

95042

**YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

# **WAVELET TRANSFORMASYONLARI İLE GÖRÜNTÜ SİKİŞTİRME**

**Müh. Tunç SARAL**

**F.B.E Matematik Mühendisliği Anabilim Dalında  
Hazırlanan**

**YÜKSEK LİSANS TEZİ**

**Tez Danışmanı**

**: Yrd.Doç.Dr. İbrahim EMİROĞLU**

*İ.Emiroğlu*  
*T. Saral*  
*AE*

**İSTANBUL, 2000**

## İÇİNDEKİLER

|                                                    | Sayfa |
|----------------------------------------------------|-------|
| SİMGE LİSTESİ .....                                | v     |
| KISALTMA LİSTESİ .....                             | vii   |
| ŞEKİL LİSTESİ .....                                | viii  |
| ÇİZELGE LİSTESİ .....                              | x     |
| ÖNSÖZ .....                                        | xi    |
| ÖZET .....                                         | xii   |
| ABSTRACT .....                                     | xiii  |
| 1 GİRİŞ .....                                      | 1     |
| 2 GÖRÜNTÜ SIKIŞTIRMA .....                         | 3     |
| 2.1 Giriş .....                                    | 3     |
| 2.1.1 Gereksiz ve İhtiyaç Fazlası Bilgiler .....   | 3     |
| 2.1.2 Veri Sıkıştırma .....                        | 4     |
| 2.2 Kayıpsız Sıkıştırma Teknikleri.....            | 4     |
| 2.2.1 Dictionary-based Teknikler.....              | 5     |
| 2.2.1.1 RLE Kod Çözme .....                        | 5     |
| 2.2.1.2 LZW Kod Çözme .....                        | 5     |
| 2.2.2 İstatistik Kodlama Metodları .....           | 5     |
| 2.2.2.1 Mesajın Bilgi İçeriği.....                 | 5     |
| 2.2.2.2 Huffman Kodlama .....                      | 6     |
| 2.2.3 Binary Görüntü Sıkıştırma Standartları ..... | 7     |
| 2.3 Kayıplı Görüntü Kodlama.....                   | 7     |
| 2.3.1 Sayısal Değerlendirme .....                  | 7     |
| 2.3.2 Oran Bozulma Teorisi.....                    | 7     |
| 2.3.2.1 Uygun Olmayan Gaussian Görüntüler .....    | 9     |
| 2.3.2.2 Uygun Gaussian Görüntüler .....            | 10    |
| 2.4 Dönüşüm Görüntü Kodlama .....                  | 11    |
| 2.4.1 Blok Kod Çözme .....                         | 11    |
| 2.4.2 Bit Ayırma.....                              | 11    |
| 2.4.3 Görüntü Kalite Özellikleri .....             | 12    |
| 2.4.4 Dönüşüm Seçeneği .....                       | 12    |
| 3 DALGACIK ANALİZİ .....                           | 13    |
| 3.1 Dalga ve Dalgacıklar .....                     | 13    |
| 3.2 Zaman - Frekans Analizi .....                  | 14    |
| 3.2.1 Dalgacıklar ve Müzik .....                   | 15    |

|       |                                                |    |
|-------|------------------------------------------------|----|
| 3.3   | Dönüşümler .....                               | 16 |
| 3.3.1 | Dönüşüm Tipleri.....                           | 16 |
| 3.3.2 | Dalgacık Dönüşümü Tipleri.....                 | 17 |
| 3.3.3 | İşaret Sistemi ve Tanımlar .....               | 17 |
| 4     | SÜREKLİ DALGACIK DÖNÜŞÜMÜ .....                | 19 |
| 4.1   | Tek Boyutlu Sürekli Dalgacık Dönüşümü.....     | 19 |
| 4.2   | İki Boyutlu Sürekli Dalgacık Dönüşümü .....    | 20 |
| 4.3   | Filtre Bankası Açıklaması.....                 | 21 |
| 4.4   | İki Boyutlu Filtreler .....                    | 22 |
| 5     | DALGACIK SERİ YAYILIMI.....                    | 24 |
| 5.1   | Dyadic Dalgacıklar .....                       | 24 |
| 5.2   | Tanım .....                                    | 24 |
| 5.3   | Compact Dyadic Dalgacıklar .....               | 25 |
| 5.3.1 | Örnek: Haar Dönüşümü .....                     | 26 |
| 6     | AYRIK DALGACIK DÖNÜŞÜMÜ.....                   | 27 |
| 6.1   | Filtre Bankası Teorisi .....                   | 28 |
| 6.1.1 | İdeal Bant-geçirgen Filtreler .....            | 29 |
| 6.1.2 | Düzgün Bant-geçirgen Filtreler.....            | 31 |
| 6.2   | Çoklu Çözüm Analizi .....                      | 33 |
| 6.2.1 | Piramit Algoritmalar .....                     | 34 |
| 6.2.2 | Laplace Piramit Kodlama .....                  | 34 |
| 6.3   | Alt-bant Kodlama .....                         | 35 |
| 6.3.1 | Daha Düşük Yarı-bant .....                     | 36 |
| 6.3.2 | Daha Yüksek Yarı-bant .....                    | 39 |
| 6.3.3 | İki Bantlı Alt-bant Kodlama ve Kodu Çözme..... | 41 |
| 6.4   | Hızlı Dalgacık Dönüşümü Algoritması .....      | 42 |
| 6.4.1 | Temel Fonksiyonlar .....                       | 43 |
| 6.5   | Ayrık Dalgacık Dönüşümü .....                  | 43 |
| 6.5.1 | Ayna Filtreler .....                           | 44 |
| 6.5.2 | Ölçek Vektör .....                             | 45 |
| 6.5.3 | Dalgacık Vektörü .....                         | 47 |
| 6.5.4 | Dalgacık Dönüşümü Hesaplama .....              | 48 |
| 6.5.5 | Örnekler .....                                 | 49 |
| 6.5.6 | Compact Support'a Sahip Dikey Dalgacıklar..... | 50 |
| 7     | İKİ BOYUTLU AYRIK DALGACIK DÖNÜŞÜMÜ .....      | 52 |
| 7.1   | İleri Dönüşüm.....                             | 52 |
| 7.2   | Ters Dönüşüm .....                             | 55 |
| 7.3   | Örnekler .....                                 | 57 |

|     |                                       |    |
|-----|---------------------------------------|----|
| 8   | BİDİKEY DALGACIK DÖNÜŞÜMLERİ .....    | 59 |
| 8.1 | Uygulama .....                        | 59 |
| 8.2 | Bidikey Dalgacıklar .....             | 60 |
| 8.3 | Bidikey Dalgacıkları Oluşturma.....   | 60 |
| 8.4 | İki Boyutlu Bidikey Dalgacıklar ..... | 62 |
| 9   | SONUÇLAR VE ÖNERİLER .....            | 63 |
| 9.1 | Dalgacık Seçimi .....                 | 63 |
| 9.2 | Sonuçlar ve Uygulama.....             | 63 |
|     | KAYNAKLAR.....                        | 69 |
|     | ÖZGEÇMİŞ .....                        | 70 |



## SİMGE LİSTESİ

|                        |                                              |
|------------------------|----------------------------------------------|
| $\{a_k\}$              | Mesaj kaynağı                                |
| $P(a_k)$               | Karakter karşılaşma olasılığı                |
| $H$                    | Entropi                                      |
| $E\{I(a_k)\}$          | Entropi                                      |
| $L_w(a_k)$             | Kod uzunluğu                                 |
| $D$                    | Bozulma                                      |
| $R(D^*)$               | Bozulma fonksiyonu                           |
| $P(u, v)$              | Görüntünün power spekturumu                  |
| $r_{i,j}$              | $i, j$ 'inci katsayı için gereken bit sayısı |
| $\sigma^2$             | Gauss katsayısı                              |
| $F(s)$                 | Fourier integral dönüşümü                    |
| $F_n$                  | Fourier seri yayılımı                        |
| $G_k$                  | Ayrık Fourier dönüşümü                       |
| $\psi(x)$              | Temel dalgacık                               |
| $W_f(a, b)$            | Tek boyutlu sürekli dalgacık dönüşümü        |
| $W_f(a, b_x, b_y)$     | İki boyutlu sürekli dalgacık dönüşümü        |
| $\tilde{\psi}_1(x)$    | Tek boyutlu filtre bankası                   |
| $\tilde{\psi}_1(x, y)$ | İki boyutlu filtre bankası                   |
| $\delta_{j,k}$         | Kronecker deltası                            |
| $c_{j,k}$              | Transfer katsayısı                           |
| $h_i(x)$               | Filtre bileşeni                              |
| $g_i(x)$               | Filtre çıktısı                               |
| $s_N$                  | Nyquist frekans                              |
| $\phi(t)$              | Tek boyutlu ölçek vektör                     |

|              |                            |
|--------------|----------------------------|
| $\psi(t)$    | Tek boyutlu temel dalgacık |
| $\phi(x, y)$ | İki boyutlu ölçek vektör   |
| $\psi(x, y)$ | İki boyutlu temel dalgacık |
| $f_i^j$      | Görüntü                    |



## **KISALTMA LİSTESİ**

**RLE**      **Run Lenght Encoding**

**LZW**     **Lemple Ziv Welch**

**CCITT**   **Consultative Comittee of the International Telephone and Telegraph**

**SNR**     **Signal to Noise Ratio**



## ŞEKİL LİSTESİ

|                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 2.1 Sayısal değerlendirme .....                                                                                          | 8  |
| Şekil 2.2 Oran bozulma fonksiyonu .....                                                                                        | 9  |
| Şekil 2.3 Oran bozulma eğrileri .....                                                                                          | 10 |
| Şekil 3.1 Dalga ve dalgacıklar.....                                                                                            | 13 |
| Şekil 3.2 Zaman frekans analizi (a) sinyal (b) ton.....                                                                        | 14 |
| Şekil 3.3 Bir görüntünün uzay frekans analizi .....                                                                            | 15 |
| Şekil 3.4 Zaman-frekans alanı üzerine bir örnek; notalar.....                                                                  | 15 |
| Şekil 4.1 Bir dalgacık.....                                                                                                    | 20 |
| Şekil 4.2 Bir sinyalin integral dalgacık dönüşümünün filtre bankası benzerliği .....                                           | 22 |
| Şekil 4.3 Bir görüntünün integral dalgacık dönüşümünün filtre bankası benzerliği.....                                          | 23 |
| Şekil 5.1 Bir dalgacığın Binary ölçekleri ve dyadic çevirileri .....                                                           | 24 |
| Şekil 6.1 Haar dönüşüm temel fonksiyonları .....                                                                               | 27 |
| Şekil 6.2 İki ton patlamalı sinyal ve leke (a) üç bileşen (b) genişlik ve faz spektrumunu .....                                | 28 |
| Şekil 6.3 Frekans axislerini bölerek bant-geçirgen filtreler oluşturma (a) impulse response<br>(b) transfer fonksiyonları..... | 30 |
| Şekil 6.4 Bir bant-geçirgen filtre uygulaması .....                                                                            | 30 |
| Şekil 6.5 Bant-geçirgen filtre çıktıları .....                                                                                 | 31 |
| Şekil 6.6 Düzgün bant-geçirgen filtreler (a) impulse response (b) transfer fonksiyonları ....                                  | 32 |
| Şekil 6.7 Düzgün bant-geçirgen filtre bankası çıktıları .....                                                                  | 32 |
| Şekil 6.8 Laplace piramit kodlama şeması.....                                                                                  | 35 |
| Şekil 6.9 Alt-bant kodlama, daha düşük yarı-bant ile .....                                                                     | 37 |
| Şekil 6.10 Alt-bant kodlama, daha yüksek yarı-bant ile.....                                                                    | 40 |
| Şekil 6.11 İki bantlı alt-bant kodlama ve yeniden oluşturma .....                                                              | 41 |
| Şekil 6.12 Ayırık dalgacık dönüşümü algoritması .....                                                                          | 42 |
| Şekil 6.13 Ters ayırık dalgacık dönüşümü algoritması.....                                                                      | 43 |
| Şekil 6.14 Ölçek fonksiyonu oluşturma.....                                                                                     | 46 |
| Şekil 6.15 Daubechies'in (a) ölçek fonksiyonu (b) dalgacığı.....                                                               | 50 |
| Şekil 6.16 $r=3$ , $r=5$ , $r=7$ ve $r=9$ için dikey dalgacıklar .....                                                         | 51 |
| Şekil 7.1 İki boyutlu ayırık dalgacık dönüşümü (a) gerçek görüntü (b) birinci aşama (c) ikinci<br>aşama (d) üçüncü aşama.....  | 54 |
| Şekil 7.2 Ayırık dalgacık dönüşümü görüntü ayrıştırma işlemi.....                                                              | 55 |
| Şekil 7.3 Frekans alanında ayırık dalgacık dönüşümü .....                                                                      | 56 |
| Şekil 7.4 Ters ayırık dalgacık dönüşümü yeniden görüntü oluşturma işlemi .....                                                 | 56 |

