d_{0,0}$  olarak ifade edilebilir. Sonuçta tüm ifadeyi verirsek;

$$f(t)=a_{0,0}\phi_{0,0}(t)+d_{0,0}\psi_{0,0}(t)+d_{1,0}\psi_{1,0}(t)+d_{1,1}\psi_{1,1}(t) \quad (4.21)$$

(4.21) Eşitliği 4.1 bölümünde açıklanıp ulaşılan sonuç dizisinin  $\{a_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}\}$  fonksiyon benzeri durumuna denk düşmektedir. (Welstead, 1999)

Ortalama ve fark katsayılarının elde edilmesi alçak geçiren ve yüksek geçiren süzgeçlerin uygulanması ve ardından alt örnekleme işlemi ile gerçekleştirilir. Alçak geçiren süzgeç (h) çıkışında alt örnekleme ile beraber ortalama katsayılar elde edilir; Yüksek geçiren süzgeç (g) çıkışında ise alt örnekleme işleminden sonra fark katsayıları elde edilerek bu değerler saklanır. Bu işlem şekil 4.4'te gösterilmiştir.



Şekil 4.4 Alçak geçiren (h) ve Yüksek geçiren (g) süzgeçlerle katsayıların elde edilmesi

### 4.3 Çokluçözünürlük Analizi (Multiresolutional Analysis)

Ayrık değerli dizilerin – sayısal görüntülerde olduğu gibi – değişik ölçeklerde ortalama ve ayrıntı değerlerine ayrıştırılması işlemine çokluçözünürlük analizi denir. Dalgacıklarla görüntü sıkıştırma işlemi anlayabilmek için temel bir kavramdır.

$V^0$ , [0,1) aralığında sabit bütün fonksiyonları gösterir ve fonksiyonların “vektör uzayı” olarak adlandırılır. İki sabit fonksiyonu bu tanımlı aralıkta toplarsak sonuç yine sabit bir fonksiyon olacağından yine  $V^0$  uzayında yer alır. Aynı şekilde fonksiyonu skaler bir değerle çarparsak sonuç aynen  $V^0$  uzayında tanımlı olacaktır. Temel ölçekleme fonksiyonu  $\phi(t)$  de  $V^0$ 'ın bir üyesidir. Gerçekte  $V^0$ 'ın herhangi bir üyesi,  $\phi(t)$  fonksiyonun uygun bir değerle çarpımından elde edilebilir. Bu nedenle  $\{\phi(t)\}$ ,  $V^0$  vektör uzayı için bir “temel” oluşturur.

$V^1$  de [0,1/2) ve [1/2,1) arsında tanımlı parçalı sabit fonksiyonların vektör uzayını oluşturur.

$V^1$ 'in bir elemanı (4.10) eşitliğinde tanımlana  $g_1(t)$  fonksiyonudur. Ölçekleme fonksiyonları  $\phi_{1,0}(t)$  ve  $\phi_{1,1}(t)$  de  $V^1$ 'in elemanlarıdır.  $V^1$  vektör uzayının diğer elemanları da  $\phi_{1,0}(t)$  ve  $\phi_{1,1}(t)$ 'in lineer kombinasyonu şeklinde ifade edilebilir ve bu nedenle  $V^1$  uzayı için temel kabul edilir.  $V^0$ 'ın herhangi bir elemanı aynı zamanda  $V^1$  uzayının bir elemanıdır.

$$V^0 \subset V^1 \quad (4.22)$$

Bu tarzda  $V^2, V^3 \dots$  uzayları tanımlanabilir.  $V^n$  vektör uzayı  $1/2^n$  eşit uzunluklu aralıklı parçalı fonksiyonlardan oluşur.  $V^n$  vektör uzayı için ölçekleme fonksiyonu  $\{\phi_{n,j}(t) \quad j=1,2,\dots,2^n-1\}$  bir temel ifade eder.  $V^n$  uzayı (4.23)'deki iç içe alt kümeler özelliğini sağlar.

$$V^0 \subset V^1 \subset \dots \subset V^n \subset V^{n+1} \subset \dots \quad (4.23)$$

### 4.3.1 İç Çarpım, Ortogonallik, Ortonormallik

$V^n$  fonksiyon uzayı için "iç çarpım" kavramını tanımlayacak olursak (Polikar, 1996):

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t) \cdot g(t) dt \quad (4.24)$$

Bir iç çarpım ile donatılan vektör uzayı "iç çarpım uzayı" olarak adlandırılır. İç çarpım kavramıyla iki fonksiyonun "ortogonallik" tanımı şu şekilde yapılabilir:

$$\langle f \cdot g \rangle = 0 \quad (4.25)$$

Ortogonallik birçok sebepten ilgi alanımıza girmektedir. Örneğin:

$$\langle \phi_{1,0}, \phi_{1,1} \rangle = 0 \quad (4.26)$$

$V^1$  uzayı için  $\{\phi_{1,0}, \phi_{1,1}\}$  fonksiyonları ortogonal temel oluşturur.

Genel olarak her  $j$  ve  $k \neq l$  için (4.27) ifadesi yazılabilir.

$$\langle \phi_{j,k} \cdot \phi_{j,l} \rangle = 0 \quad (4.27)$$

Bu sebeple  $\{\phi_{j,k}; k=0, \dots, 2^j-1\}$ ,  $V^j$  fonksiyon uzayı için karşılıklı ortogonal temel vektörlerdir.

Aynı yaklaşımla bu dalgacık fonksiyonları için de yazılabilir:

$$\langle \psi_{j,k} \cdot \psi_{j,l} \rangle = 0 \quad k \neq l \quad (4.28)$$

Ortogonallik diğer bir sebepten daha ilgi alanımıza girmektedir. verilen bir  $U$  iç çarpım

uzayının daha geniş bir S uzayının alt kümesi olduğunu varsayarsak S'teki vektör kümelerinin U alt kümesindeki tüm vektörlere ortogonal olduğu söylenebilir. Bu küme U'nun S'teki ortogonal tamamlayıcısı (complement) denir.

$$W^j \equiv \{h \in V^{j+1} : (h, f) = 0 \text{ tüm } f \in V^j\} \quad (4.29)$$

$W^j$ ,  $V^j$  vektör uzayının  $V^{j+1}$  uzayındaki ortogonal tamamlayıcısı olarak tanımlanır. Dikkat edilmesi gerekir ki,  $\psi_{j,k}$ 'nın sabit olduğu aralık  $V^j$  uzayının elemanların sabit olduğu aralığın yarısıdır. Diğer bir anlatımla  $\psi_{j,k} \in V^{j+1}$  her k için;  $\langle \psi_{j,k}, f \rangle = 0$  her  $f \in V^j$  ve  $\psi_{j,k} \in W^j$  her j,k için denilebilir. Örneğin:

$$f \equiv f_0 \phi_{1,0} + f_1 \phi_{1,1} \in V^1 \text{ } f_0 \text{ ve } f_1 \text{ skaler sabitlerdir.}$$

$$\langle \psi_{1,0}, f \rangle = f_0 \langle \psi_{1,0}, \phi_{1,0} \rangle + f_1 \langle \psi_{1,0}, \phi_{1,1} \rangle = 0 + 0 = 0$$

$$\langle \psi_{1,0}, \phi_{1,0} \rangle = \int_0^{1/4} 1 \cdot 1 dt + \int_{1/4}^{1/2} -1 \cdot 1 dt = 0$$

$$\langle \psi_{1,0}, \phi_{1,1} \rangle = \int_{1/2}^1 0 \cdot 1 dt = 0$$

Böylece  $\psi_{1,0} \in W^1$  ve benzer olarak  $\psi_{1,1} \in W^1$  yazılabilir ve  $\psi_{j,k} \in W^j$  tezi doğrulanmış olur. Açıkça  $W^j$  uzayı,  $V^{j+1}$  uzayından büyük olamaz ( $W^j \in V^{j+1}$ ) ; dolayısıyla  $W^j$  uzayının boyutları da  $2^{j+1}$ 'den büyük olamaz.  $\{\psi_{j,k}; k=0, \dots, 2^j-1\}$ ,  $2^j$  adet karşılıklı ortogonal, dolayısıyla bağımsız, fonksiyon kümesi oluşturduğundan,  $W^j$ 'nin boyutu da en az  $2^j$  olur. Sonuç olarak  $\{\phi_{j,k}; k=0, \dots, 2^j-1\}$  vektör kümesi,  $\{\psi_{j,k}; k=0, \dots, 2^j-1\}$  vektör kümesindeki her bir elamana ortogonaldir.

Dalgacıklar burada verilmiş bir çözünürlükte tanımlı uzay fonksiyonlarının ortogonal tamamlayıcısı olarak bir temel oluşturur. Bu ortogonal tamamlayıcı bir çözünürlük seviyesinden daha düşük çözünürlük seviyesine geçerken kaybolan ayrıntıları ifade etmekte kullanılır. Sonuç olarak eldeki yüksek çözünürlük uzayı dalgacık fonksiyonları ile alternatif temelde (4.30)'daki gibi ifade edilebilir:

$$\{\phi_{j,0}, \dots, \phi_{j,2^j-1}, \psi_{j,0}, \dots, \psi_{j,2^j-1}\} \quad (4.30)$$

Örneğin  $g_{j+1} \in V^{j+1}$  uzayında orijinal olarak ifade edilsin (4.31). Ardından bu ifadeyi (4.30)'deki alternatif temelin terimleri cinsinden yazalım (4.32).

$$g_{j+1} = a_{j+1,0} \phi_{j+1,0} + \dots + a_{j+1,2^{j+1}-1} \phi_{j+1,2^{j+1}-1} \quad (4.31)$$

$$g_{j+1} = a_{j,0} \phi_{j,0} + \dots + a_{j,2^j-1} \phi_{j,2^j-1} + d_{j,0} \psi_{j,0} + \dots + d_{j,2^j-1} \psi_{j,2^j-1} \quad (4.32)$$

Elde edilen fark katsayıları  $\{d_{j,0}, \dots, d_{j,2^j-1}\}$  dalgacık dönüşümünün bir parçasıdır. Bir sonraki adım  $g_j$ 'nin  $\{\phi_{j-1,k}\}$  ve  $\{\psi_{j-1,k}\}$  terimleriyle ifade edilmesidir. (Welstead, 1999)

$$g_j = a_{j,0} \phi_{j,0} + \dots + a_{j,2^j-1} \phi_{j,2^j-1} \in V^k \quad (4.33)$$

### 4.3.2 Normalizasyon

Bu çalışmada normalize ölçek ve dalgacık fonksiyonları ile çalışmak daha uygun bir yoldur (Polikar, 1996). İç çarpım uzayı bir vektörün normunu tanımlayacak olursak:

$$\|f\| = \sqrt{\langle f, f \rangle} \quad (4.34)$$

İç çarpım uzayındaki “u” vektörü  $\|u\|=1$  ise normalize edilmiştir. normalize “u” vektörü sıfır olmayan vektörün, vektörün normuna bölünmesiyle elde edilir. ( $u = f/\|f\|$ )

$\{\phi_{j,k}\}$  ve  $\{\psi_{j,k}\}$  normlarını bu yolla belirleyecek olursak;

$$\|\phi_{j,k}\|^2 = \int_0^1 \phi_{j,k}^2(t) dt = \frac{1}{\sqrt{2^j}} \text{ ise,}$$

$$\|\phi_{j,k}\| = \frac{1}{\sqrt{2^j}} \text{ ve } \|\psi_{j,k}\| = \frac{1}{\sqrt{2^j}} \quad k=0, \dots, 2^k-1$$

bu durumda normalize edilmiş ölçekleme ve dalgacık fonksiyonları (4.35)'te belirtilmiştir.

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \text{ ve } \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad k=0, \dots, 2^k-1 \quad (4.35)$$

### 4.4 Dalgacık Dönüşümü

$2^n$  Noktadan oluşan tanımlı diziyi  $V^n$  uzayında tanımlı fonksiyonlar cinsinden ifade edelim:

$$f(t) = x_1 \phi_{n,0}(t) + \dots + x_{2^n} \phi_{n,2^n-1}(t) \quad (4.36)$$

dalgacık dönüşümü için ilk adım bu fonksiyonu  $V^n$ 'in alternatif temelinde ifade etmektir.

$$f(t) = a_{n-1,0} \phi_{n-1,0}(t) + \dots + a_{n-1,2^{n-1}-1} \phi_{n-1,2^{n-1}-1}(t) + d_{n-1,0} \psi_{n-1,0}(t) + \dots + d_{n-1,2^{n-1}-1} \psi_{n-1,2^{n-1}-1}(t) \quad (4.37)$$

Dalgacık dönüşümü sonucu elde edilen fark katsayıları saklanır, dönüşüm işlemi aynı şekilde



kalan ortalama katsayılarına tekrar uygulanır.