|                                                                                                                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 7.5 İki boyutlu ayırık dalgacık dönüşümünün hesaplanmasına ilişkin örnek .....                                                                                                                                                                                | 57 |
| Şekil 7.6 İki boyutlu ters ayırık dalgacık dönüşümünün hesaplanmasına ilişkin örnek.....                                                                                                                                                                            | 58 |
| Şekil 8.1 Bidikay dalgacık dönüşümünün ayırıştırma ve yeniden oluşturma işlemi.....                                                                                                                                                                                 | 60 |
| Şekil 9.1 Görüntü sıkıştırma örneği, Anna Kournikova (a) 512 x 512, BMP orjinal görüntü<br>(b) İki iterasyon ile ileri dönüşüm (c) %25 sıkıştırma oranı ile (d) Ters dönüşüm (e) %6.25 sıkıştırma oranı ile (f) Ters dönüşüm.....                                   | 64 |
| Şekil 9.2 Görüntü sıkıştırma örneği, parmak izi a) 512 x 512 BMP orjinal görüntü<br>(b) Dört iterasyon ile ileri dönüşüm (c) %25 sıkıştırma oranı ile (d) Ters dönüşüm (e) %6.25 sıkıştırma oranı ile ters dönüşüm (f) %3,82 sıkıştırma oranı ile ters dönüşüm..... | 66 |
| Şekil 9.3 Şekil 9.3 Görüntü sıkıştırma örneği, Akhan a) 512 x 512, BMP orjinal görüntü<br>(b) Dört iterasyon ile ileri dönüşüm (c) %25 sıkıştırma oranı ile (d) Ters dönüşüm .....                                                                                  | 68 |



## ÇİZELGE LİSTESİ

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| Çizelge 6.1 Şekil 6.16'daki dikey dalgacıkların ayırık filtre dizileri ( $r=3,5,7$ ve $9$ )..... | 51 |
| Çizelge 8.1 Şekil 8.1'de gösterilen bidikey dalgacıklar için ayırık dalgacık değerleri.....      | 61 |



## **ÖNSÖZ**

Görüntü sıkıştırma alanında yapılan bu çalışmanın, okul içi ve dışındaki bir takım projelere destek olması dileğini taşımaktayım.

Ayrıca, bu tezin gerçekleştirilmesinde her türlü desteğini esirgemeyen değerli danışmanım Yrd.Doç.Dr. İbrahim Emiroğlu'na ve ayrıca yardımlarından dolayı Y.Müh. Murat Gümüşer'e teşekkür ederim.

Haziran 2000

Tunç Saral



## ÖZET

Dijital görüntü işleme, genelde dijital görüntü bilgileri içeren büyük dosyaların önemli değerleri üzerinde çalışır. Bu değerler, sık sık farklı kullanıcılar ve sistemler içinde saklanmalı veya revize edilmelidir. Bunun için, dijital görüntü verilerinin uygun şartlarda transferi ve saklanması için kullanılması gerekli olan metodlar vardır.

Son yıllarda özellikle görüntü sıkıştırma problemleri, ayrıt saptama ve yapı analizi ile ilgili yeni dönüşüm teknikleri konusunda önemli tezler ortaya atılmıştır. Bu teknikler çoklu analiz, zaman-frekans analizi, piramit algoritması ve dalgacık dönüşümü başlıkları altında toplanabilir. Biz bu çalışmada dalgacık dönüşümü üzerinde çalışacağız.

Görüntü sıkıştırma alanında kullanılan önemli metodlardan biri Fourier dönüşümüdür. Bu doğrultuda dalgacık dönüşümü incelenebilir.

Dalgacık dönüşümü bileşenleri, algoritma ve uygulamaları tez içinde gösterilmektedir. Uygulama aşamasında yapılan işlemler; (1) Orjinal görüntünün (BMP file ) gösterilmesi, (2) görüntünün yoğunluğuna göre 2 veya 4 iterasyon ile ileri dönüşüm yapılması, (3) İleri dönüşüm uygulanmış görüntünün, yoğunluğuna göre %25'i, %6,25'i veya %3.82'si kullanılarak sıkıştırma oranının belirlenmesi, (4) görüntünün ters dönüşüm ile tekrar oluşturulması.

Uygulama çalışmaları sonucunda, görüntüler arasındaki fark değişmektedir. Sıkıştırmanın ne derece başarılı olduğu görüntü kalitesi ile irdelenebilir.

## ABSTRACT

Digital image processing generally creates significant numbers of large files containing digital image data. Very often, these must be archived or exchanged among different users and systems. This calls for efficient methods for the storage and transfer of digital image data files.

Considerable interest has arisen in recent years regarding new transform techniques that specifically address the problems of image compression, edge and feature detection, and texture analysis. These techniques come under the headings of multi-resolution analysis, time-frequency analysis, pyramid algorithms, and wavelet transforms. In this study, we will work at The Wavelet Transforms.

The significant one of these methods is Fourier Transform. Resemble to Fourier Transforms, we can study with Wavelet Transforms.

Wavelet Transforms components, algorithms and practices is shown in this study. The steps of practice study ; (1) original image (BMP file) (2) according to pixels density, forward transform by two or four iterations (3) to be determined with compression ratio, %25, %6,25 or %3,82 (4) inverse transform(backward) or reconstruction

Finally, the difference ratio between originally and reconstructed image is changeable. It is generally able to seem with vision quality.



## 1. GİRİŞ

Dalgacık dönüşümü, işaretlerin değişik ölçeklerde incelenmesi amacıyla yapılan çalışmalardan doğmuştur. 1980'li yıllarda Fransız araştırmacılardan J. Morlet, A. Grossman ve Y. Meyer tarafından ortaya atılmıştır. Daha sonraları, I. Daubechies ve S.G. Mallat, dalgacık teorisini işaret işleme alanında geliştirmişlerdir. Günümüzde, birbirinden farklı birçok alanda dalgacık dönüşümünden yararlanılmaktadır.

Dalgacık dönüşümü, diğer dönüşümler ile kıyaslandığında birçok avantaja sahiptir. Kararlı olmayan işaretlerin analizinde kullanılabilmesi, filtre katsayılarının kısa sürede sıfıra düşmesi, frekans spektrumunda sabit bant genişliğine sahip olması ve çok ölçekli işaret ayrıştırma özelliğine sahip olması bunlardan birkaçıdır.

Çok ölçekli ayrıt saptama yöntemi olarak dalgacık dönüşümü, S.G. Mallat ve Zong tarafından geliştirilmiş ve yeniden oluşturma işlemi için algoritmalar belirlenmiştir.

Bu çalışmada aşağıdaki kronoloji doğrultusunda çalışmamızı yürüteceğiz:

Bölüm 2'de görüntü sıkıştırma işleminin genel özelliklerinden bahsedeceğiz. Görüntü sıkıştırma teknikleri, sıkıştırma standartları hakkında açıklamalarda bulunacağız.

Bölüm 3'te Dalgacık Analizi nedir?, bileşenleri nelerdir? gibi sorulara cevap arayacağız. Dalga ve dalgacıklar arasındaki ayrımı saptayıp, kullanım alanlarını örnekler ile açıklayacağız.

Bölüm 4'te ise dalgacık dönüşüm tiplerinden birincisi olan, tek ve çift boyutlu sürekli dalgacık dönüşümünü ve filtreler ile ilgili tanımları belirleyeceğiz.

Bölüm 5'te dalgacık dönüşüm tiplerinden ikincisi olan dalgacık seri yayılımını açıklayacağız.

Bölüm 6'da dalgacık dönüşüm tiplerinden sonuncusu ve en önemlisi olan ayrık dalgacık dönüşümünü inceleyeceğiz. Filtre çeşitleri ve kullanım alanlarını eşitlikler ve şekiller ile irdedeceğiz. Çoklu çözüm analizi ve Laplace Piramit kodlamadan kısaca bahsedeceğiz. Değişik bantlarda kodlama ve kod çözme çalışmalarını ve algoritmalarını oluşturacağız.

Bölüm 7'de üzerinde ileriki bölümlerde uygulamasını da yapacağımız iki boyutlu ayrık dalgacık dönüşümünü örneklerle açıklayacağız. Dönüşümün aşamaları incelenecek, uygulama öncesi muhtemel sonuçlar hakkında görüş bildirilecektir.

Bölüm 8’de bidikey dalgacıklardan kısaca bahsedilecek, bölüm 9’da dalgacık seçimi belirlenerek, uygulama yapılacaktır.



## 2. GÖRÜNTÜ SIKIŞTIRMA

### 2.1 Giriş

Madem ki dijital görüntü yapısı gereği tamamıyla yoğundur, boyutunu küçültmek, daha pratik olmasının yanında hızlı ve kullanışlı çözümler oluşturabilir. Görüntü sıkıştırmanın etkinliği ve faaliyet alanı, gereksiz ve ihtiyaç dışı bilgileri ortadan kaldırmaktır.

#### 2.1.1 Gereksiz ve İhtiyaç Fazlası Bilgiler

Görüntü dosyaları kayde değer büyüklükte gereksiz ve ihtiyaç fazlası bilgi içerir. Bunları yalın hale getirme, görüntü sıkıştırma teknikleri ile yapılır. Gereksiz ve ihtiyaç fazlası bilgiler arasındaki ayrım bir örnek ile açıklanabilir.