Bu noktada bu katsayıların elde edilmesi için ortogonalliğin önemli rolünün hatırlanması gerekir. Aynı fonksiyon uzayının elemanları olan ölçek ve dalgacık fonksiyonlarının hepsi birbirine ortogondur (tüm  $\phi_{n,j}(t)$  ve tüm  $\psi_{n,j}(t)$  fonksiyonları). Ortogonallik ve normalizasyon özelliklerini kullanarak gerekli katsayıları elde edilebilir.

$$\int_0^1 f(t)\phi_{n-1,j}(t).dt = a_{n-1,j} \quad \text{ve} \quad \int_0^1 f(t)\psi_{n-1,j}(t).dt = d_{n-1,j} \quad (4.38)$$

Bulunan bu katsayılar genel olarak, (4.37) eşitliği için şu şekilde yazılabilir;

$$a_{n-1,j} = (x_{2^{j+1}} + x_{2^{j+2}})/\sqrt{2} \quad \text{ve} \quad d_{n-1,j} = (x_{2^{j+1}} - x_{2^{j+2}})/\sqrt{2} \quad j=0, \dots, 2^{n-1}-1 \quad (4.39)$$

(4.39) İfadelerinin matrisel gösterimi daha kolay ve kullanışlı bir yaklaşım olacaktır.

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ x_{2^n} \end{bmatrix} = \begin{bmatrix} a_{n-1,0} \\ \dots \\ a_{n-1,2^{n-1}-1} \\ d_{n-1,0} \\ \dots \\ d_{n-1,2^{n-1}-1} \end{bmatrix} \quad (4.40)$$

$$A_n = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & 0 & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (4.41)$$

$$D_n = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \quad (4.42)$$

$A_n$  ve  $D_n$ ,  $2^{n-1} \times 2^{n-1}$ lik matrislerdir.  $R^{2^n} \rightarrow R^{2^{n-1}}$ ;  $A_n$ , ortalama işlemcisi,  $D_n$  ise fark işlemcisi olarak adlandırılır. Vektör notasyonu ile katsayıların bulunmasını şu şekilde ifade edebiliriz:

$$\begin{bmatrix} A_n \\ D_n \end{bmatrix} x = \begin{bmatrix} a_{n-1} \\ d_{n-1} \end{bmatrix} \quad (4.43)$$

(4.43) eşitliğinden fark katsayıların yarısı elde edilir ( $d_{n-1}$ ), bir sonraki adımda ise işlemciler daha düşük çözünürlükteki ortalama katsayılara uygulanır.

$$\begin{bmatrix} A_n \\ D_n \end{bmatrix} a_{n-1} = \begin{bmatrix} a_{n-2} \\ d_{n-2} \end{bmatrix} \quad (4.44)$$

Bu yolla ortalama ve fark işlemleriyle gerekli fark katsayıları elde edilir [ $a_0, d_0, \dots, d_{n-1}$ ]. 128x128 Lenna görüntüsüne iki seviyeli dönüşüm uygulayarak elde edilen tek boyuttaki 32 ortalama ( $a_0$ ), 32 ikinci seviye ayrıntı ( $d_0$ ) ve 64 ilk seviye ayrıntı ( $d_1$ ) iki boyutlu olarak uygulanırsa elde edilen görüntü şekil 4.5 de gösterilmiştir. Elde edilen küçük katsayılar alt bantlardaki siyah bölgelere denk düşmektedir. Fakat resmin kenar geçişlerinde fark katsayılarının değeri büyüdüğünden bu alt bantlarda geçişler belli olmaktadır. (Valens, 1999)



Şekil 4.5 128x128 Lenna görüntüsünün iki boyutlu Haar dalgacık dönüşümü

#### 4.5 Ters Dalgacık Dönüşümü

Dalgacık dönüşümünün görüntü sıkıştırma işleminde yararlı olabilmesi için aynı işlemlerin tersten uygulanabilir olması gerekir. Matris gösterimiyle devam edecek olursak; yüksek çözünürlüklü ortalama değerleri ,bir düşük seviyedeki ortalama ve ayrıntı katsayıları yardımıyla elde edilir.

$$a_n = \begin{bmatrix} A_n \\ D_n \end{bmatrix}^{-1} \begin{bmatrix} a_{n-1} \\ d_{n-1} \end{bmatrix} \quad (4.45)$$

Ters dönüşüm matrisi incelendiğinde ileri yönlü dönüşümde kullanılan matrisin eşleniğidir.

$$\left[ \begin{array}{c} A_n \\ D_n \end{array} \right]^{-1} = \left[ A_n^* | D_n^* \right] \quad *, \text{ matrisin eşleniğini belirtir.} \quad (4.46)$$

$$a_n = \left[ \begin{array}{c} A_n \\ D_n \end{array} \right]^{-1} \left[ \begin{array}{c} a_{n-1} \\ d_{n-1} \end{array} \right] = \left[ A_n^* | D_n^* \right] \left[ \begin{array}{c} a_{n-1} \\ d_{n-1} \end{array} \right] = A_n^* a_{n-1} + D_n^* d_{n-1} \quad (4.47)$$

(4.47) Eşitliği yardımıyla  $a_n$  ortalama katsayıları,  $a_{n-1}$  ve  $d_{n-1}$  katsayılarından tekrar elde edilir.

## 4.6 Daubechies Dalgacıkları

Tezin bu bölümünde bir önceki bölümde yapılan çalışmalar doğrultusunda ağırlıklı ortalamalar ve farklar kavramları eklenip, değişik dalgacık sistemleri incelenecektir.

### 4.6.1 Ağırlıklı Ortalamalar ve Farklar

Görüntülerde olduğu gibi bir veri dizisi  $\{x_1, x_2, \dots, x_n\}$  olduğunu varsayalım. Haar dalgacıklarında geliştirilen basit nümerik ortalamalar yerine ağırlıklı ortalama katsayıları olsun;

$$c_0 x_1 + c_1 x_2 + \dots + c_{N-1} x_N \quad N > 2 \text{ ve } \{c_k = 0, k < 0 \text{ ve } k \geq N\}$$

benzer yaklaşımla basit fark işlemi ise;

$$c_1 x_1 - c_0 x_2 + c_3 x_3 - c_2 x_4 + \dots$$

Bu özelliği sağlayan katsayı seçimleri (4.48) ortogonallik özelliğini de sağlamaktadır. (Daubechies, Antonini, Barlaud, 1992)

$$(c_0, c_1, c_2, c_3, \dots) \bullet (c_1, -c_0, c_3, -c_2, \dots) = 0 \quad (4.48)$$

### 4.6.2 Alçak Geçiren ve Yüksek Geçiren Süzgeçleme

$x_n$  dizisi  $n=2^m$  ve  $m > 0$  olmak üzere periyodik bir dizi olsun.

$$\dots, x_1, x_2, \dots, x_n, x_1, x_2, \dots, x_n, x_1, x_2, \dots$$

$$x_0 = x_n \quad x_{-1} = x_{n-1} \quad x_{n+1} = x_1$$

$L_n: \mathbb{R}^n \rightarrow \mathbb{R}^{n/2}$  alçak geçiren süzgeç operatörünü tanımlarsa:

$$(L_n x)_i = \sum_{j=1}^n c_{j-2i+1} x_j, \quad i=1, 2, \dots, n/2 \quad (4.49)$$

$H_n: \mathbb{R}^n \rightarrow \mathbb{R}^{n/2}$  yüksek geçiren süzgeç operatörünü tanımlarırsa;

$$(H_n x)_i = \sum_{j=1}^n (-1)^{j+1} c_{2i-j} x_j, \quad i=1, 2, \dots, n/2 \quad (4.50)$$

$H_n$  ve  $L_n$  birlikte "Quadrature Mirror Filter -QMF-" olarak adlandırılırlar.

### 4.6.3 Matrisel Biçimde Gösterim

$L_n[i,j]$ , alçak geçiren süzgeç işlemcisinin matris gösteriminin  $(i,j)$  elemanını belirtir.

$$L_n[i,j] = \begin{cases} c_{j-2i+1} & , \quad 0 \leq j-2i+1 \leq N-1 \\ c_{j-2i+1+n} & , \quad j-2i+1+n < 0 \\ 0 & , \quad j-2i+1 \geq N \end{cases}$$

$H_n[i,j]$ , yüksek geçiren süzgeç işlemcisinin matris gösteriminin  $(i,j)$  elemanını belirtir.

$$H_n[i,j] = \begin{cases} (-1)^{j+1} c_{j-2i+1} & , \quad 0 \leq 2i-j \leq N-1 \\ (-1)^{j+1} c_{2i-j+n} & , \quad 2i-j < 0 \\ 0 & , \quad 2i-j \geq N \end{cases}$$

Her iki gösterimde de  $i=1,2, \dots, n/2$ ,  $j=1,2, \dots, n$  ve  $H_n$  ve  $L_n$   $n/2 \times n$  matrislerdir. Örneğin  $n=8$  ve  $N=6$  için;

$$L_8 = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ c_4 & c_5 & 0 & 0 & c_0 & c_1 & c_2 & c_3 \\ c_2 & c_3 & c_4 & c_5 & 0 & 0 & c_0 & c_1 \end{bmatrix} \quad H_8 = \begin{bmatrix} c_1 & -c_0 & 0 & 0 & c_5 & -c_4 & c_3 & -c_2 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & c_5 & -c_4 \\ c_5 & -c_4 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 \\ 0 & 0 & c_5 & -c_4 & -c_3 & -c_2 & c_1 & -c_0 \end{bmatrix}$$

Ortogonalite özelliğini matrisel olarak (4.51) şeklinde ifade edebiliriz:

$$H_n L_n^T = L_n H_n^T = 0_{n/2} \quad (4.51)$$

Matris yapısını incelediğimizde  $H_n$  ve  $L_n$  matrislerinin (4.52) özelliğini de sağladığını görebiliriz:

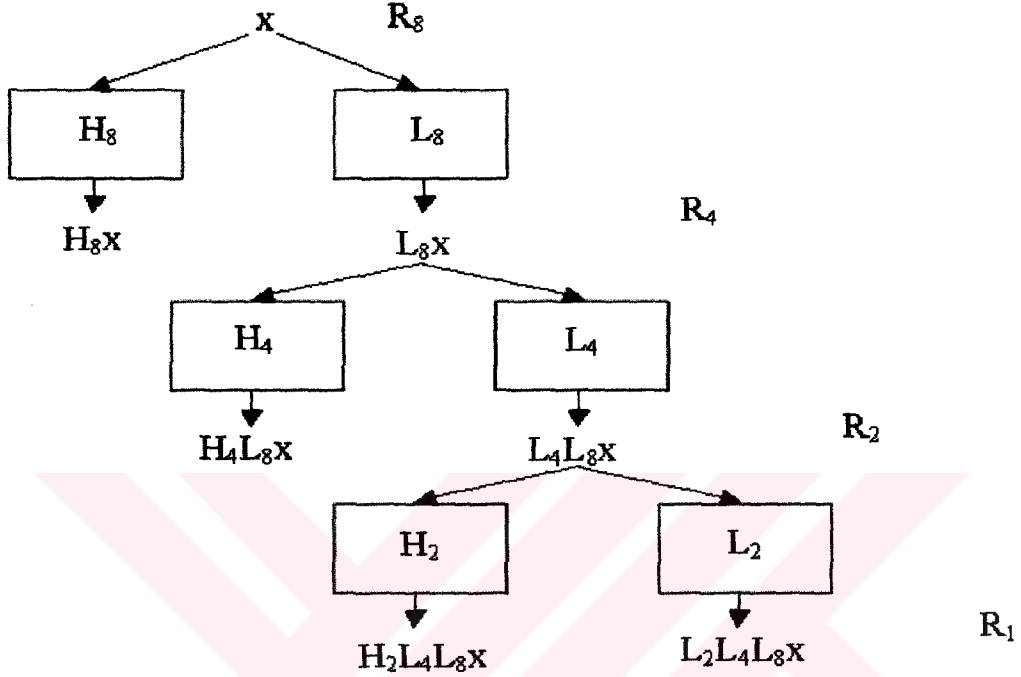
$$L_n L_n^* = I_{n/2} \quad \text{ve} \quad H_n H_n^* = I_{n/2} \quad (4.52)$$

Bu eşitlik aynı zamanda aşağıdaki (4.53) ifadesine de eşittir:

$$\sum_{k=0}^{N-1} c_k c_{k+2m} = \delta_{0m}, \quad m=0, \pm 1, \pm 2, \dots \quad \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \quad (4.53)$$

#### 4.6.4 Dalgacık Dönüşümü

Dalgacık dönüşümü yüksek ve alçak geçiren süzgeçlerden oluşan bir ağaç yapısı şeklinde düşünülebilir. Tekrar edilecek olursa alçak geçiren süzgeç  $\{L_n\}$  işaretteki veya veri dizisindeki enformasyon miktarını azaltırken yüksek geçiren süzgeç ise kaybolan bu enformasyonu saklar. (Daubechies, Antonini, Barlaud, 1992)