Ender Bey, Avrupa'ya yaptığı bir iş seyahatinin sonunda, evine Marmaris'e dönecektir, bununla birlikte Marmaris'e indiği zaman hava alanından evine nasıl gideceğini henüz organize etmemiştir. Seyahatinin son gününde, İstanbul'dan bir mesaj alır. Mesajda:

“ Eşiniz, Sinem Hanım yarın akşam sizi Marmaris'te, Dalaman Havalimanı'nda saat altıyı beş geçe karşılayacak. “

Bu mesajın nasıl kısaltılabileceğini düşünelim. Mesajı incelediğimizde Ender Bey tarafından normalde bilinen birtakım bilgilerin varolduğunu görüyoruz. Ender Bey, Dalaman Havalimanı'nın Marmaris'te olduğunu biliyor. Bununla birlikte eşinin isminin Sinem olduğunu da biliyor. Bunlar verilmesi gereksiz bilgilerdir. Mesaj gereksiz bilgilerden arındırılmış hali ile aşağıdaki gibidir.

“ Eşiniz, yarın akşam sizi Dalaman Havalimanı'nda saat altıyı beş geçe karşılayacak. “

Biz burada, mesajı herhangi bir anlam kaybı olmaksızın kısaltmayı veya sıkıştırmayı başardık. Verilmek istenen ana mesajı vurgulayarak, ciddi bir anlam kaybı yaratmaksızın daha da fazla sıkıştırabiliriz. Mesaj tekrar düzenlenirse:

“ Sinem yarın akşam, sizi saat altıda Dalaman'da karşılayacak. “

Bu cümlede biraz anlam kaybı gerçekleşti, fakat bu çok önemli değildir. Ender Bey, bahsi geçen Sinem'in eşi olduğunu tahmin etmek zorunda ( belki de kuzeninin ismi de Sinem'di. ), ineceği hava alanının Dalaman olduğunu biliyor ama eşinin onu Dalaman havaalanında değil de Dalaman'daki ofislerinde karşılayacağını da düşünebilir. Buluşma saati içinde 5 dakikalık bir süre ihmal edilebilir. Buna rağmen kısa bir mesaj vermek zorundaysanız, bu yeterli

olacaktır.

### 2.1.2 Veri Sıkıştırma

Veri sıkıştırma teknikleri bir veri dosyasını küçültürken, gerçek görüntü tamamıyla veya yaklaşık olarak tekrardan yaratılabilecek düzeyde değerlendirilir ve bunu yaparken varolan gereksizliği ve ihtiyaç fazlalığı kullanılır. Sonradan yaratılan ile gerçek görüntü arasındaki fark, sıkıştırmanın verimlilik ölçüsünü verir.

Veri sıkıştırma algortimaları, kayıplı ve kayıpsız olabilir. Kayıpsız algoritmalar, sadece gereksiz bilgiyi çıkarır, tamamıyla kayıpsız bir şekilde bilgiyi tekrar yaratabilir. Kayıplı algoritmalar, olabildiğince ihtiyaç dışı bilgiyi çıkarır, verinin tekrar yaratılması durumunda yaklaşık bir çözüm ile karşılaşılır. Kayıplı sıkıştırma algoritmaları, daha yüksek sıkıştırma oranlarına ulaşırlar. Bazı görüntüler değişime izin vermezler, kayıpsız sıkıştırma bu tip görüntüler için kullanılır.

Sıkıştırma ve yeniden yaratma için gerekli zaman önemli değildir. En yoğun sıkıştırmayı gerçekleştiren algoritma genellikle en hızlı olanı da değildir. Bazı görüntü sıkıştırma programları, kullanıcıya kayıba karşı kayıpsız seçimler ve sıkıştırma oranına karşı hızlı seçenekler sunar. Kayıplı algoritmalar genellikle aslına uygun yeniden yapılanma ve bütünlük derecesi arasındaki ilişki bağlamında seçenekler sunar.

### 2.2 Kayıpsız Sıkıştırma Teknikleri

Kayıpsız veri sıkıştırma algortimaları iki kategoride incelenebilir; dictionary-based teknikler ve istatistik metodlar. Dictionary-based teknikler, fixed-length kodlar ( genellikle 12'den 16 bit'e kadar olanlar ) içeren sıkıştırılmış bir dosyayı oluşturur, kodların her biri belli bir düzende temsil edilir.

İstatistik metodlar, dosyada daha çok karşılaşılan karakterleri temsil etmek için daha az sayıda bit kullanır. Görüntü üzerine olmasa da bu tür sıkıştırma veya kısaltma örneği olarak Morse alfabesinin kullanımını verebiliriz. Samuel F.B. Morse, zaman kazanmak amacıyla İngilizce de sık kullanılan " e " harfini, telgraf aletindeki vuruşlarda single dot olarak, nadir kuallanılan " z " harfini dash, dash, dot, dot olarak kullanmıştır.

## 2.2.1 Dictionary-Based Teknikler

### 2.2.1.1 RLE Kod Çözme

En basit dictionary-based tekniği, run-length(RLE) kod çözme tekniğidir. Görüntüler, özellikle az sayıda gri tona sahip olanlar, aynı gri tonda veya renkte komşu pixellerden oluşan bölgeler içerirler. Satır satır saklanmış bir görüntüde, aynı gri tondaki pixeller serisi, **run** olarak adlandırılır. Pixel değeri ve sayısı aynı değeri birçok defa tekrarlamak yerine bir kod ile saklanabilir. Buna run-length kod çözme denir. Bu başarılı bir sıkıştırma elde etmemizi sağlar. Örneğin sabit bir zemin üzerinde bulunan objelerin görüntü ve grafikleri gibi. Olumsuz koşullar altında ( örneğin her pixelin komşusundan farklı olması durumunda ) RLE genellikle dosya boyutunu iki katına çıkarabilir.

### 2.2.1.2 LZW Kod Çözme

Kayıpsız bir teknik olan LZ kodlama sistemi, ilk olarak Lemple ve Ziv tarafından tanımlanmıştır. Welch tarafından kullanımı yaygınlaştırılmış, LZW algoritması olarak tescil edilmiştir. Bu teknik de, aynı RLE gibi karakter dizilerini kodlayarak işlevini gerçekleştirir. Bununla birlikte, RLE'den farklı olarak, dosyanın kodunu çözmek için bir dizi tablosu ve benzer kodları oluşturur. Örneğin 8 bitlik bir dosya, 12 bit'likler ile kodlanabilir. 4,096 belli kodun 256 tanesi single byte olarak belirlenir. Kalan 3,840 tanesi ise, veri içinde sıkıştırma süresince eşleşen dizilere ayrılır.

Dizi tablo içine ilk kez alındığında, kendine atanan kod ile birlikte tamamıyla saklanır. Bu işlemlerden sonra dizi ile tekrar karşılaşıldığında sadece kodu saklanır. Bu da dosyadaki fazlalığı ortadan kaldırır. Geri dönüştürme algoritması, sıkıştırılmış dosya bilgisinden görüntüyü kayıpsız yeniden oluşturabilir.

## 2.2.2 İstatistik Kodlama Metodları

### 2.2.2.1 Mesajın Bilgi İçeriği

İstatistik kodlama tekniklerini incelemeden önce, bilgi içeriğinin klasik teorisine bir göz atalım.  $\{a_k\}$ ,  $k=0,1,2,...K-1$  serisini kullanan, belleksiz bir mesaj kaynağı düşünün. Buradaki her bir  $a_k$  serinin bir karakteridir. Bunun yanında farzedelim ki, her bir karakterin karşılaşma olasılığı biliniyor ve  $P(a_k)$  olarak tanımlanıyor. Kaynak içindeki karakterlerin düzeni

önemsizdir. Sadece mesajın içinde olmaları göz önünde bulundurulur.

Shanon bir mesaj içindeki  $a_k$  cinsinden gösterilen bir ölçü tanımlamıştır.

$$I(a_k) = -\log[P(a_k)] \quad (2.1)$$

Bu ölçü yeterlidir. Çünkü, (a) Bir sembol ne kadar az sayıda ise bu sembolün varlığı mesaja o oranda fazla katkıda bulunur. (b) bir mesajdaki ana bilgi, mesajı oluşturan sembollerdeki bilgilerin toplamıdır. Her mesajda her zaman kullanılan bir sembol, örneğin  $P(a_i) = 1$ , bir şey ifade etmez  $I(a_i) = 0$ .

Mesaj kaynağının entropisi, kaynak tarafından oluşturulan mesajdaki sembol başına ortalama bilgi içeriğini belirler.

$$H = E\{I(a_k)\} = -\sum_{k=0}^{K-1} P(a_k) \log[P(a_k)] \quad (2.2)$$

Mesaj kaynağının entropisi negatif değildir, bütün semboller benzer ise entropi maximum değerini alır. Mesajda, belli bir kodlama algoritması uygulandıktan sonra kalan fazla bilgi, kodun ortalama kelime uzunluğu ile kaynağın entropisinin farkıdır. Yani;

$$R = E\{L_w(a_k)\} - H \quad (2.3)$$

Burada  $L_w(a_k)$  sembolünü gösteren kelime kodunun uzunluğudur ( ikilik kodlama için bit cinsinden ). Ortalama kelime uzunluğu mesaj kaynağının entropisine eşit olduğu anda, bütün gereksiz bilgilerin ortadan kalktığı görülür. Buna ulaşmak için, aşağıda tanımlanan şekilde kelime uzunlukları bulunmalıdır.

$$L_w(a_k) = -\log[P(a_k)] \quad (2.4)$$

Bu formül, ortalama kelime uzunluğu için alt sınırı belirler.

### 2.2.2.2 Huffman Kodlama

Huffman kodlama her zaman minimum kalanlı ve değişken uzunluklu kod bulan, kayıpsız bir istatistik metottur. Bu kodlama metodu, daha çok karşılaşılan değerleri daha az bit ile, daha az karşılaşılan değerleri daha çok bit ile gösteren bir ikili kodlama ağacı kullanır. Statik Huffman kodlama, sıkıştırılmadan önce oluşturulmuş bir kodlama ağacı kullanır. Burada olası veri değerlerinin karşılaşma olasılıklarının olduğu bir tablo kullanılır. Dinamik kodlama ise, kodlama ağacını sıkıştırma işlemi sırasında oluşturur.

Daha ileri istatistiksel metodlar, daha yüksek sıkıştırma oranları sağlayabilir, fakat kodlama ve kod çözme süreleri artar.

### 2.2.3 Binary Görüntü Sıkıştırma Standartları

Binary görüntü sıkıştırma işleminde CCITT ( Consultative Committee of the International Telephone and Telegraph ) tarafından özellikle faks iletişimi için oluşturulmuş olan iki adet standart yöntem kullanılmaktadır. Grup 3 standardı, run-length'i belirlemek için Huffman kod kullanan yatay RLE üzerine kurulmuştur.

RLE'de siyahtan beyaza veya beyazdan siyaha geçiş yerleri önemlidir. Tipik bir dökümanda bu kodlama metodları ile ulaşılan sıkıştırma oranı yaklaşık 15'dir.

## 2.3 Kayıplı Görüntü Kodlama

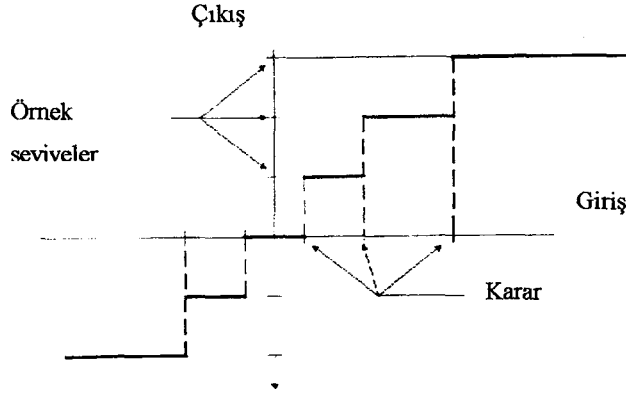
### 2.3.1 Sayısal Değerlendirme

Veri hacmini azaltmanın en basit yollarından biri, görüntüdeki gri ton sayısını azaltmaktır. Şekil 2.1, bir niceliğin gri-ölçek dönüşüm fonksiyonunu göstermektedir. Giriş, iki karar eşiğinin arasında olduğunda, çıkış değeri olarak uygun bir ton belirlenmektedir.

Lloyd ve Max, gri ton histogramı ile birlikte olasılık dağılım fonksiyonu bilinen bir görüntü için, niceleme işlemi tarafından oluşturulan hatayı minimize eden yöntemin, aşağıdaki iki adet birbirine bağımlı özelliği sağladığını göstermiştir. (1) Her karar eşiği, iki bitişik gösterge tonunun arasına düşer. (2) Her gösterge tonu, iki ardışık karar eşiği arasında ortanın merkezine düşer. Bu, karar eşikleri ve gösterge tonlarını belirlemeye yarayan, normalde ardışık olarak çözülebilen bir denklem sistemini verir.

### 2.3.2 Oran Bozulma Teorisi

Oran bozulma teorisi ,sabit uzunluklu kodlama şemasındaki bozulma(tekrar oluşturma hatası) ile veri oranı( örneğin, pixel başına bit sayısı ) arasında ilişki kurar. Teori, girilen görüntünün sürekli olduğunu kabul ettiğinden, değerlendirme hatası nedeniyle bozulma, sonlu veri oranıyla asla sıfır olamaz.



Şekil 2.1 Sayısal değerlendirme

Kayıplı bir sıkıştırma yöntemi kullanıldığında tekrar oluşturulan  $g(x,y)$  görüntüsü, orjinal  $f(x,y)$  görüntüsünden farklıdır. Bu fark(bozulma) aşağıdaki gibi gösterilir.

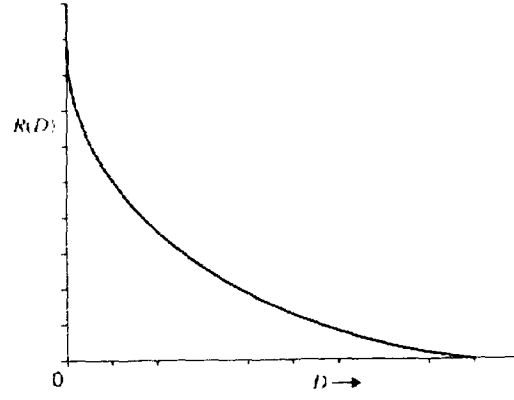
$$D = E\{[f(x, y) - g(x, y)]^2\} \quad (2.5)$$

Eğer  $D^*$ 'a maximum olası bozulma miktarı dersek, kodlama şemasında gerekli bit oranının en düşük sınırı  $R(D^*)$  fonksiyonu,  $D^*$ 'ın monoton azalan bir fonksiyonudur.  $R(D^*)$ , oran bozulma fonksiyonu olarak adlandırılır ( Şekil 2.2 ). Bazen de bunun ters fonksiyonu kullanılır.

Tekrar oluşturma hatasının entropisi aşağıdaki şekilde sınırlandırılmıştır.

$$H[f(x, y) - g(x, y)] \leq \frac{1}{2} \log(2\pi e D^*) \quad (2.6)$$

Fark görüntüsünün istatistiksel olarak bağımsız pixelleri ve bir Gauss katsayısı var ise bağıntıda eşitlik sağlanır. Bu eşitlik bize en iyi kodlama yönteminin, sadece beyaz lekeli bir hata görüntüsü ürettiğini gösterir. Böylece, orjinal ve kodu çözülmüş görüntü arasındaki fark incelenerek, bir görüntü kodlayıcı belirlenebilir. Bu fark görüntüsündeki her tanımlanabilir yapı, kodlayıcının optimalliğinin kanıtıdır.



Şekil 2.2 Oran bozulma fonksiyonu

### 2.3.2.1 Uygun Olmayan Gaussian Görüntüler

İstatiksel olarak bağımsız pixelleri ve  $\sigma^2$  Gauss katsayısı olan, bir kodlanmış  $f(x,y)$  görüntüsü düşünelim. Bu durumda oran bozulma fonksiyonu

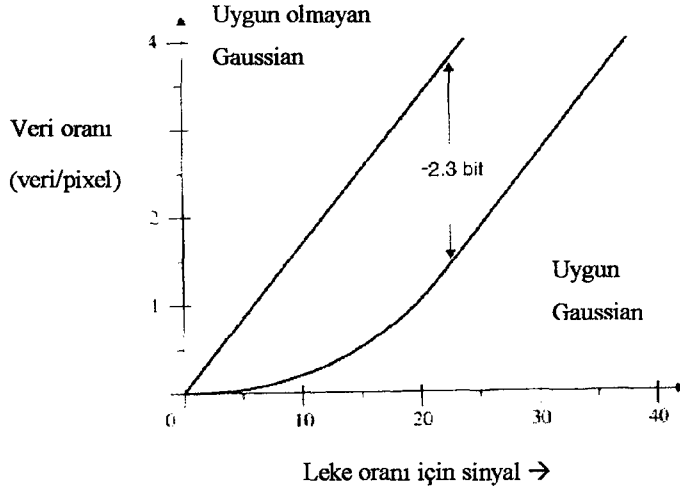
$$R(D) = \frac{1}{2} \max \left[ \log_2 \left( \frac{\sigma^2}{D} \right), 0 \right] \quad (2.7)$$

pixel başına bittir. Kodlama yönteminin leke oranını veren eşitliği, aşağıdaki şekilde tanımlayabiliriz.

$$SNR = 10 \log_{10} \left( \frac{\sigma^2}{D} \right) \quad (2.8)$$

Şekil 2.3 'de görüldüğü gibi, bit başına 6 dB eğimli olan düzgün doğrusal bir ilişki vardır. Uzunluk kodunun her bir ek biti, SNR'ye 6 dB eklemektedir.

Birçok görüntü, ne Gauss histogramına ne de uncorrelated pixellere sahip olmadığından, bu en zor kodlama durumudur. Yani, gaussian olmayan ve uygun kaynaklar için gereken bit oranı, aynı bozulma seviyesi için gerekenden her zaman daha düşüktür.



Şekil 2.3 Oran bozulma eğrileri

### 2.3.2.2 Uygun Gaussian Görüntüler

Görüntüdeki komşu pixeller arasındaki korelasyon, görüntünün autokorelasyon fonksiyonu veya bunun dengi güç spectrumu tarafından belirlenir. Uygun Gaussian görüntülerin oran bozulma fonksiyonu kapalı biçimde yazılamadığından, hem bozulma hem de oran, başka bir  $\theta$  parametresinin fonksiyonu olarak yazılabilir.

$$D(\theta) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \min[\theta, P_f(u, v)] du dv \quad (2.9)$$

$$R(\theta) = \frac{1}{8\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \max\left[\theta, \log\left(\frac{P_f(u, v)}{\theta}\right)\right] du dv \quad (2.10)$$

Burada  $P(u, v)$ ,  $f(x, y)$  görüntüsünün power spektrumudur.

Örneğin, görüntü bir Gaussian od'ye ve üstel olarak bozulan autokorelasyon fonksiyonuna sahipse, SNR oran bozulma eğrisi, daha yüksek bit oranları için uygun olmayan durumun yaklaşık 2.3 bit aşağısına düşer (Şekil 2.3). Burada bitişik pixeller arasındaki korelasyon, bit oranını pixel başına iki bitten daha fazla düşürmek için kullanılabilir. Ne yazık ki, oran bozulma teorisi bu işin nasıl yapılacağını belirleyemez.

## 2.4 Dönüşüm Görüntü Kodlama

Ayrık görüntü dönüşümlerinin en yararlı uygulamaları, görüntü sıkıştırma alanında. Diğer sıkıştırma teknikleri ile beraber kullanıldıklarında, görüntülerin ve resimlerin pratik olarak iletimine, saklanmasına ve gösterimine olanak sağlarlar.

Elimizde kodlanacak bir dizi resim olduğunu düşünelim. Bu görüntüleri dönüştürebiliriz, sifra yakın katsayıları eleyebiliriz ve küçük olanlara kabaca bir değer atayabiliriz, böylece veri iletim ve saklama kaynaklarını, görüntü hakkında en çok bilgi içeren katsayılar üzerinde yoğunlaştırabiliriz. Daha sonra görüntü tekrar oluşturulduğunda, önemsiz içerik kaybolacaktır. Bu yaklaşıma dönüşüm görüntü kodlama adı verilir.

### 2.4.1 Blok Kod Çözme

Görüntü genellikle 8 x 8 veya 16 x 16 pixellik bloklara ayrılır ve her bloğa ayrı dönüşüm uygulanır. Bu yaklaşım özvektörlerin hesaplanmasını gerektiren durumlarda dönüşüm işlemini basitleştirir. Bazı dönüşüm katsayılarının sonradan eliminasyonu ve diğerlerinin kabaca değerlendirilmesi, blokların kenarlarındaki gri tonlarında farkedilebilir değişikliklere neden olur. Buna manuel bloklama denir ve blok sınırlarını istenmeyen şekilde ortaya çıkarır.

### 2.4.2 Bit Ayırma

Görüntü bloğunun oluşumu sonucu, ortaya çıkan katsayıların kodlanması için kaç tane bitin kullanılması gerektiği belirlenmelidir. Oran bozulma teorisini kullanarak ve bloğun Gaussian rasgele dağılım olduğunu kabul ederek

$$r_{i,j} = \frac{1}{2} \max \left[ \log_2 \left( \frac{\sigma_{i,j}^2}{D} \right), 0 \right] \quad (2.11)$$

elde edilir. Bu sonuç,  $i,j$ 'inci katsayı için gereken bit sayısıdır. Burada  $\sigma_{i,j}^2$ , bu katsayının varyansını ve D, maximum olası bozulmayı gösterir.

### 2.4.3 Görüntü Kalite Özellikleri

Mevcut sıkıştırma oranı ile kodlama sonucu kaybedilen bilgi miktarı arasında bir ilişki vardır.

Normalde görüntü kaybının kabul edilebilir veya edilemez olduğuna insan gözü karar verir.

Dönüştürülmüş görüntünün sayısal ölçümü ile insanın görüş açısı birbirini tutmayabilir.

### 2.4.4 Dönüşüm Seçeneği

Ne derecede özel bir dönüşümün veri sıkıştırmayı gerçekleştireceği, hem dönüşüme hem de sıkıştırılacak görüntünün doğasına bağlıdır. Bir görüntü kodlama yönteminin pratikliği, kodlama ve kod çözme adımlarının hesaplanabilir çalışmasına olduğu kadar, elde edilen sıkıştırma derecesine de bağlıdır.