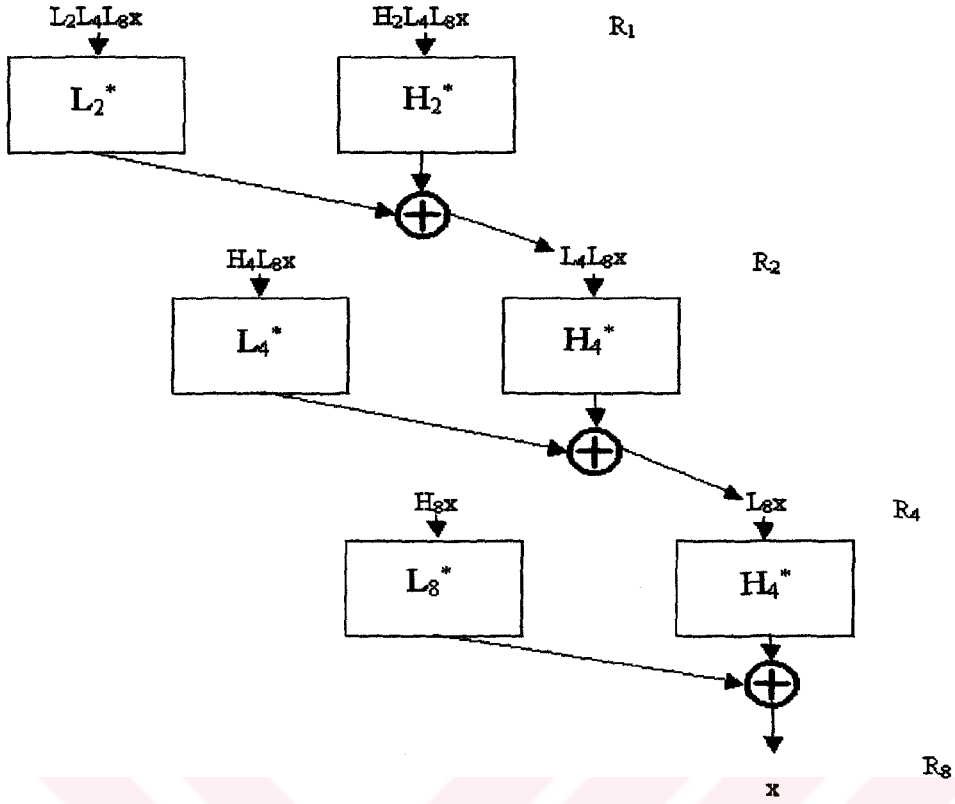
Şekil 4.6 Alçak ve yüksek geçiren süzgeçlerle ağaç yapısındaki dalgacık dönüşümü

Dalgacık dönüşümü yüksek ve alçak geçiren süzgeçlerin oluşturduğu bir ağaç yapısında ayrıştırma işlemidir (Şekil 4.6). Şekilde  $R^8$ 'de tanımlı  $x$  işareti veya dizisi üzerinde dönüşüm işlemi gerçekleştirilmektedir. Dönüşüm sonucundaki elde ise alçak geçiren terim  $L_2L_4L_8x \in R^1$  ve üç yüksek geçiren terim olan  $H_8x \in R_4$ ,  $H_4L_8 \in R_2$  ve  $H_2L_4L_8 \in R_1$ 'dir.

Ters dönüşüm ise önceki uygulanan adımların tersten uygulanmasıdır (Şekil 4.7).

$$L_2^*L_2L_4L_8x + H_2^*H_2L_4L_8x = (L_2^*L_2 + H_2^*H_2)L_4L_8x = L_4L_8x$$

Burada  $(L_2^*L_2 + H_2^*H_2) = I_2$  ifadesi  $R^2$ 'deki özdeşlik ifadesidir.



Şekil 4.7 Ters dalgacık dönüşümünün ağaç yapısındaki gösterimi

#### 4.6.5 Ölçekleme ve Dalgacık Fonksiyonları

Verilen bir katsayı kümesi ( $c_0, c_1, c_2, c_3, \dots, c_{N-1}$ ) için genel bir ölçekleme fonksiyonu,  $\phi(t)$  tanımlanırsa;

$$\phi(t) = s(c_0\phi(2t) + c_1\phi(2t-1) + \dots + c_{N-1}\phi(2t-(N-1))) \quad (4.54)$$

Bu eşitlik genişletme (dilation) eşitliği olarak adlandırılır (Welstead, 1999). “s” sabiti sıfırdan farklıdır ve (4.55) ve (4.56) özelliklerini sağlar;

1.) Normalizasyon:

$$\|\phi\|^2 = \int_{-\infty}^{\infty} \phi^2(t) dt = 1 \quad (4.55)$$

2.) Ortogonalite:

$$\int_{-\infty}^{\infty} \phi(2t-j) \cdot \phi(2t-m) dt = \delta_{j,m} \quad (4.56)$$

Genişletme eşitliği entegre edilirse;

$$\int \phi(t) dt = s \sum_{j=0}^{N-1} c_j \int \phi(2t-j) dt = \left(\frac{1}{2}\right) s \sum_{j=0}^{N-1} c_j \int \phi(u) du \quad (4.57)$$

Bu eşitlikten  $\phi(t)$  fonksiyonunun varlığı için gerekli şart şu şekilde bulunur:

$$\frac{2}{s} = \sum_{j=0}^{N-1} c_j \quad (4.58)$$

Normalizasyon (4.55) ve ortogonallik (4.56) özelliklerini de hesaba katılarak genişletme eşitliğine uygulanırsa şu sonuca ulaşılır.

$$s^2 \left( \sum_{j=0}^{N-1} c_j^2 \right) \cdot \left( \frac{1}{2} \right) = 1$$

(4.53) Eşitliğinin de hesaba katılmasıyla  $m=0$  için  $s = \sqrt{2}$  bulunur.

Yüksek geçiren süzgeç olarak daha genel dalgacıkları tanımlayacak olursak (örneğin  $N=4$ ):

$$\psi(t) = \left( \frac{1}{\sqrt{2}} \right) (c_3 \phi(2t) - c_2 \phi(2t-1) + c_1 \phi(2t-2) - c_0 \phi(2t-3)) \quad (4.59)$$

Eğer katsayılar bilinirse  $\phi(t)$  ve  $\psi(t)$  (4.52) ve (4.57) de olduğu gibi tekrarlamalı olarak elde edilebilir.

Örneğin  $N=2$  için  $c_j$  katsayılarının (4.51) ve (4.56) koşullarını sağladığı durum için;

$$c_0^2 + c_1^2 = 1 \quad , \quad \sqrt{2} = c_0 + c_1 \Rightarrow c_0 = c_1 = \frac{1}{\sqrt{2}} \text{ için Haar katsayılarını elde edilir. Haar fonksiyonu}$$

$N=2$  için Daubechies fonksiyonuna denktir.

$N>2$  için  $c_j$  katsayılarını belirlemek için ek koşullara ihtiyaç vardır.

#### 4.6.6 Daubechies Dalgacığı Katsayıları

Daubechies  $c_j$  katsayılarının belirlenmesi için aşağıdaki koşulları öne sürmüştür.  $W=(c_3, -c_2, c_1, -c_0)$  için (1,1,1,1) ve (1,2,3,4) vektörleri  $W$  boyunca sıfır bileşene sahiptir. Bu (4.60)'de gösterilmiştir.

$$c_3 - c_2 + c_1 - c_0 = 0 \quad \text{ve} \quad c_3 - 2c_2 + 3c_1 - 4c_0 = 0 \quad (4.60)$$

Bu koşullar  $W$ 'nin kaybolan sıfırncı ve birinci mertebeden momentlere sahip olduğunu gösterir. Bunun manası sabit ve lineer işaret enformasyonunun büyük oranda sıkıştırılabileceğidir. Göreceli olarak yumuşak geçişli görüntülerde sıkıştırma olarak büyük avantaj sağlar. Haar dalgacıklarında görülen bloklu yapı görünümü ve verilen sıkıştırma oranı

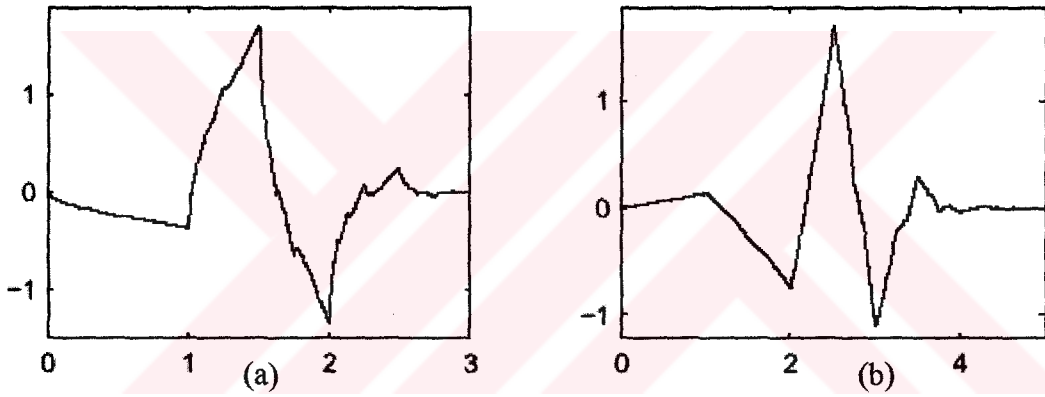
için hata da azalır. (4.60)'den bulunan koşulların normalizasyon ve ortogonalite koşullarıyla birleştirilmesiyle bulunan Daubechies katsayıları:

$$c_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \quad c_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \quad c_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \quad c_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

bu katsayılar D4 Daubechies dalgacıkları katsayılarıdır. Daha fazla kaybolan momentlere sahip sistemler daha fazla katsayı içeren Daubechies dalgacık kümesini oluşturur. Örneğin D6 için (4.61), kaybolan sıfıncı, birinci ve ikinci mertebeden momentlere sahiptir. (Daubechies, Antonini, Barlaud, 1992)

$$1^2c_5 - 2^2c_4 + 3^2c_3 - 4^2c_2 + 5^2c_1 - 6^2c_0 \quad (4.61)$$

Dalgacık fonksiyonlarının çiziminde belirli bir veri uzunluğunda birim vektöre ters dalgacık dönüşümü uygulanır.



Şekil 4.8 Daubechies D4 (a) ve D6 (b) dalgacık fonksiyonları (Misitis, Oppenheim, Poggi, 2001)

Şekil 4.8 (a) da görülen D4 dalgacık fonksiyonu bazı dalgacık fonksiyonu bazı ilginç özelliklere sahiptir;  $k$  ve  $n$  tamsayı olmak üzere  $k/2^n$  noktalarında devamlı bir fonksiyon olmasına karşı türevi almamaz.

Daubechies dalgacıklarının yumuşak geçişliliği (smoothness) kaybolan momentlerin sayısına bağlı olarak artar. Bu geniş yumuşak geçişli alanlara sahip görüntülerde oldukça iyi sonuçlar elde edilebileceği manasına gelir. Örneğin görüntü dilimleri yakın değerlerde veya lineer tonda ise sıkıştırılmada dalgacık fonksiyonlarının kullanılması yararlı olacaktır.



#### 4.7 Diğer Dalgacık Sistemleri

Haar ve Daubechies dalgacıkları işaret ve görüntü işleme uygulamalarında kullanılan dalgacık sistemleri dünyasının küçük bir bölümünü oluşturmaktadır. Haar ve Daubechies dalgacık sistemleri sıkı destek ve ortogonallik özelliklerine sahiptir. Ölçekleme ve dalgacık fonksiyonları (4.62) özelliklerini sağlar. (Welstead, 1999)

$$\left. \begin{aligned} \langle \phi_{j,k}, \phi_{j,l} \rangle &= 0 \\ \langle \psi_{j,k}, \psi_{j,l} \rangle &= 0 \end{aligned} \right\} k \neq l \quad (4.62)$$

buna ek olarak bütün  $k$  ve  $l$  değerleri için (4.63) özelliğini de sağlar.

$$\langle \phi_{j,k}, \psi_{j,l} \rangle = 0 \quad (4.63)$$

Bir dalgacık sisteminde istenen başka özelliklerin de olması beklenebilir. Bu özelliklerden biri simetri özelliğidir. Ölçekleme dalgacık fonksiyonları merkez noktaları referans alınarak simetrik tasarlanabilir. Bu özelliği Haar dalgacıkları sağlarken Daubechies dalgacıkları sağlamaz.

Eğer sıkı desteğe sahip yumuşak geçişli, simetrik dalgacıklar istenirse ortogonallik özelliğinden taviz verilmek zorunda kalınabilir. Uygulamalarda kullanılan yeni dalgacık sistemlerinin bir çoğu ortogonallik özelliğini taşımazlar. Fakat ortogonallığın zayıflatılmış bir biçimi olan yarı-ortogonallığı (semiorthogonality) özelliğini sağlarlar. Bu dalgacık sistemleri (4.63) özelliğini sağlar fakat (4.62) özelliğini taşımazlar.