### 3. DALGACIK ANALİZİ

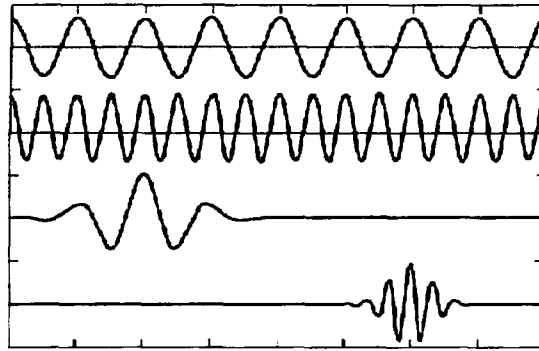
#### 3.1 Dalga ve Dalgacıklar

Fourier dönüşümünü hatırlarsak, dönüşümün dikey temel fonksiyonları, sinüs dalgalar olarak kabul edilir. İntegral dönüşümleri için bu fonksiyonlar, her iki doğrultuda da sonsuzluğa yayılır. Ayırık Fourier dönüşümünün temel vektörleri sıfır da olmaz, bu bilgiler doğrultusunda compact support'a sahip olmadığı görülmektedir.

Mukayese edersek, kısa süreli, geçici sinyaller sadece kısa bir aralık boyunca sıfır değildir. Ayrıca görüntülerdeki birçok önemli özellik, uzaysal bir pozisyon içinde yüksek derecede sınırlanmıştır. Bu gibi bileşenler, herhangi bir Fourier fonksiyonuna benzemez. Bu da Fourier ve diğer dalga dönüşümlerini, kısa süreli, sınırlı görüntü ve sinyal sıkıştırma ve analizi konularında daha düşük optimal seçenekler arasına sokar.

Açıkçası, Fourier dönüşümünü herhangi bir analitik fonksiyonu değerlendirebilen, bir sinüs toplamı olarak düşünebiliriz. Bununla birlikte, birçok aralık boyunca sıfır olan bir fonksiyon yaratma imkanı sağlanır. Bu tabii ki tersi alınabilir bir dönüşüm için geçerli bir yoldur.

Matematikçi ve mühendisler, böyle bir eksiklik ile mücadele etmek için, temel fonksiyonları olan dönüşümler kullanarak birtakım yaklaşık çözümler araştırmışlardır. Bu temel fonksiyonlar, pozisyon içindeki en az frekans kadar değişirler. Sınırlı anların dalgalarıdır ve dalgacık olarak adlandırılırlar. Dalgacıklara temel olan dönüşümlere, dalgacık dönüşümü adı verilir. Şekil 3.1 'de dalga ve dalgacıklar arasındaki fark gözlemlenebilir. Üstteki iki eğri cosinüs dalgalarıdır ve duraksama yoktur. Altındaki iki eğri ise dalgacıktır ve her iki frekans ve pozisyonda yatay doğrultu boyunca sürelidir.



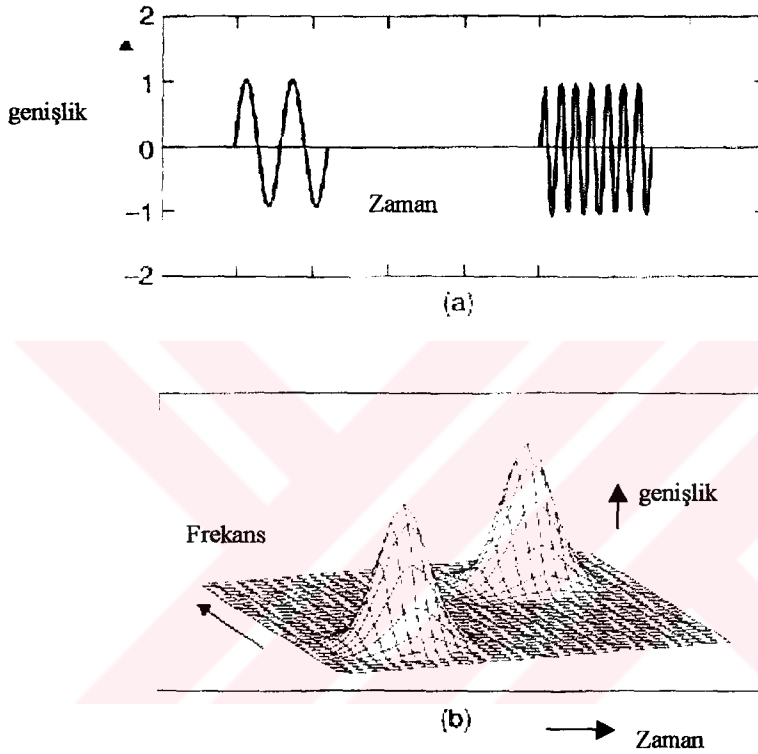
Şekil 3.1 Dalga ve dalgacıklar

Haar dönüşümü, dalgacık dönüşümü konusunda ilk örnektir. İlginç bir köşeli sinyal özelliği

olan Haar fonksiyonu, en eski ve en basit dalgacıdır.

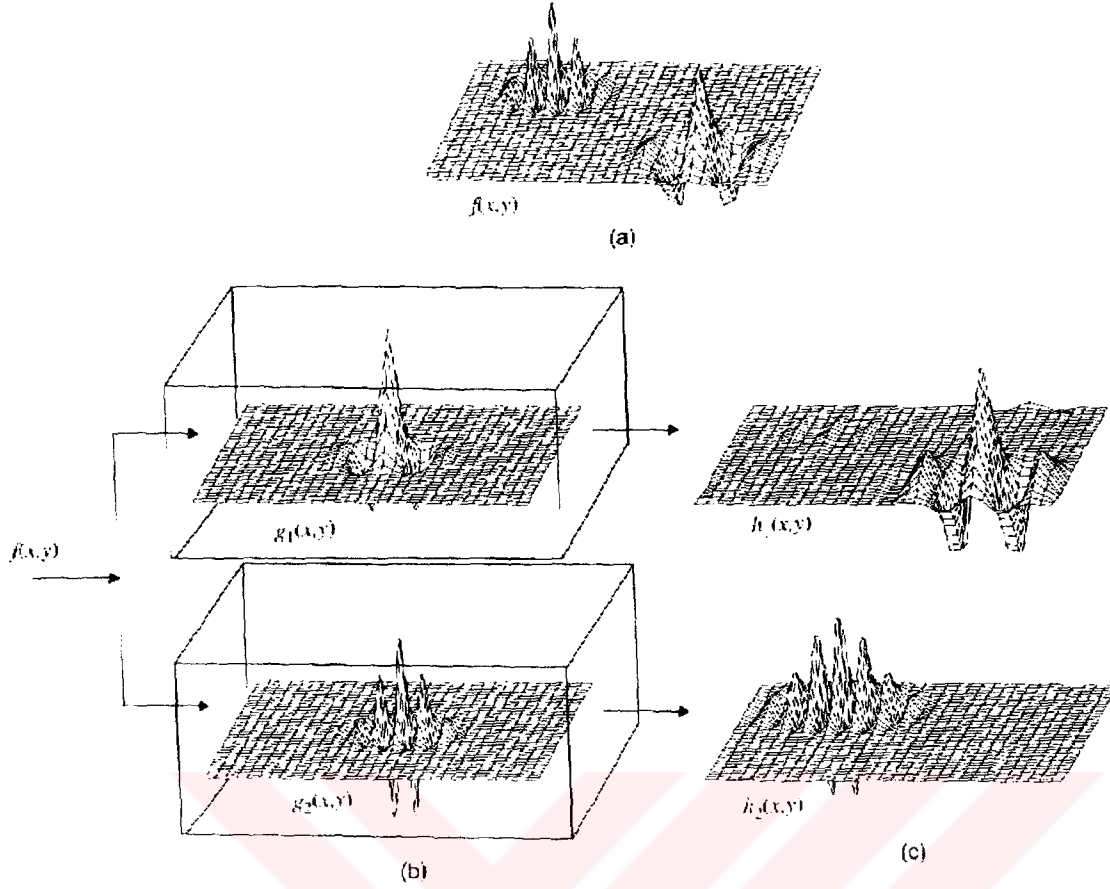
### 3.2 Zaman – Frekans Analizi

Sinyal denetleme işlemi, iki boyutlu bir zaman frekans analizinin terimleri içinde, sinyal analizi hususunda önemli bir çalışma gerektirir. Bu yaklaşım aslında dalgacık dönüşümlerinden önce düşünülmüştür ve dalgacık dönüşümüne temel olmuştur. Bu yaklaşıma göre, bir sinyal haritasının her süreksiz bileşeni, o bileşenin etkili frekansına ve zamanına tekabül eder. Şekil 3.2 .



Şekil 3.2 Zaman frekans analizi (a) sinyal  
(b) ton

Görüntü analizinde uzay üç boyutludur ve bir görüntü kümesi olarak görünür. Belirli bir yere sınırlanan bileşen, etkili frekansına tekabül eden küme içinde en üst düzeyde görünecektir. Şekil 3.3 , iki sınırlı bileşen içeren bir görüntüyü göstermektedir. Buna karşın iki filtre, iki bileşeni tamamiyle izole etmektedir. Bu yaklaşım Gabor'un sınırlanmış Fourier dönüşümü ile başlamış, sonrada sırasıyla kısa zamanlı Fourier dönüşümüne ve altbant kodlama işlemine temel oluşturmıştır.



Şekil 3.3 Bir görüntünün uzay-frekans analizi

### 3.2.1 Dalgacıklar ve Müzik

Şekil 3.4'teki gibi bir nota grubu düşünelim. İlk başta, iki boyutlu bir zaman frekans uzayı olarak görülebilir. Frekans aşağıdan yukarıya artar, zamanı geldiğinde sağa doğru ilerler. Kolon üzerindeki herbir nota bir dalgacığa tekabül eder. Her dalgacık süresi belirli bir tipte nota ile kodlanır.

Eğer bir melodiyi notalara dökersek bu bir dalgacık dönüşümü işlemidir. Benzer bir şekilde notalara bakılarak müziğin icra edilmesi ise bir ters dalgacık dönüşümü işlemidir. Bu, bir zaman frekans tablosundan sinyalleri tekrar oluşturma işlemidir.



Şekil 3.4 Zaman-frekans alanı üzerine bir örnek; notalar

### 3.3 Dönüşümler

Bir dönüşümdeki her bir katsayının, giriş fonksiyonu ve temel fonksiyonlardan biri arasındaki iç çarpım alınarak belirlendiğini hatırlayalım. Bu değer, giriş fonksiyonu ve özel temel fonksiyon arasındaki benzerliğin derecesini gösterir. Eğer temel fonksiyonlar dikey ise, iç çarpım birbirleri ile tamamen farklı olan iki temel fonksiyon arasında alınır. Görüntü veya sinyal bir veya birkaç temel fonksiyona benzer bileşenler olarak oluşturulursa, bu özet sadece birkaç önemli genişlikli terime ihtiyaç duyar. Birçok terim ihmal edilir ve sinyal veya görüntü sadece birkaç dönüşüm katsayısı tarafından gösterilir.

Bundan başka, sinyal veya görüntü içindeki bileşenler bir veya birkaç temel fonksiyona benzer iseler, bu bileşenler kendilerini sadece temel fonksiyonları için büyük katsayılarla göstereceklerdir. Böylece dönüşüm içinde “bulması kolay” bir duruma geleceklerdir. Sonuç olarak, istenmeyen bir bileşen bir veya birkaç temel fonksiyona benzer olursa, bulunması kolay olacaktır. Aynı zamanda uygun dönüşüm katsayılarını azaltarak taşınması da kolay olacaktır.