Ortogonallığın diğer zayıflatılmış biçimi biortogonalliktir. Biortogonallik dualite (duality) fikriyle alakalı bir kavramdır. Ortogonal olmayan fonksiyonların kümesi  $\{u_1, u_2, \dots, u_n\}$  olsun. “ $f(x)$ ” fonksiyonunun bu temel fonksiyonların lineer kombinasyonu şeklinde ifade edilebilmesi mümkündür. (Misitis, Oppenheim, Poggi, 2001)

$$f(x) = \sum_{j=1}^n a_j u_j(x)$$

Ortogonallığın eksikliği katsayıların  $\{a_j\}$  belirlenmesinde zorluklara neden olmaktadır. Bununla beraber başka temel fonksiyonlar bu özelliği sağlayabilir  $\{\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_n\}$ .

$$a_j = \langle f, \tilde{u}_j \rangle$$

Ayrıca bu temel fonksiyonlar için ortogonallikten faydalanarak şu özellik yazılabilir;

$$\langle u_k, \tilde{u}_j \rangle = 0 \quad j \neq k$$

Bu temel fonksiyonlar  $\{u_1, u_2, \dots, u_n\}$  kümesinin dual temeli olarak adlandırılır. biortogonal dalgacık sistemi dört temel fonksiyondan oluşur. Ölçekleme temel fonksiyonu  $\{\phi_{j,k}\}$  ve dual temeli  $\{\tilde{\phi}_{j,k}\}$ ; dalgacık temel fonksiyonu  $\{\psi_{j,k}\}$  ve dual temeli  $\{\tilde{\psi}_{j,k}\}$ . Biortoganallik koşulları şu şekilde verilebilir;

$$\left. \begin{aligned} \langle \phi_{j,k}, \tilde{\psi}_{j,l} \rangle &= 0 \\ \langle \psi_{j,k}, \tilde{\phi}_{j,l} \rangle &= 0 \end{aligned} \right\} \text{bütün } j,k,l \text{ değerleri için}$$

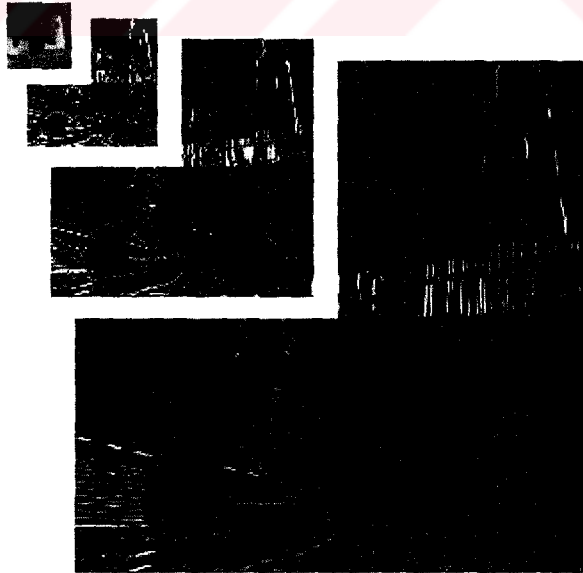
Ek olarak duallik prensibinden;

$$\left. \begin{aligned} \langle \phi_{j,k}, \tilde{\phi}_{j,l} \rangle &= 0 \\ \langle \psi_{j,k}, \tilde{\psi}_{j,l} \rangle &= 0 \end{aligned} \right\} \text{bütün } j \neq l \text{ için}$$

Biortogonal dalgacık sistemleri görüntü sıkıştırma oldukça popüler olarak kullanılmaya başlamıştır. Villsenor, Belzer ve Liao (1995) görüntü sıkıştırma uygulamaları için 4300 biortogonal dalgacık sistemini değerlendirmiştir. (Welstead, 1999)

## 5. AĞAÇ İÇERİKLİ GÖRÜNTÜ KODLAMASI

Genel olarak görüntünün enerji yoğunluğunun büyük bir kısmı alçak frekans bölgelerinde yoğunlaşmıştır. Tipik olarak bir görüntünün spektral bileşenleri frekans yükseldikçe azalır. Görüntünün bu özelliği aynen kendisini ADD'de de gösterir. Herhangi bir görüntünün  $n$  düzeylik ADD ayrıştırmasını yapmış olduğumuzu varsayalım. Bu durumda alçak düzeyler yüksek frekans alt bantlarına denk düşer. Örneğin, birinci düzey en yüksek frekans alt bandını gösterirken aynı zamanda çözünürlüğün en yüksek olduğu düzeyi de gösterir. Buna ait olarak  $n$ . düzey en alçak frekans alt bandını gösterirken çözünürlüğün en kaba seviyesine de denk düşmektedir. Bu nedenle düzeylerden alçak düzeylere ilerlerken (düşük çözünürlükten yüksek çözünürlüğe doğru) , alt bandın alt bantların enerji içeriğinin azalmakta olduğunu görürüz. Aynı yaklaşımla belirli düzeydeki alt bandın piksel genlik değerlerinin düşük olduğu gözlemlendiğinde, daha düşük düzeydeki alt banttaki piksel değerinin daha düşük olmasını bekleriz. Bunlara ek olarak, uzamsal düzlemde alt bantlar arası benzerlikler de bulunmaktadır. Bu benzerlik şekil 5.1'de piramit yapısında daha rahat bir biçimde gözlemlenebilir. Alçak geçiren alt banttaki görüntünün kaba hatları alçak düzeylerde ise görüntünün ayrıntıları yani yüksek frekans bileşenleri-hızlı kenar geçişleri –görülür. (Shapiro, 1993) tarafından tanıtılan gömülü sıfır ağacı dalgacık kodlama algoritması dönüşümün bu özelliğinden faydalanır. Daha sonra bu algoritmayı temel alan “Hiyerarşik Ağaçlarda Küme Bölümleme (HAKB)” (Said, Pearlman, 1996) tarafından tanıtılmıştır. Her iki yöntem de gömülü bir bit dizisi oluşturur.



Şekil 5.1 Piramitte alt bantlar arası ilişki

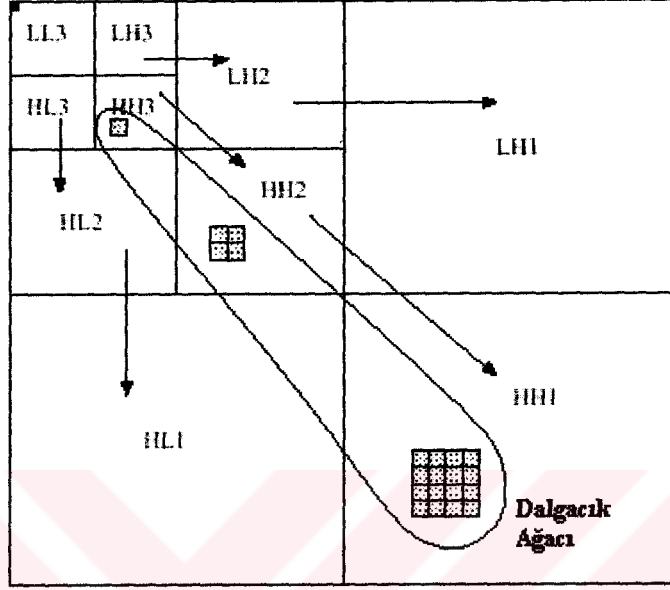
Bu çalışmada daha gelişmiş bir algoritma olan HAKB algoritması kullanılmıştır. Görüntülerin ADD'ü dengesiz bir karakteristik gösterir. Diğer bir deyişle ayrıntı görüntülerin piksel değerleri büyük bir oranda sıfır veya sıfıra yakın değerlerden oluşur. Bu nedenle HAKB algoritmasının bu özelliği nasıl kullandığı incelenecektir. Adımlar şeklinde ilerlenecek olursa;

- 1.) Giriş görüntüsünün ADD'ü elde edilir. Ardından elde edilen katsayılar sonlu sayıda bitle varolan dönüşüm sonuçları görüntüyü çok hassas bir nicemlemeyle temsil edilir. Gösterimde 16 bit işaretli genlik değeri kullanılır (en değerli bit işaret biti geri kalan 15 bit ise genlik değerinin gösterimde kullanılmaktadır). Bu durumda piksel değerleri  $-(2^{15}-1)$  ile  $(2^{15}-1)$  arasında değer alırlar.
- 2.) Eşik genliği  $2^{14}$  olmak üzere katsayıları  $-1,0,1$  değerleriyle nicemlenir. Diğer bir anlatımla dalgacık katsayısı değeri  $-2^{14}$  'ten küçükse  $-1$ 'e, sıfır ise  $0$ 'a,  $2^{14}$ 'ten büyük ise  $1$ 'e nicemleyelim. Sıfır olmayan nicemlenmiş değerler katsayıların ikinci en değerlikli bitinden elde edilen 1 değerli bittir. Elde edilen sonuç görüntü "değerlilik haritası - significance map-"olarak adlandırılır. Bunun nedeni herhangi bir katsayının eldeki eşik değerine göre değerli (significant) olup olmadığının bu şekilde gösterilmesidir.
- 3.) Değerlilik haritası ağaç yapısında kodlanır.
- 4.) Eşik  $2^{13}$  değerine getirilir ve bir önceki kodlama işlemi benzer şekilde uygulanır. Elde edilen ikili kod bir önceki adımdan elde edilen koda eklenir. Bir önceki adımda değerli bulunan piksellerin bir sonraki en değerlikli bitleri bu adımda kodun sonuna eklenir
- 5.) Önceki adımlar azalan eşik değerleriyle devam ettirilir. ( $2^{12}, 2^{11}, 2^{10}, \dots$ )

Eşik seviyesi her azaltıldığında görüntüyü temsil eden bit miktarı ve tekrar oluşturulacak olan görüntünün kalitesi artar. Görüntünün tekrar oluşturulması için yapılan çözme işleminde önce ilk eşığe göre görüntü yeniden oluşturulur, ardından ikinci eşığe göre görüntü tekrar oluşturulur ve bu işlem bu şekilde devam eder. Bu internetten de tecrübe edilebilecek olan görüntünün kaba halden aşama aşama daha net hale geldiği etkiyi yaratır. Bu etki ilerleyen iletim (progressive transmission) olarak adlandırılır.

Kodlama yöntemi daha yakından inceleyecek olursak ADD katsayılarının uzamsal yönelimli ağaçlar (spatial orientation trees) şeklinde organize edildiğini görürüz. Şekil 5.2 bu ağaç yapısının nasıl yapılandırıldığını göstermektedir. Bu resim iki seviyeye ayrıştırılmıştır. LL2 veya alçak geçiren alt bant orijinal resmin en kaba yaklaşıklılığını göstermektedir. Alt banttaki pikseller  $2 \times 2$  pikselden oluşan gruplar şeklinde yapılandırılmıştır. Bunun haricinde

(1,1) koordinatındaki yani en üst soldaki piksel hiçbir ağacın elemanı değildir. Her bir grubun diğer üç pikseli dört çocuğa (offsprings), bir düşük seviyedeki alt banttaki 2x2'lik gruba sahiptir. Bunlar LH2, HL2 ve HH2 alt bantlarındadır. Bu şekilde ağaç yapısının ilk kısmı oluşturulur. Ağacı daha da genişletmek için aynı tarzda bir sonraki alt bantlarla (LH1, HL1 ve HH1 sırasıyla) birleştirilir.



Şekil 5.2 Pikseller ve uzamsal yönelimli ağaçlar arasındaki ilişki

Bir banttaki piksel  $(i,j)$  eğer en düşük bantta değilse bir sonraki alt banttaki çocuk piksellere-  $(2i,2j)$ ,  $(2i+1,2j)$ ,  $(2i,2j+1)$  ve  $(2i+1,2j+1)$ - bir kök olarak rol oynar.

Genel olarak  $n$  düzeyli bir ayrıştırma için ardışıl alt bantları  $LH_n$ ,  $HL_n$ ,  $HH_n$  sırasıyla bağlamak bir kuraldır. Böylece verilen bir piksel, uzay ağaç yapısında çocuk ve torunları (descendants) için sadece bir kök yer alır. Burada ağacın ilk dallanmasıyla daha sonraki ardışıl dallanmalar arasındaki farka dikkat etmek gerekir. İlk adımda alçak geçiren alt banttaki pikseller aynı seviyedeki üç yüksek geçiren alt banda bağlanılır.

### 5.1 HAKB Algoritmasının Temel Geçişleri

Her bir eşikte-ilk hariç olmak üzere-iki geçişten oluşur. Bu geçişler; sınıflama geçişi (sorting pass) ve inceltme geçiştir (refinement pass). İlk eşikte sadece sınıflandırma geçişi uygulanır. Verilen bir  $n$  değeri için genliği  $2^n$ 'in üstündeki katsayılar değerli olarak, diğerleri ise değersiz olarak kabul edilir. Geçişlerde veri değerlendirilirken  $2^n \leq |c_{ij}| \leq 2^{n+1}$  arasındaki katsayılar seçilir, her bir geçişte  $n$  bir azaltılır.

Sınıflama geçişi pikselleri bölümlenmiş alt kümelerle ayırır ve bu alt kümeleri değerli pikselleri araştırarak test eder. Alt kümelerde değerli piksel bulunması durumunda bu alt küme daha küçük alt kümelere daha önce belirlenmiş kurallara göre bölünür. İşlem alt kümelerdeki bütün pikseller test edilinceye kadar devam eder.

İnceltme geçişinde bir önceki geçişte değerli bulunan bütün katsayıların n. en değerli biti iletilir. Elde edilen bitle sınıflama geçişindeki bitlerin sonundan itibaren aynı genel düzenle eklenir.

Bu algoritmanın gerçekleşmesinde şu üç liste kullanılır; Değerli pikseller listesi (DPL)-List of significant pixels-, Değersiz pikseller listesi (DzPL)-List of insignificant pixels-, Değersiz kümeler listesi (DKL)-List of insignificant sets-. Her bir sınıflama geçişinin sonunda DPL listesi kullanılan eşige göre değerli bulunan piksellerin koordinatlarını içerir. DKL ise belirli piksel kümelerinin koordinatlarını içerir. Listenin girişleri A tipinde ya da B tipinde olabilir (literatürde sırasıyla D ve L tipi olarak da geçmektedir). “A” tipindeki bir giriş bütün soyu temsil ederken, “B” tipindeki girişler ise çocuklar hariç daha ötedeki soy üyelerini yani torunlar ve daha büyük torunları temsil eder. ADD görüntüsünün elemanlarını  $c_{i,j}$  şeklinde temsil edersek algoritmayı şu adımlardan oluşur:

- 1.) **Başlangıç:** Bu adımda  $n = \lfloor \log_2(\max_{i,j} |c_{i,j}|) \rfloor$  değeri bulunur. Burada  $\lfloor \cdot \rfloor$  sembolü  $|c_{i,j}|$ 'den küçük en büyük tamsayı değerini göstermektedir. Böylece değerlilik haritasını oluşturulacak pikselleri test etmek için gerekli n değeri bulunmuş olacaktır. Başlangıçta DPL listesi boş bir liste olarak oluşturulur. DKL listesi ise soylara kök olan alçak geçiren banttaki pikselleri içerir. Başlangıç olarak bütün pikseller “A” tipli olarak belirlenir. DzPL ise alçak geçiren banttaki bütün pikselleri içerir.
- 2.) **Sınıflama Geçişi:** DzPL listesinin girişleri belirlenen n değerine göre değerlilik testinden geçirilir. Eğer giriş değerli bulunursa 1, ardından ise iletilen pikselin işaret biti iletilir. Bunu takiben pikselin koordinatları DPL listesinin sonuna eklenir. Eğer değerlikli değil ise bir 0 biti iletilir bu yolla bir değerlilik haritası oluşturulur. Ardından DKL listesinin her bir girişi değerli soyların varlığına ilişkin test edilir. Eğer yoksa bir 0 biti iletilir, ama en azından bir tane değerli soy elemanına sahipse bir 1 iletilir. Takiben bu değerli piksele ait çocukların (4 çocuk) değerli olup olmadığına bakılır. Bu çocuklardan değerli olan varsa bir 1 biti ve çocuk pikselin işaret biti iletilir. Pikselin koordinatı ise DPL'nin sonuna eklenir. Eğer çocuk piksel değerli değilse bir 0 iletilerek koordinatları DzPL

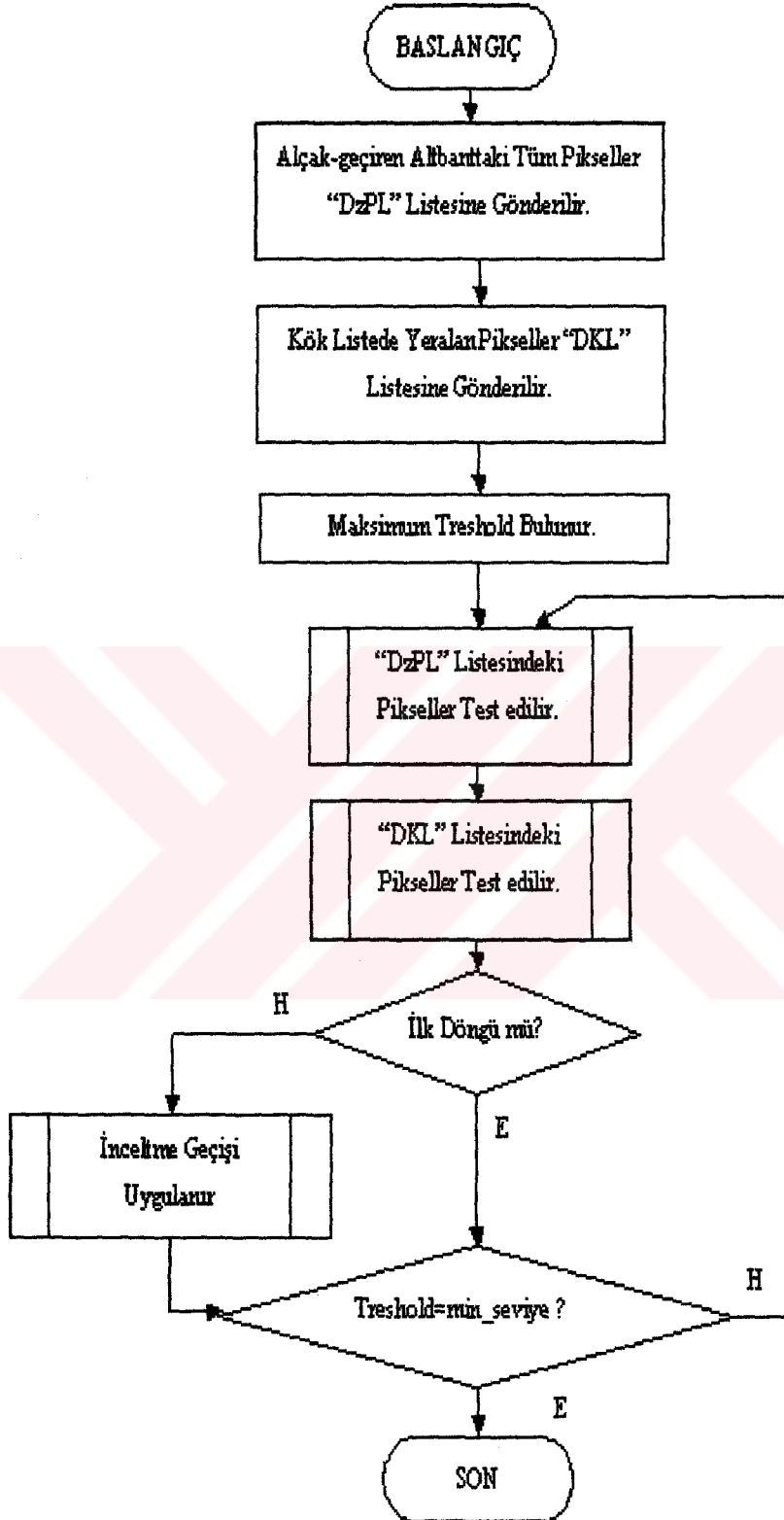
listesinin sonuna eklenir. Böylece bir sonraki alt banttaki çocukların değerlilik haritası iletilmiş olur Ardından bu pikselin torunlara sahip olup olmadığına bakılır, sahipse DKL'nin sonuna "B" tipinde eklenir. DKL listesindeki "B" tipindeki girişler için daha ilerideki torunlar ve büyük torunlar değerli olup olmadığına göre test edilir. En azından bir tane değerli piksel varsa bu giriş listeden çıkarılır ve bu pikselin çocukları listenin sonuna "A" tipinde eklenir.

**3.) İnceltme Geçişi:** DPL listesine bir önceki sınıflandırma geçişinde eklenmiş olan girişlerin dışındaki elemanların genliğinin n. En değerlikli biti iletilir. Özellikle dikkat edilmesi gereken husus ilk sınıflandırma geçişinden sonra DPL boş olduğundan herhangi bir bit iletimi olmaz.

**4.) Nicemleme adımının güncelleştirilmesi:** n değeri bir azaltılır ve işlem ikinci adımdan itibaren tekrar edilir.

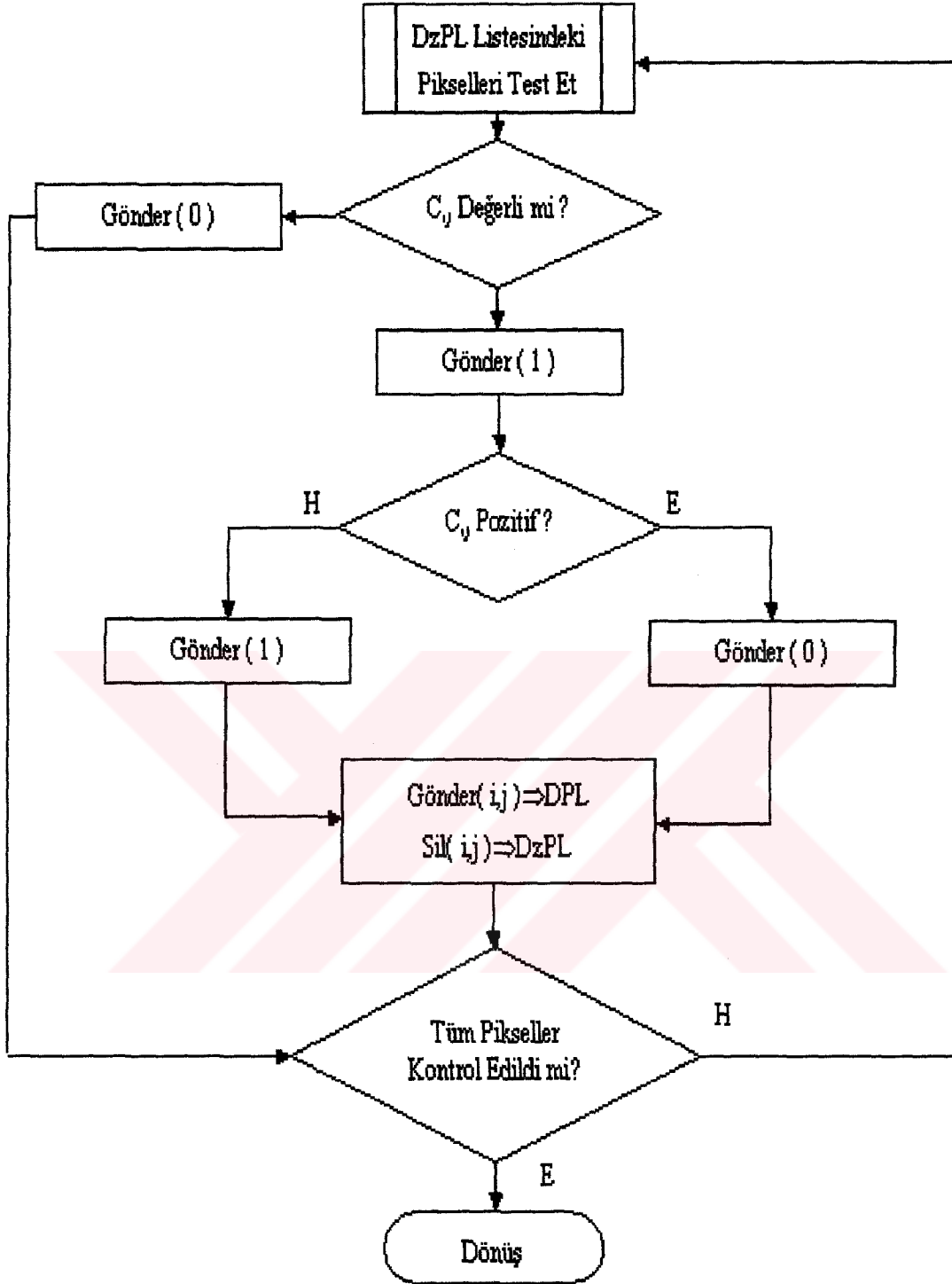
DKL listesinin sonuna eklenmiş olan girişler inceltme geçişine geçmeden değerlendirilir. Kodlama bit tabanında ilerlerken istenilen kritere göre mesela bit oranına veya belirli bir bozunum ölçümüne göre herhangi bir noktada durdurulabilir. Kodlama algoritması belirli bir yol izler ve bitler belirli bir düzende iletilir. çözücü tarafta aynı düzen takip edilerek değerlilik haritası yeniden çizilir. Değerli pikseller uygun koordinatlara (  $[2n+1-2n]/2$  ) değeriyle yerleştirilir. Bu tarzla devam edilerek alt bant bütünleştirilmiş bir yaklaşımla yeniden oluşturulur. Sonuç olarak ters ADD'yle alt batlara bölünmüş görüntü orijinal resme kayıplı olarak dönüştürülebilir.