Bütün bu bilgilerden yola çıkarak, dönüştürülmüş sinyal veya görüntü bileşenlerine benzer temel fonksiyonlar ile, kullanılan dönüşümler içinde potansiyel değer olduğu sonucuna varıyoruz. Bununla birlikte süreksiz bileşenlerin Fourier ve diğer dönüşüm tiplerinin temel fonksiyonlarına benzer olamayacağını da not edelim.

#### 3.3.1 Dönüşüm Tipleri

Fourier dönüşüm tekniğinin bilindiği gibi üç farklı tipi vardır. Bunlar; Fourier integral dönüşümü, Fourier seri yayılımı ve ayrık Fourier dönüşümü.

Fourier integral dönüşümü, iki sürekli fonksiyonu birleştirir. Bu dönüşüm ve tersi aşağıdaki gibi tek boyutludur.

$$F(s) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi(xs)} dx \quad \text{ve} \quad f(x) = \int_{-\infty}^{\infty} F(s) e^{-j2\pi(xs)} ds \quad (3.1)$$

Fourier seri yayılımı, Fourier katsayılarının sonlu yada sonsuz bir dizisi olarak periyodik bir fonksiyonu gösterir. Bu yayılım ve tersi  $s = n\Delta s$ , bir ayrık değişken oluşturularak elde edilir.

$$F_n = F(n\Delta s) = \int_0^L f(x) e^{-j2\pi(n\Delta s x)} dx \quad \text{ve} \quad f(x) = \Delta s \sum_{n=0}^{\infty} F_n e^{j2\pi(n\Delta s x)} \quad (3.2)$$

$L$  fonksiyonun periyodu ve  $\Delta s = 1/L$  'dir.

Ayrık Fourier dönüşümü, örneklenmiş bir spectrum ile örneklenmiş bir fonksiyon sunar ve bağımsız örneklerin sayısı her iki alanda da aynıdır. Bu sayı,  $x = i\Delta x$  bir ayrık değişkeni oluşturularak elde edilir. Eğer  $g(x)$  bant limitli ve örneklenmiş ise buradan  $g_i = g(i\Delta x)$  ve

$$G_k = \frac{1}{N} \sum_{i=0}^{N-1} g_i e^{-j2\pi k \frac{i}{N}} \text{ ve } g_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} G_k e^{j2\pi k \frac{i}{N}} \text{ 'dir.} \quad (3.3)$$

Bütün bu dönüşüm tekniklerinde farklı frekansların sinüs ve cosinüs eğrileri ortagonal temel fonksiyonların bir grubunu oluşturur. Her dönüşüm katsayısı, dönüştürülmüş bir fonksiyonun iç çarpımı ve temel fonksiyonlardan biri tarafından belirlenir. Bir integral iç çarpımı ve sürekli temel fonksiyonlar diğer iki dönüşüm için uygun olduğu zaman, bir ayrık iç çarpım ve ayrık temel fonksiyonlar, ayrık Fourier dönüşümü için kullanılır. Her bir durumda ters dönüşüm, dönüşüm katsayıları tarafından belirlenen genişliğin temel fonksiyonlarının toplamından ibarettir. Bu toplam sürekli Fourier dönüşümleri için bir integraldir.

### 3.3.2 Dalgacık Dönüşümü Tipleri

Fourier dönüşümünde olduğu gibi dalgacık dönüşümünde de üç ayrı dönüşüm tekniği kullanılır. Sürekli dalgacık dönüşümü, dalgacık seri yayılımı ve ayrık dalgacık dönüşümü. Dalgacık dönüşümündeki durum biraz daha komplekstir. Bununla birlikte dalgacık temel fonksiyonu dikey olmayabilir.

Bir dalgacık temel fonksiyonu grubu, fonksiyonlar dikey olmasa bile dönüşümü destekleyebilir. Örneğin bir dalgacık seri yayılımı son derecede fazla katsayı ile bir bant limitli fonksiyonu temsil edebilir. Eğer katsayılar dizisi sonlu uzunluğa kısaltılırsa, orjinal fonksiyonun sadece yaklaşık bir şeklini yeniden oluşturabiliriz. Aynı şekilde ayrık bir dalgacık dönüşümü, tamamıyla yeniden oluşturma işlemi veya hatırı sayılır bir yaklaşık çözüm için, örnek noktalara sahip orjinal fonksiyondan daha çok katsayı gerektirir.

### 3.3.3 İşaret Sistemi ve Tanımlar

İleriki bölümlerde dalgacık dönüşümün konseptine açıklık getirmek için bazı tanımlar üzerinde çalışacağız.

Dalgacıklarda literatürün en önemli kısmına uymak için,  $j$  değerini bir integer gösterge olarak kullanacağız. Buna göre  $j = \sqrt{-1}$  olarak alınacaktır.

Fonksiyonlar sınıfı, fonksiyonun karesinin integre edilmesi ile bulunur. Bu sınıf  $L^2(R)$  ile gösterilir. Burada işaret sistemi  $f(x) \in L^2(R)$ , aşağıdaki eşitliği oluşturur.

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty \quad (3.4)$$

Dalgacık analizinde tek bir örnek, fonksiyon halinde genişletilip açılarak bir temel fonksiyonlar kümesi oluşturulur, buna  $\psi(x)$  temel dalgacık adı verilir. Bu, biraz titreşimli bir fonksiyondur, genellikle orijinde merkezlenir,  $|x| \rightarrow \infty$  iken hızla yok olur. Böylece  $\psi(x) \in L^2(R)$  gerçekleşir.

#### 4. SÜREKLİ DALGACIK DÖNÜŞÜMÜ

Süreklı dalgacık dönüşümü Grosman ve Morlet tarafından ortaya atılmıştır.

##### 4.1 Tek Boyutlu Sürekli Dalgacık Dönüşümü

Eğer bir  $\psi(x)$  Fourier spektrumunun gerçek değerli bir fonksiyonu ise,  $\psi(s)$  uygulanabilir kriterini karşılar.

$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\psi(s)|^2}{|s|} ds < \infty \quad (4.1)$$

$\psi(x)$  temel dalgacık olarak adlandırılır.

$$\psi(0) = 0 \Rightarrow \int_{-\infty}^{\infty} \psi(x) dx = 0 \quad (4.2)$$

Üstelik,  $\psi(\infty) = 0$  olduğu için, bir bant-geçirgen filtrenin transfer fonksiyonuna benzer, uygun bir dalgacık genişlik spektrumunu görebiliriz. Bir dalgacık temel fonksiyonları grubu,  $\{\psi_{a,b}(x)\}$ , temel fonksiyonlar dönüştürülerek ve ölçülerek oluşturulur.  $a > 0$  ve  $b$  'nin gerçek sayılar olduğu yerde  $\psi(x)$ ,

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad (4.3)$$

olur.  $a$  değişkeni, özel bir temel fonksiyonun genişliğini ifade eder.  $b$  ise, fonksiyonun  $x$  eksenı boyunca çevrilmiş durumunu belirtir.

Normalde temel dalgacık  $\psi(x)$ , orjinde merkezlenir, bu sebeble  $\psi_{a,b}(x)$ ,  $x = b$  'de merkezlenir. Şekil 4.1 , bir dalgacık örneğini gösterir. Bu parça aşağıdaki eşitlik ile verilir.

$$\psi(x) = \frac{2}{\sqrt{3}\sqrt{\pi}} (1-x^2)e^{-x^2/2} \quad (4.4)$$

$\psi(x)$  dalgacığına uygun  $f(x)$  'in sürekli dalgacık dönüşümü;

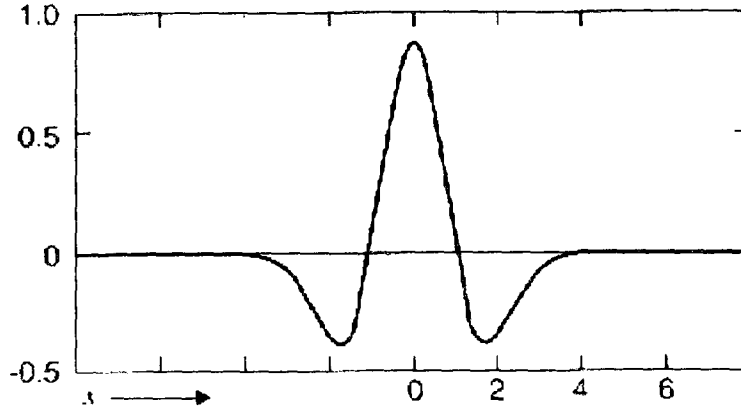
$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x) \psi_{a,b}(x) dx \quad (4.5)$$

olur. Dalgacık dönüşüm katsayıları ilk başta, temel fonksiyonların herbiri için dönüştürülmüş

fonksiyonun iç çarpımları olarak tekrar verilir.

Grossman ve Morlet, ters sürekli dalgacık dönüşümünü aşağıdaki gibi göstermişlerdir.

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_f(a,b) \psi_{a,b}(x) db \frac{da}{a^2} \quad (4.6)$$



Şekil 4.1 Bir dalgacık

(4.3) numaralı eşitliğin sağ tarafındaki ölçek çarpan, tam eşit dalgacık temel fonksiyonlarının normlarını kesinleştirir.

$$\left\| f\left(\frac{x-b}{a}\right) \right\| = \sqrt{\int_{-\infty}^{\infty} \left| f\left(\frac{x-b}{a}\right) \right|^2 dx} = \sqrt{a} \|f(x)\| \quad (4.7)$$

## 4.2 İki Boyutlu Sürekli Dalgacık Dönüşümü

Bir boyutlu bir  $f(x)$  fonksiyonunun sürekli dalgacık dönüşümü  $W_{a,b}$ , iki değişkenli bir fonksiyondur. Bilgi içeriğinde ve bilgi saklanması için gerekli miktarda önemli bir artışa sebep olur. Birden fazla değişkenli fonksiyonlar için bu dönüşüm, boyutu birer birer artırır.

$f(x,y)$  iki boyutlu bir fonksiyon ise, sürekli dalgacık dönüşümü,  $b_x$  ve  $b_y$  iki boyutluda çeviriyi açıkça belirttiği yerde

$$W_f(a,b_x,b_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \psi_{a,b_x,b_y}(x,y) dx dy \quad (4.8)$$

olarak alınır. İki boyutlu sürekli dalgacık dönüşümünün tersi de

$$f(x, y) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b_x, b_y) \psi_{a, b_x, b_y}(x, y) db_x db_y \frac{da}{a^3} \quad (4.9)$$

dönüşümü ile bulunur.. Burada

$$\psi_{a, b_x, b_y}(x, y) = \frac{1}{|a|} \psi\left(\frac{x - b_x}{a}, \frac{y - b_y}{b}\right) \quad (4.10)$$

alınır.  $\psi(x, y)$  iki boyutlu bir temel dalgacıdır. Aynı yaklaşım, iki değişkenliden fazla

fonksiyonlar için de geçerlidir.