## 5.2 Algoritma

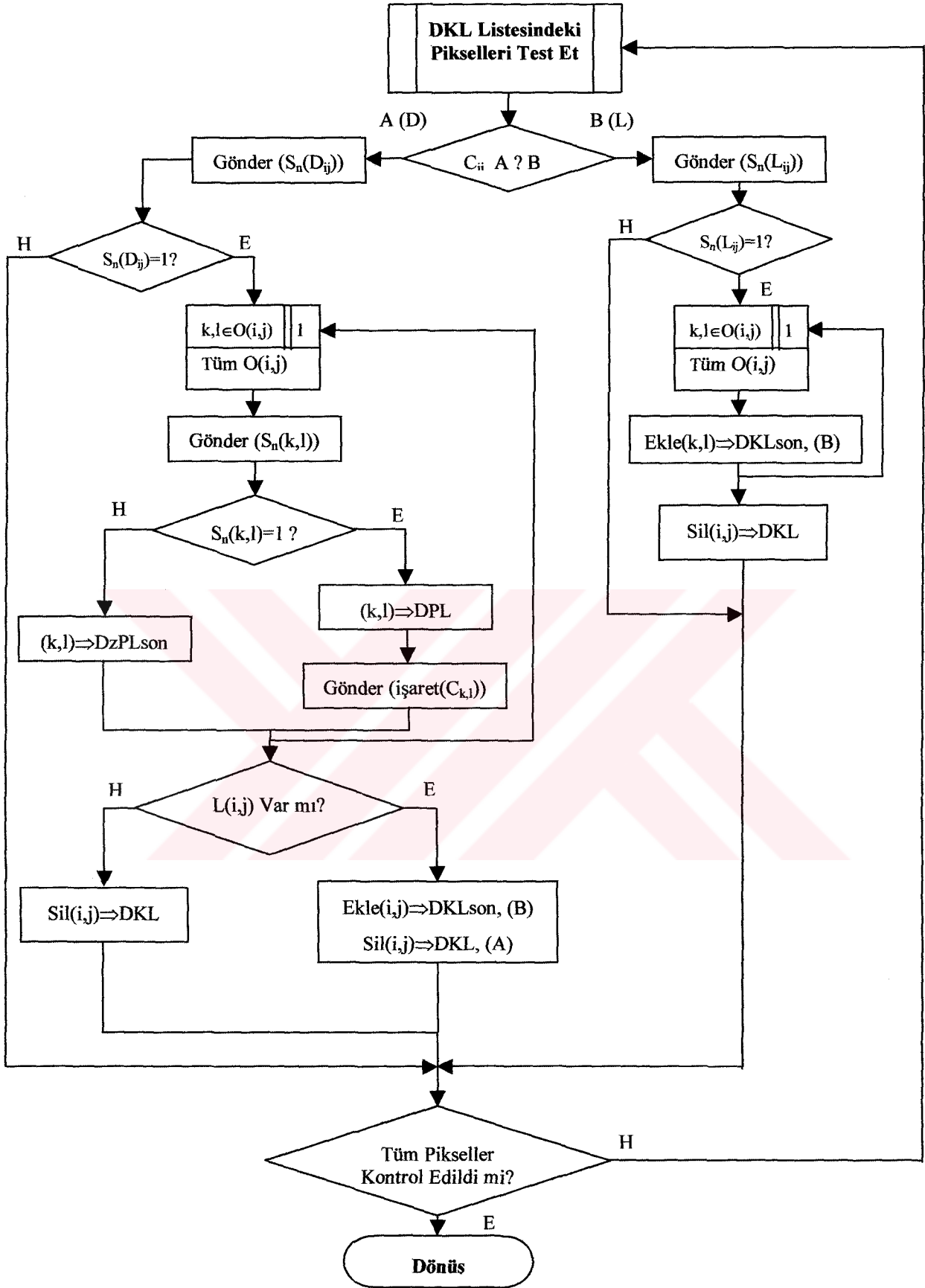


Şekil 5.3 Genel Hiyerarşik Ağaçlarda Küme Bölümleme Akış Diyagramı





Şekil 5.4 DzPL Test Algoritması



Şekil 5.5 DKL Test Algoritması

## 6. SONUÇLAR VE ÖNERİLER

### 6.1 Sonuçlar ve Uygulamalar

Dalgacık seçimi uygulamaya bağlı olarak belirlenir. Seçilen dalgacığın şekline bağlı olarak uygun görüntü bileşenleri elde edilir. Seçilen dalgacık süzgeç katsayıları nedeniyle geri dönüşüm işleminde kayıp söz konusudur. Örneğin uygulama sonuçlarında da görülebileceği gibi Haar dalgacığı ile orijinal veri (görüntü) aynen elde edilebildiği halde Daubechies D4 ve Biorthogonal 2.2 dalgacıklarında veri birebir elde edilemez. Bu durum 6.1, 6.3, 6,5 çizelgelerinde eşik değerleri sıfır olarak seçildiğinde ölçülen sinyal gürültü oranlarında da görülebilir. Haar dalgacığı seçilerek yapılan geri dönüşüm işleminde piksel değerlerinde hata olmadığından sonsuz sinyal gürültü oranı elde edilir; diğerlerinde ise geri dönüşüm hatalarından dolayı bu oran daha düşüktür.

Dalgacık dönüşümü ile elde edilen katsayılara uygulanan hiyerarşik ağaçlarda küme bölümlene algoritması ile sıkıştırma oranı artmaktadır. Aynı şekilde dönüşüm katsayılarına uygulanan eşik değeri HAKB yönteminin sonuçlarını ve performansını oldukça etkilemektedir. Eşik değerinin artırılması görüntü içindeki sıfır değerli katsayıların sayısını arttırmaktadır. Bu durum 6.1, 6.3, 6,5 çizelgelerinde eşik değeri ve görüntü içerisindeki eşikle beraber artan sıfır değerli katsayıların oranları rahatça gözlemlenebilir. Fakat eşik değerinin artırılmasıyla geri dönüşümde katsayı kayıplarından dolayı görüntü kalitesi düşmektedir. Bu durumda öznel olarak görüntü kalitesi ve HAKB algoritması için gerekli olan sıfır ağaçlarının oranını arttırmak için gerekli sıfır değerli katsayı yüzdesi arasında seçim yapılmalıdır. Haar ve Daubechies D4 dalgacığı için eşik değeri 20, biortogonal dalgacık için 40 olarak seçilerek görüntü sıkıştırma işlemi gerçekleştirilmiştir.

Seçilen görüntülere HAKB algoritması, artan minimum seviye değerleri uygulanarak görüntü sıkıştırma işlemi gerçekleştirilmiştir. Minimum seviye değeri arttıkça katsayıların orijinal görüntüdeki değerlerine yaklaşırabilmek için yapılan nicemleme adımlarının sayısı da azalmaktadır. Dolayısıyla yüksek minimum seviye seçimlerinde görüntü kalitesi giderek düşmektedir.

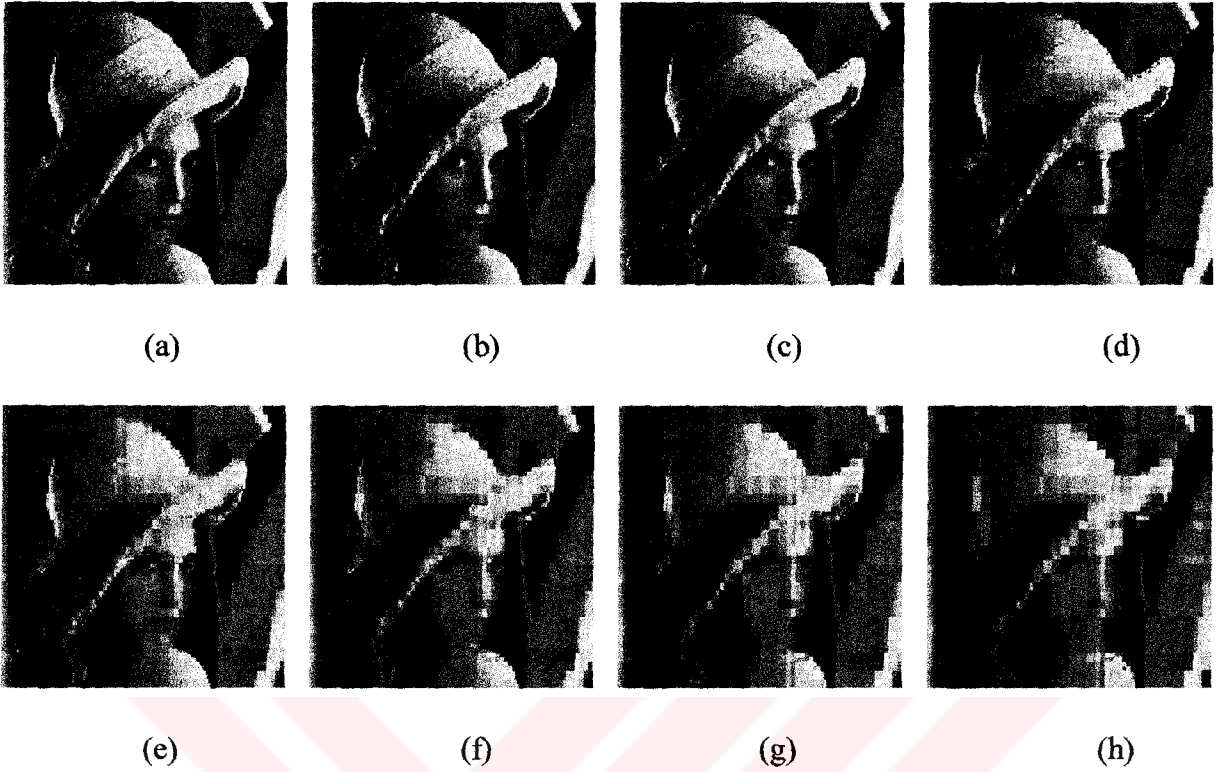
Uygulama yazılımı Matlab ortamında gerçekleştirilmiştir. *Wd2.m* dosyası ile dalgacık seçimi menüsü ile istenilen dalgacık seçilir. Dönüşüm işleminin ardından eşik değeri kullanıcıya sorulur. Eşik değerine göre katsayılardaki sıfır değerli elemanların oranı elde edilir. *Enspiht.m* dosyası *soy\_kontrol.m* ve *ileri\_soy\_kontrol.m* alt fonksiyonlarının yardımıyla HAKB algoritmasının kodlama işlemini gerçekleştirir. Program girdi olarak kullanıcıdan minimum

seviye deęerini ister. Bu programın sonunda grnt bit dizisi haline dnştrlr, piksel bařına dřen bit sayısı hesaplanır. *Despiht.m* programı ise ters HAKB algoritması uygulayarak dnřm katsayıları tekrar elde edilir Elde edilen dnřm katsayıları *Wr2.m* programı ile ters dnřm uygulanarak grnt dosyası bięiminde yapılandırılır.

Uygulamada grnt iřleme alanında sık olarak kullanılan 24 bitlik 128x128 Lenna ve 128x128 Lighthouse grntleri kullanılmıřtır.



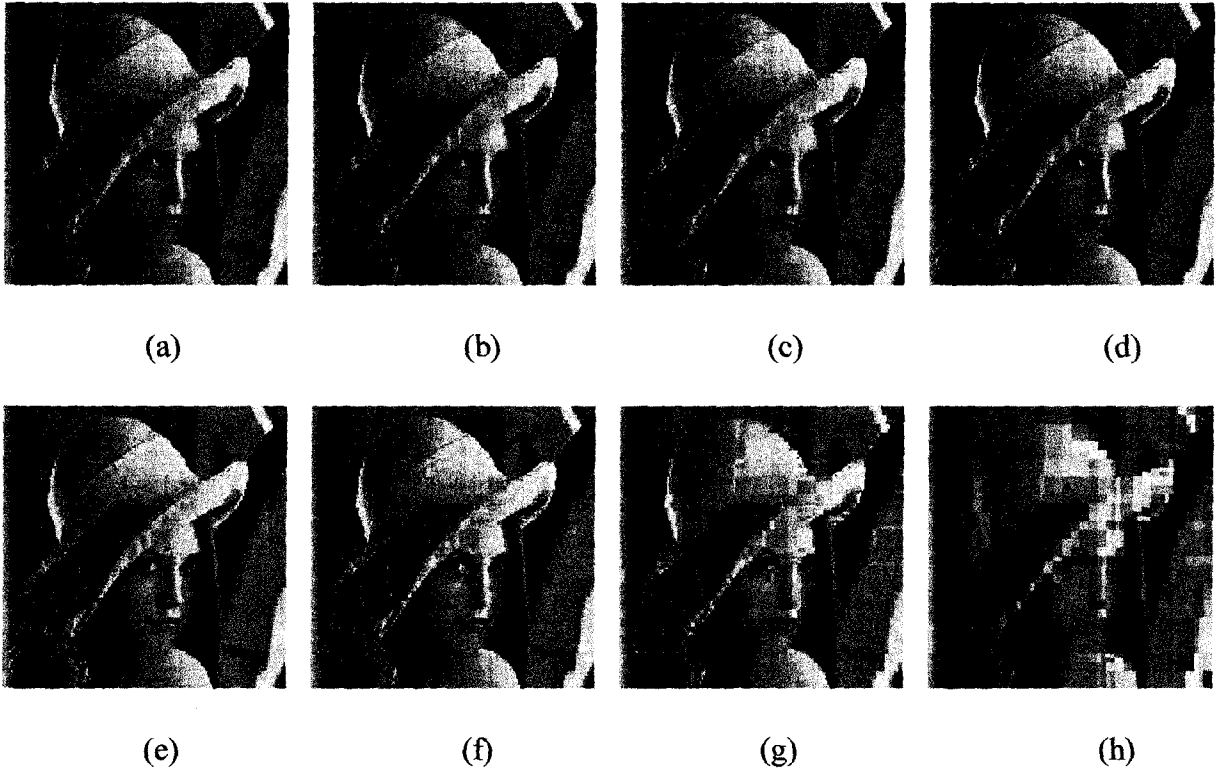
### 6.1.1 Haar Dalgacıklarıyla Geri Dönüşüm