### 4.3 Filtre Bankası Açıklaması

Şimdi üzerinde durulacak örnek, sürekli dalgacık dönüşümünü gözlemleme yollarından biridir.

$$\psi_a(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x}{a}\right) \quad (4.11)$$

Genel dalgacık temel fonksiyonu,  $a$  tarafından ölçekli ve  $a^{-1/2}$  tarafından normalize edilen temel dalgacıdır. Artan  $a$  ile daha geniş olan bir fonksiyonlar kümesini tanımlar.

$$\tilde{\psi}_a(x) = \psi_a^*(-x) = \frac{1}{\sqrt{a}} \psi^*\left(-\frac{x}{a}\right) \quad (4.12)$$

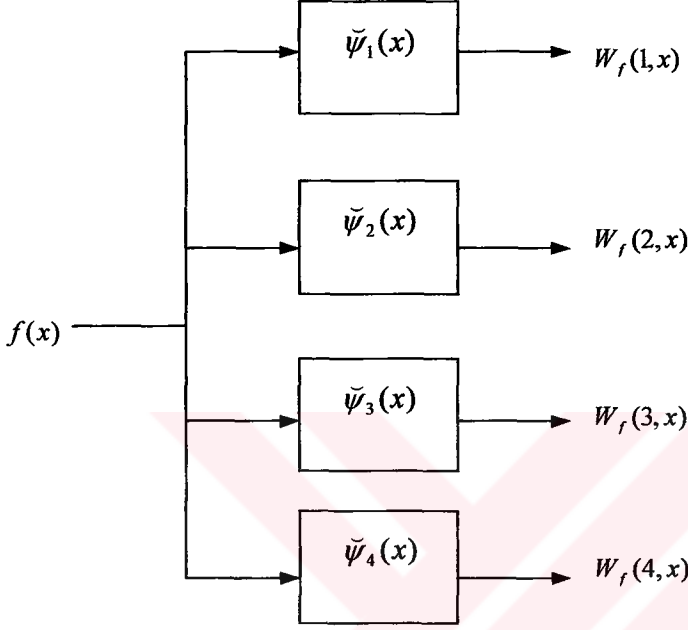
eşitliği ise sınırlı dalgacığın kompleks çiftini ifade eder. Eğer  $\psi(x)$  gerçektir ise yansıma ve birleşmenin etkisi yoktur. Buradan yola çıkarak şimdi sürekli dalgacık dönüşümünü yazabiliriz.

$$\psi_f(a, b) = \int_{-\infty}^{\infty} f(x) \tilde{\psi}_a(b - x) dx = f * \tilde{\psi}_a \quad (4.13)$$

Şekil 4.2 , integral dalgacık dönüşümünü,  $f(x)$  üzerinde lineer bir filtre bankası olarak gösterir.  $a$  'nın her değeri, farklı bir bant-geçirgen filtre tanımlar ve tüm bu filtrelerin çıktıları birlikte alınır, dalgacık dönüşümünden oluşur. Buradan (4.6) numaralı eşitlik, filtre çıktılarını içeren

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty [f * \tilde{\psi}_a](b) \psi_a(b * x) db \frac{da}{a^2} = \frac{1}{C_\psi} \int_0^\infty [f * \tilde{\psi}_a * \psi_a](x) \frac{da}{a^2} \quad (4.14)$$

şeklini alır. Herbiri filtrelendir, ölçülür,  $f(x)$ 'i yeniden yapılandırmak için birleştirilir. Bu Colderon'un bir ifadesidir. Grossmann ve Morlet'ten 20 yıl önce ifade edilmiştir.



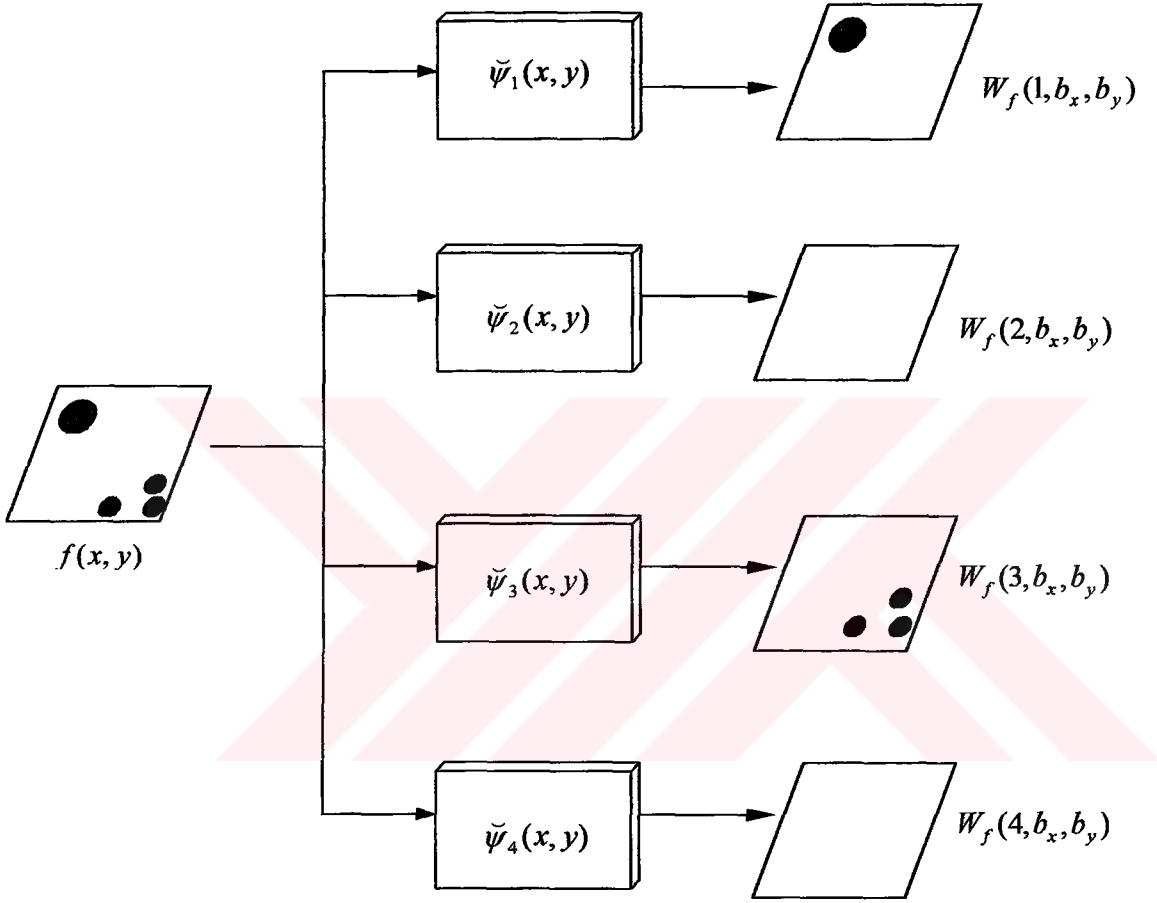
Şekil 4.2 Bir sinyalin integral dalgacık dönüşümünün filtre bankası benzerliği

#### 4.4 İki Boyutlu Filtreler

Şekil 4.3 , iki boyutluda filtre bankası teorisini açıklamaktadır. Burada her filtre  $\psi_a(x, y)$  iki boyutludur ve çıktısı görüntünün bir bant-geçirgen filtredir. Filtrelenmiş görüntü kümesi, dalgacık dönüşümünününden oluşmaktadır.

Bununla birlikte fazlalık dikkate alınmalıdır. Aslında  $\Psi(u, v)$ ,  $\psi(x, y)$ 'nin transfer fonksiyonu, orijinin dışında herhangi bir yerde sıfır değildir. Teorik olarak ters filtre kullanarak filtre çıktılarından gerçek görüntüyü çıkarmak mümkündür. Alternatif olarak, eğer görüntü sıfır olmayan en az bir  $\Psi_a(u, v)$  'den bir aralığa bant limitli ise,  $f(x, y)$  yalnızca filtre çıktısından tekrar oluşturulabilir. Sonuç, sadece sinyal ve görüntünün yeniden düzenlemesi hususunda, bütün bir algoritma içinde yer almayan integral dalgacık dönüşümünün potansiyel değeridir.

Bunu açıklamak için, Şekil 4.3'ü göz önünde bulunduralım. Örnek olarak, farklı boyutta dairesel objeler ve temel dalgacık, sadece yarıçapın dairesel objelerine karşılık seçilsin. Bu durumda, görüntü çıktısı kümesinin test edilmesi, objenin yerini gösterir. Dahası, her obje sadece belirli boyutuna tekabül eden özel görüntü çıktısı içinde görünür.



Şekil 4.3 Bir görüntünün integral dalgacık dönüşümünün filtre bank benzerliği

## 5. DALGACIK SERİ YAYILIMI

### 5.1 Dyadic Dalgacıklar

Dalgacık dönüşümünün ikinci tipi ilkinden biraz daha sınırlayıcıdır. Yeniden, bir temel dalgacık ölçülür ve temel fonksiyonlar kümesi olutmak için dönüştürülür. Burada ayrıca ölçme ve dönüştürme, gerçek sayılara tercihen integer sayılar tarafından belirtilir.



Şekil 5.1 Bir dalgacığın binary ölçekleri ve dyadic çevirileri

Bu ikinci tanımda, çalışmamızı temel fonksiyonları oluşturma aşamasında ikili düzenler ve temel dalgacığın dyadic çevirileri ile sınırlayacağız. Bir dyadic çeviri, ikili düzen çarpanının bir integer katı olan,  $k/2^j$  kadar bir değişimdir. İkili ölçmeler ve dyadic çeviriler Şekil 5.1'deki örnekle açıklanmaktadır.

### 5.2 Tanım

Bir  $\psi(x)$  fonksiyonu dikey bir dalgacıktır.  $-\infty < j, k < \infty$  integer değerleridir. Buradan dikey dalgacık;

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad (5.1)$$

olur,  $L^2(R)$  'nin dikey temelini oluşturur.  $j$  integer değeri genişliği belirler,  $k$  ise çeviriyi belirtir.

Dalgacık grubundan önce gelen dikey temeli oluşturur. Birincisi;

$$\langle \psi_{j,k}, \psi_{l,m} \rangle = \delta_{j,l} \delta_{k,m} \quad (5.2)$$

$l$  ve  $m$  integer değerler,  $\delta_{j,k}$ , Kronecker delta fonksiyonudur ve  $\langle \cdot, \cdot \rangle$  iç çarpımı gösterir.

İkincisi, eğer  $f(x) \in L^2(R)$  fonksiyonu aşağıdaki gibi yazılırsa;

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \psi_{j,k}(x) \quad (5.3)$$

olur, dönüşüm katsayılarının iç çarpım olarak verildiği eşitlikte transfer katsayıları;

$$c_{j,k} = \langle f(x), \psi_{j,k}(x) \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \psi(2^j x - k) dx \quad (5.4)$$

şeklinde gerçekleşir. (5.3) ve (5.4) numaralı eşitlikler,  $f(x)$ 'in bir dalgacık seri yayılımının  $\psi(x)$  dalgacığına bağlı olduğunu gösterir.

Burada sürekli bir fonksiyonun iki katlı bir düzenle gösterildiğini farzedelim. Genelde dönüşüm tekrar overcomplete'tir. Her iki yönde, genellikle fonksiyonlar sonsuzluğa doğru gittiği için, tam bir yeniden oluşturma işlemi tüm terimleri kapsamaktadır.

Eğer bir  $\psi(x)$  özellikle seçilmiş ise, seriler ciddi bir yaklaşık hata olmaksızın kısaltılabilir.