Şekil 6.1 Haar dalgacığıyla artan eşik değerinde (a-h) geri dönüşüm görüntüleri (bkz. Çizelge 6.1)

Çizelge 6.1 Haar dalgacığı geri dönüşüm değerleri

Haar	EŞİK DEĞERİ	SIFIR ORANI	SNR (dB)	PSNR (dB)
(a)	0	0,0750	Sonsuz	Sonsuz
(b)	10	0,6675	31,72	37,48
(c)	20	0,8036	26,31	32,08
(d)	40	0,9097	21,26	27,05
(e)	60	0,9483	18,89	24,71
(f)	80	0,9665	17,43	23,26
(g)	100	0,9758	16,47	22,33
(h)	120	0,9819	15,70	21,57

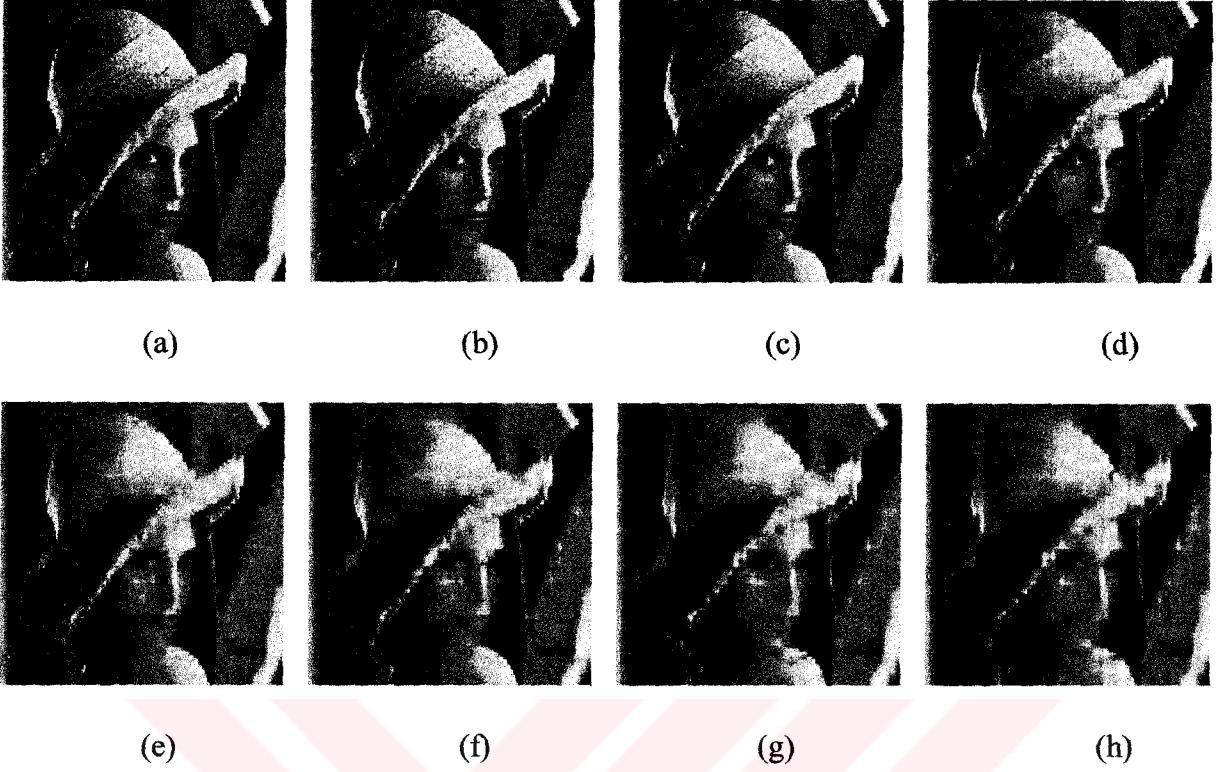


Şekil 6.2 Haar dalgacığı eşik değeri=20, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.2)

Çizelge 6.2 Haar dalgacığı tabanlı HAKB geri dönüşüm sonuçları

Haar	MİN. SEVİYE	(bpp)	SIKIŞTIRMA	SNR(dB)	PSNR (dB)
(a)	0	3,3073	2,4149	22,38	28,39
(b)	1	2,8307	2,8261	22,05	28,04
(c)	2	2,3541	3,3982	22,55	28,56
(d)	3	1,8776	4,2607	22,75	28,75
(e)	4	1,4010	5,7100	22,97	28,95
(f)	5	0,8082	9,8982	20,98	26,93
(g)	6	0,3278	24,4036	17,75	23,70
(h)	7	0,1095	73,0206	15,04	21,02

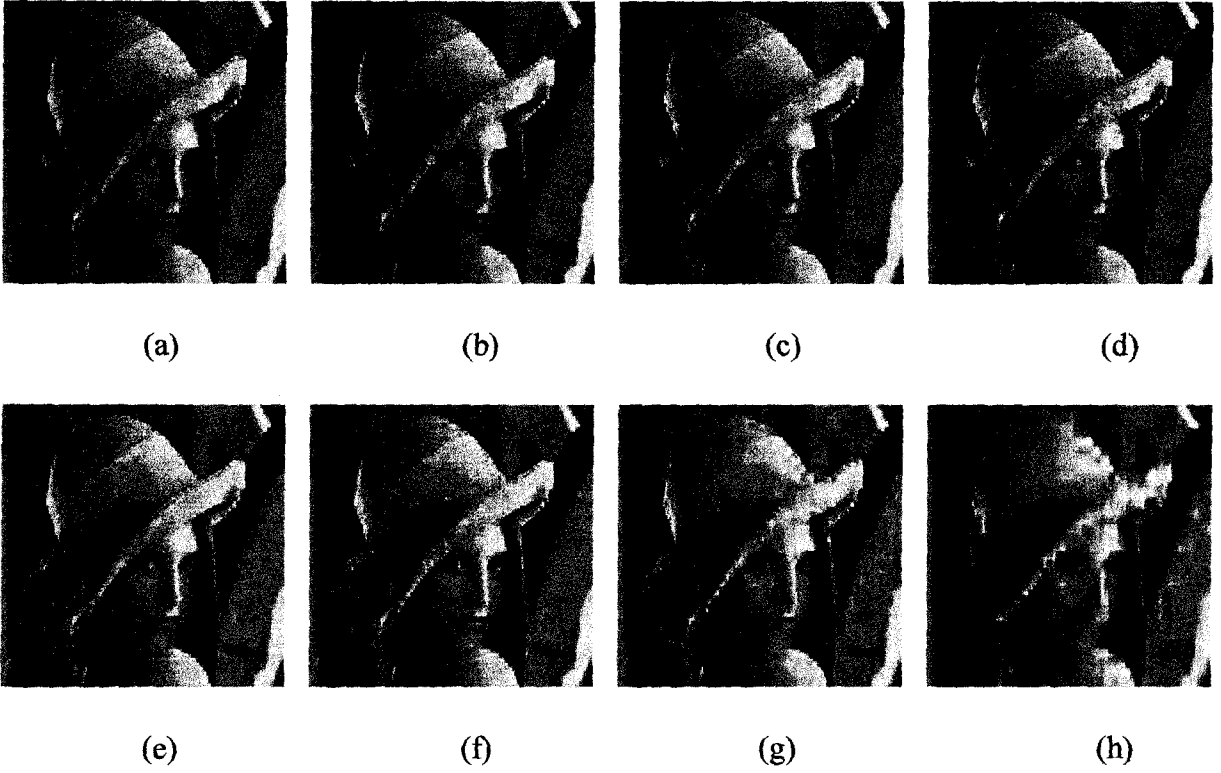
### 6.1.2 Daubechies (D4) Dalgacıklarıyla Geri Dönüşüm



Şekil 6.3 Daubechies D4 dalgacı ile artan eşik değerinde (a-h) geri dönüşüm görüntüleri.  
(bkz.Çizelge 6.3)

Çizelge 6.3 Daubechies D4 dalgacı geri dönüşüm değerleri.

Daubechies	EŞİK DEĞERİ	SIFIR ORANI	SNR (dB)	PSNR (dB)
(a)	0	$9,1553 \cdot 10^{-4}$	27,94	33,71
(b)	10	0,6666	26,63	32,40
(c)	20	0,8010	24,55	30,33
(d)	40	0,9026	21,32	27,11
(e)	60	0,9383	19,50	25,31
(f)	80	0,9556	18,32	24,15
(g)	100	0,9644	17,52	23,37
(h)	120	0,9716	16,74	22,60



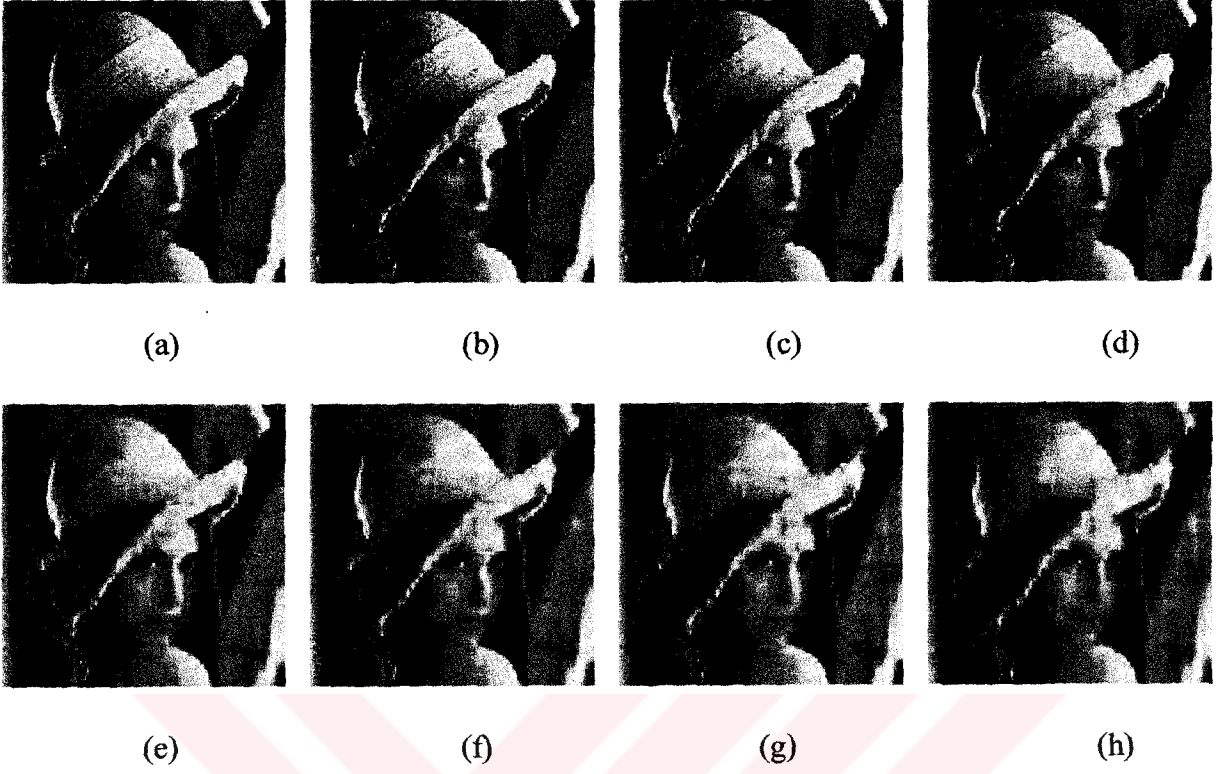
Şekil 6.4 Daubechies D4 dalgacığı eşik değeri=20, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.4)

Çizelge 6.4 Daubechies D4 dalgacığı tabanlı HAKB geri dönüşüm sonuçları

Daubechies	MİN. SEVİYE	(bpp)	SIKIŞTIRMA	SNR(dB)	PSNR (dB)
(a)	0	3,6081	2,2172	19,30	25,61
(b)	1	3,0997	2,5809	19,32	25,64
(c)	2	2,5912	3,0872	19,37	25,68
(d)	3	2,0828	3,8408	19,45	25,76
(e)	4	1,5744	5,0811	19,55	25,83
(f)	5	0,9506	8,4150	18,70	24,98
(g)	6	0,4490	17,8160	16,97	23,25
(h)	7	0,2033	39,3373	14,99	21,31



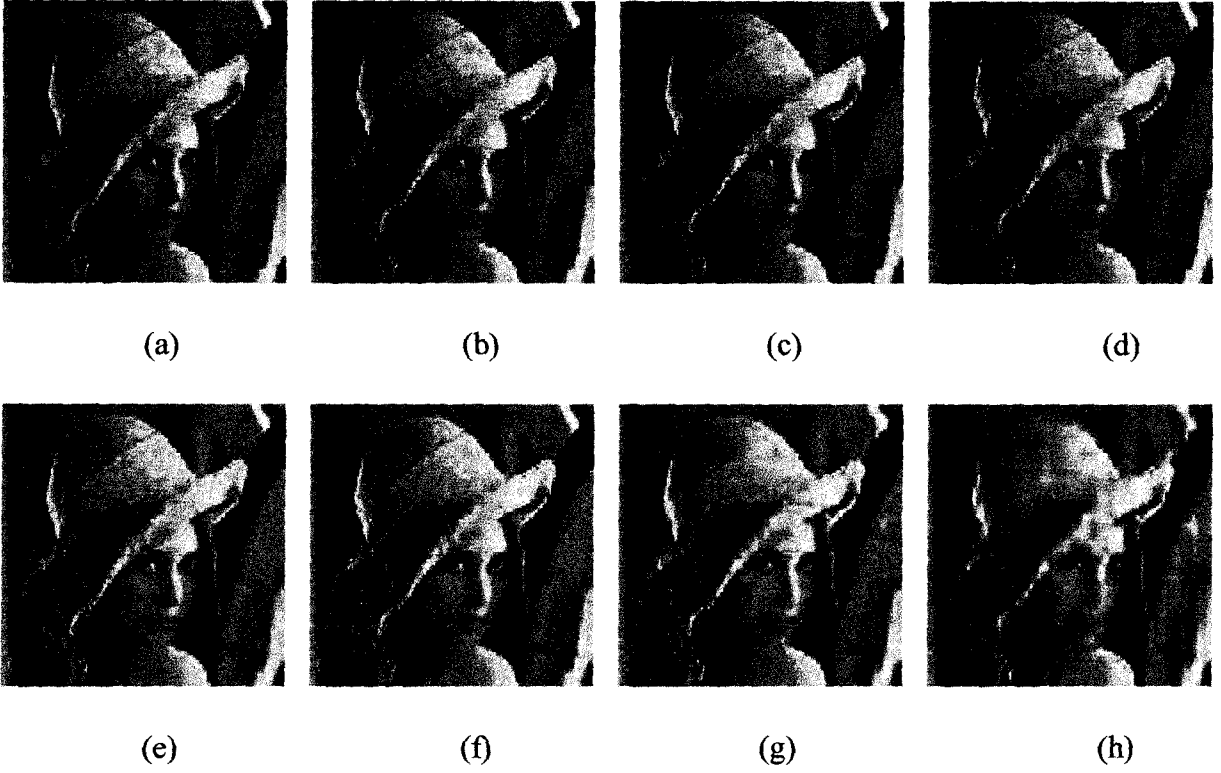
### 6.1.3 Biortogonal Dalgacıklarla (Bior 2.2) Geri Dönüşüm



Şekil 6.5 Biortogonal 2.2 dalgacığı ile artan eşik değerinde (a-h) geri dönüşüm görüntüleri.  
(bkz.Çizelge 6.5)

Çizelge 6.5 Biortogonal 2.2 dalgacığı geri dönüşüm değerleri.

Biortogonal	EŞİK DEĞERİ	SIFIR ORANI	SNR (dB)	PSNR (dB)
(a)	0	0,0057	24,95	30,73
(b)	10	0,6263	24,37	30,14
(c)	20	0,7528	23,31	29,10
(d)	40	0,8531	21,37	27,16
(e)	60	0,8975	19,97	25,77
(f)	80	0,9191	19,10	24,91
(g)	100	0,9349	18,24	24,07
(h)	120	0,9448	17,54	23,38

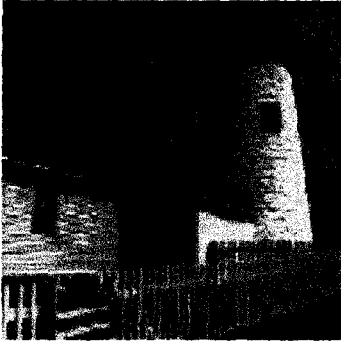


Şekil 6.6 Biortogonal 2.2 dalgacıđı eşik değeri=40, HAKB yöntemi minimum seviyeler (a-h) artan biçimde geri dönüşüm görüntüleri (bkz. Çizelge 6.6)

Çizelge 6.6 Biortogonal 2.2 dalgacıđı tabanlı HAKB geri dönüşüm sonuçları

Biortogonal	MİN. SEVİYE	(bpp)	SIKIŞTIRMA	SNR(dB)	PSNR (dB)
(a)	0	3,2559	2,4571	18,22	24,52
(b)	1	2,8653	2,7920	18,24	24,53
(c)	2	2,4747	3,2326	18,26	24,56
(d)	3	2,0842	3,8384	18,32	24,60
(e)	4	1,6936	4,7235	18,41	24,68
(f)	5	1,3030	6,1392	18,53	24,79
(g)	6	0,8298	9,6398	17,60	23,84
(h)	7	0,4686	17,0711	15,94	22,18

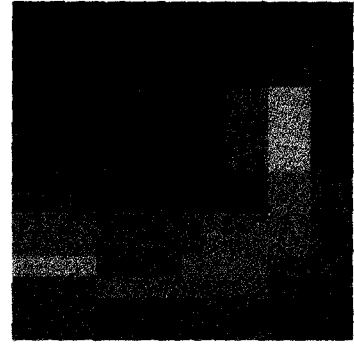
### 6.1.4 İlerleyen İletim



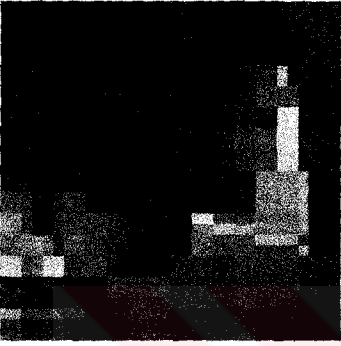
(a) Orijinal Resim



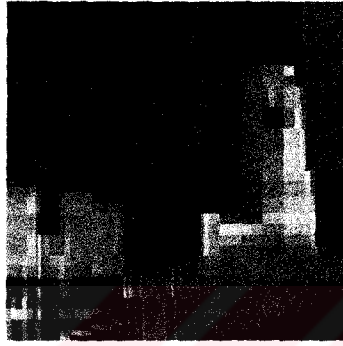
(b) Min. Seviye=10



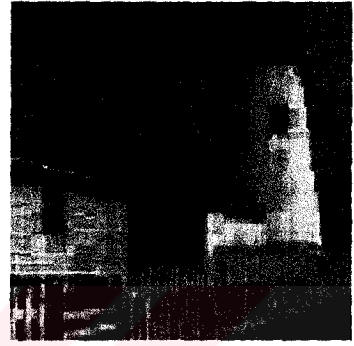
(c) Min. Seviye=9



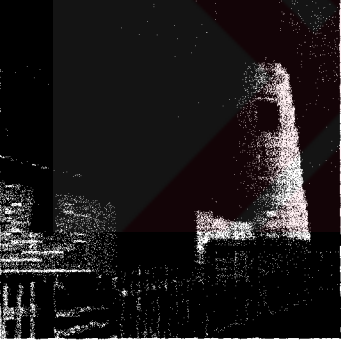
(d) Min. Seviye=8



(e) Min. Seviye=7



(f) Min. Seviye=6



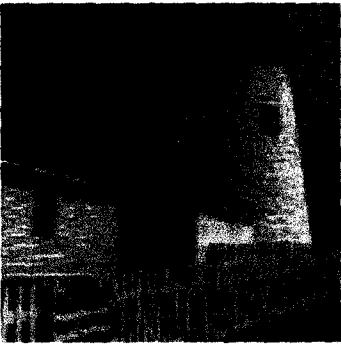
(g) Min. Seviye=5



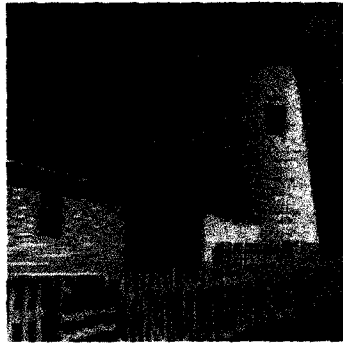
(h) Min. Seviye=4



(i) Min. Seviye=3



(j) Min. Seviye=2



(k) Min. Seviye=1



(l) Min. Seviye=0

Şekil 6.7 İlerleyen İletim (a-l) HAKB kodlaması azalan minimum seviyeleri ile geri dönüşüm

Çizelge 6.7 İlerleyen iletim HAKB kodlama ve geri dönüşüm sonuçları

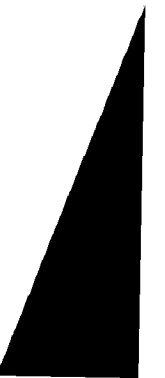
Haar Eşik=10	MİN. SEVİYE	(bpp)	SIKIŞTIRMA	SNR(dB)	PSNR (dB)
(b)	10	0,0027	1702,2	22,89	29,66
(c)	9	0,0084	636,2718	22,97	29,73
(d)	8	0,0288	212,4344	23,10	29,86
(e)	7	0,0958	68,7320	23,30	30,05
(f)	6	0,2767	25,4756	22,61	29,35
(g)	5	0,6058	12,6781	20,72	27,44
(h)	4	1,1819	6,2813	18,18	24,90
(i)	3	1,8819	4,0100	15,43	22,16
(j)	2	2,4538	3,0923	13,23	19,97
(k)	1	3,0257	2,5165	11,07	17,87
(l)	0	3,5976	2,1214	9,84	16,62

## 6.2 Öneriler

Dalgacık dönüşümleri ile dalgacığa bağlı olarak kayıplı ve kayıpsız ters dönüşümler gerçekleştirilebilir. Ayrıca seçilen dalgacıklar veri (resim) üzerinde farklı etkilere sahiptir. Örneğin Daubechies dalgacıklarında yumuşak geçişli alanlara sahip görüntülerde oldukça iyi sonuçlar elde edilebilir, diğer bir deyişle görüntü dilimleri yakın değerlerde veya lineer tonda ise sıkıştırma da Daubechies dalgacık fonksiyonlarının kullanılması yararlı olacaktır. Haar dalgacıkları ve biortogonal dalgacıklar ise yüksek frekanslı bölgelerde iyi sonuç vermektedir.

Dalgacık ailesi oldukça geniştir. Her bir dalgacığın kendine has özellikleri bulunmaktadır. Görüntülerde ise genel olarak farklı dalgacıkların daha iyi sonuç vereceği bölgeler bulunmaktadır. Dalgacık ailesi ve görüntü üzerindeki etkileri incelenerek görüntü alt görüntü bloklarına ayrıştırılarak bölgesel tabanlı dönüşüm işlemleri gerçekleştirilebilir. Her bir bloğun özelliğine bağlı olarak dalgacık dönüşümü uygulanması tek bir dalgacıkla görüntünün dönüştürülmesi işleminden daha kaliteli bir sonuç vereceği açıktır. Bu işlem gerçekleştirilirken HAKB algoritmasının performansını yükselten; sıfır ağaçlarının yüzdesinin

arttırılması da hesaba katılarak yapılacak optimal bölgesel dalgacık seçimleri başarında önemli avantaj sağlayacaktır.



**KAYNAKLAR**

- Castleman, K.R., (1996), "Digital Image Processing", Prentice Hall, 303-320.
- Daubechies, I., Antonini, M., Barlaud, M., (1992), "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1 April 1992.
- Graps, A., (1995), "An Introduction to Wavelets", IEEE Computational Science & Engineering Summer 1995, 50-57.
- Misiti, M., Misiti, Y., Oppenheim, G., Poggi, J.M., (2001), "Wavelet Toolbox User Guide", The Mathworks Inc.
- Rao, R.M. ve Bopardikar, A.S., (1998), "Wavelet Transforms", Addison-Wesley, 133-170.
- Rioul, O. ve Vetterli, M., (1991), "Wavelets and Signal Processing", IEEE SP Magazine October 1991, 14-32.
- Said, A. ve Pearlman, W.A., (1996), "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6 June 1996.
- Salamon, D., (1997), "Data Compression The Complete Reference", Springer Press, 163-168, 245-249.
- Shapiro, J., (1993), "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE Transactions on Signal Processing December 1993 Vol 40, 10-21.
- Umbaugh, S.E., (1999), "Computer Vision and Image Processing", Prentice Hall, 237-282
- Welstead, S., (1999), "Fractal and Wavelet Image Compression Techniques", SPIE Engineering Press, 93-130.

**INTERNET KAYNAKLARI**

- [1] Saha, S., (2000), "Image Compression -from DCT to Wavelets: A Review", <http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html> .
- [2] Polikar, R., (1996), "The Wavelet Tutorial", <http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html> .
- [3] Valens, C., (1999), "A Really Friendly Guide to Wavelets", <http://www.mindless.com/cvalens/wavelets.html>

**ÖZGEÇMİŞ****Doğum tarihi** 25.03.1977**Doğum yeri** Düzce**Lise** 1991-1995 Düzce Anadolu Öğretmen Lisesi**Lisans** 1995-1999 Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak.  
Elektronik ve Haberleşme Mühendisliği Bölümü**Çalıştığı kurumlar**

1999-2001

Adam Elektronik Ltd Şti.

2001-Devam

YTÜ Fen Bilimleri Enstitüsü Araştırma Görevlisi