Eğer  $f(x)$  sonlu sürenin ise, temel dalgacık iyi oluşturulmuştur.  $|k|$  nedeniyle birçok katsayı önemsiz olacaktır. Aynı şekilde,  $|j|$  nedeniyle katsayılar çoğunlukla mümkün olduğu kadar küçük olacak, dalgacık temel fonksiyonu son derece geniş veya dar olacaktır.

### 5.3 Compact Dyadic Dalgacıklar

Eğer  $f(x)$ 'i sınırlarsak ve temel dalgacık  $[0,1]$  aralığında sıfır ise, ortonormal temel fonksiyonlar ailesi bir single  $n$  index değeri ile belirlenir. Buradan;

$$\psi_n(x) = 2^{j/2} \psi(2^j x - k) \quad (5.5)$$

elde edilir.

$$n = 2^j + k \quad j = 0, 1, \dots, \quad k = 0, 1, \dots, 2^j - 1 \quad (5.6)$$

Herhangi bir  $n$  için  $2^j \leq n$  gibi  $j$  en büyük integer değerdir ve  $k = n - 2^j$  'dir.

Ters dönüşüm,  $\psi_0(x) = 1$  kabul edilirse,

$$f(x) = \sum_{n=0}^{\infty} c_n \psi_n(x) \quad (5.7)$$

olur. Dönüşüm katsayıları iç çarpım olarak verilirse eşitlik

$$c_n = \langle f(x), \psi_n(x) \rangle = 2^{j/2} \int_{-\infty}^{\infty} f(x) \psi(2^j x - k) dx \quad (5.8)$$

şeklinde gerçekleşir. Burada sürekli bir fonksiyon, bir single sonsuz düzen ile gösterilir. İntegral dalgacık dönüşümünün çok büyük fazlalığı yoktur. Aslında bir veya birkaç  $\psi_n(x)$ ,  $f(x)$ 'e benzer ise, seriler birkaç terime oranla, fark edilir bir yaklaşık hata olmaksızın kısaltılabilir.

Bu durumda artık, ayrık dalgacık dönüşümünün temelini görebiliyoruz. Eğer  $f(i\Delta t)$ ,  $N$

noktada örneklenmiş bir ayrık fonksiyon ise ve  $\psi(x)$  compact dyadic bir dalgacık ise, (5.7) ve (5.8) numaralı eşitlikleri kullanarak, bir ayrık dalgacık dönüşümünü hesaplayabiliriz. Her iki eşitlik de  $N$  terimin toplamıdır. Haar dönüşümü buna örnek gösterilebilir.

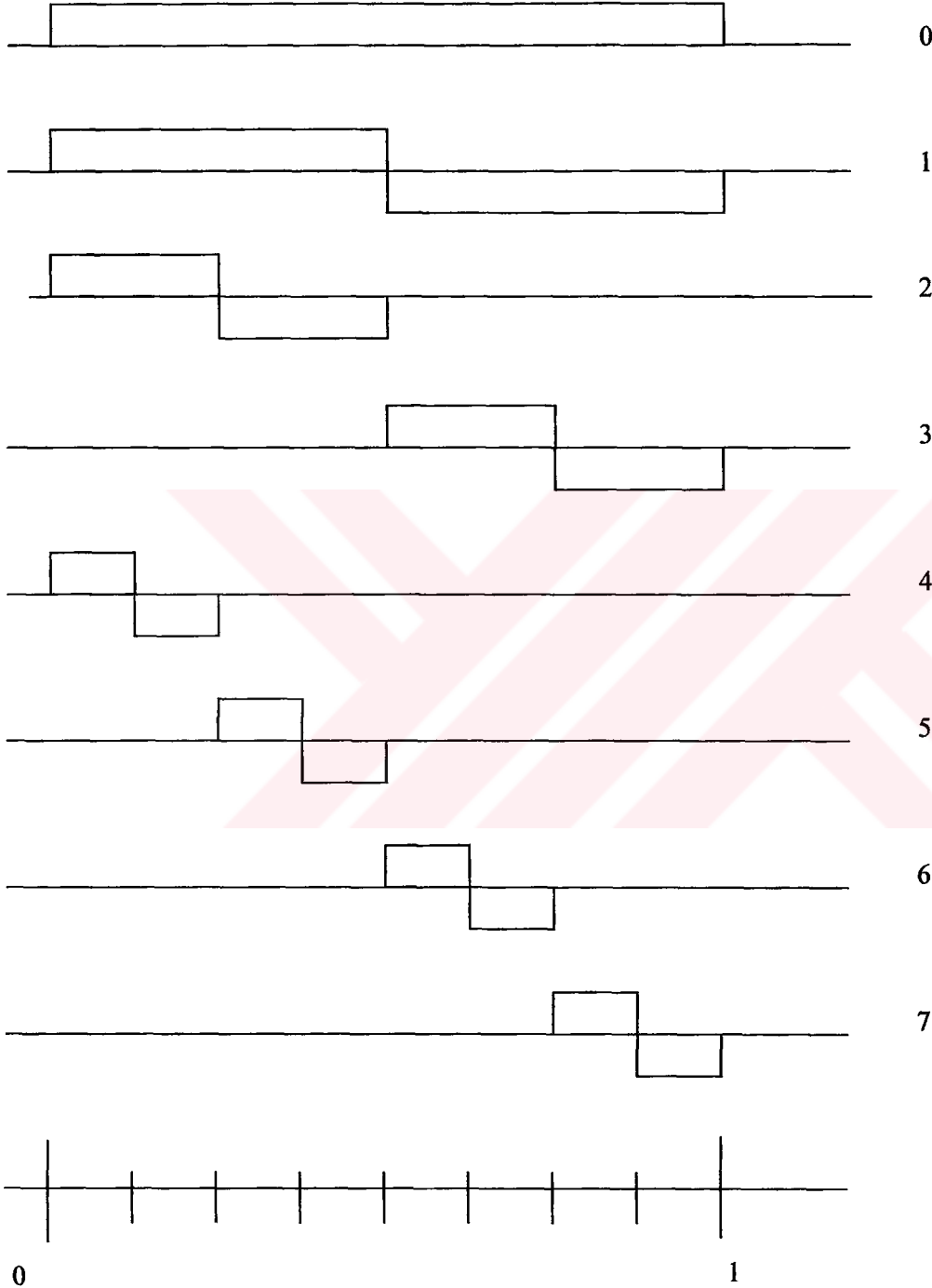
### 5.3.1 Örnek: Haar Dönüşümü

Haar dönüşümü, compact, dyadic, dikey bir dalgacık dönüşümü diye belirtebileceğimiz en eski örneklerden biridir. Diğer dönüşümlerden farklıdır. Dikdörtgen biçiminde ilginç bir sinyal çifti olan Haar fonksiyonu, compact support'u olan en basit ve en eski dikey dalgacıktır.

Temel dalgacık ikinin katları ile gittikçe küçültülür. Her bir daha küçük dalgacık, genişliğine eşit artışlar ile çevrilir, bu yüzden herhangi bir ölçekte tüm dalgacık grubu tamamen aralığı kapsar. Temel dalgacık ikinin katları ile ölçekli düştürüldüğü için, genişliği  $\sqrt{2}$ 'nin katları ile ölçekli arttırılır. Bunun sonucu dikey temel fonksiyonların bir grubudur. Bu temel fonksiyon indeksi, (5.6) numaralı eşitlikte tanımlanmıştır.

## 6. AYRIK DALGACIK DÖNÜŞÜMÜ

Ayrık dalgacık dönüşümü, birimsel dönüşümlere daha çok benzer. Görüntü sıkıştırma, görüntü işleme ve analizi için en yararlısı olmayı vaadeder. Uygun bir temel dalgacık elde etmek, bununla birlikte daha detaylı bir zemin elemanına ihtiyaç duyar.



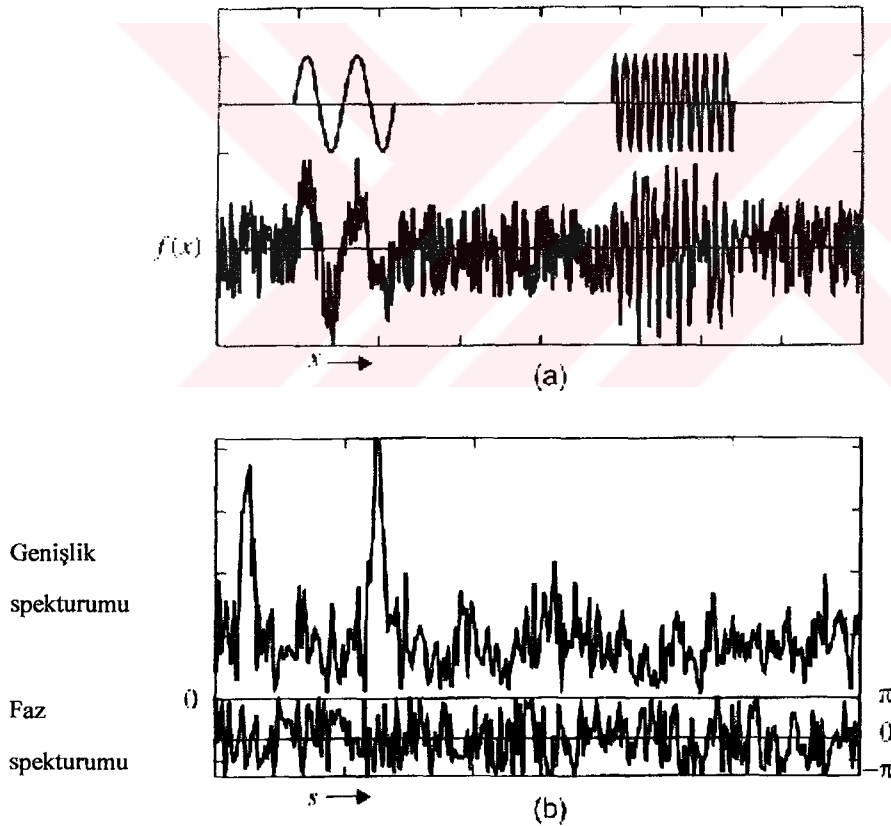
Şekil 6.1 Haar Dönüşüm Temel Fonksiyonları

Bu bölümde ilk olarak, ayrık dalgacık dönüşümünün gelişimini sağlayan üç teknik üzerinde çalışacağız. Bunlar (1) Filtre Bankası Teorisi, (2) Çoklu çözüm veya zaman-ölçü analizi, özellikle pramit gösterimini kullanarak, (3) Alt-bant kodlama.

### 6.1 Filtre Bankası Teorisi

Konuşma analizi ve akustik sinyal işleme alanlarında çalışanlar, bir sinyali farklı frekanslarda bileşenlere ayırmak için uzun bir süre, bant-geçirgen filtre bankası kavramını kullandılar. Gerçekten de, bu metod zaman-frekans analizi için bir aşamadır. Sinyalin bileşenleri ve boyutları salınımın frekansı ve oluşum zamanı olan iki boyutlu bir uzayda gösterilir. Burada, yaklaşımın temellerini yeniden gözden geçireceğiz.

Rasgele bir ses içine yerleştirilmiş iki ton patlamasına benzer bir sinyal üzerinde çalıştığımızı düşünelim (Şekil 6.2a). Bununla birlikte bu sinyali sayıları, frekansları ve ton patlaması durumlarını bulmak için analiz etmek istediğimizi düşünelim.



Şekil 6.2 İki ton patlamalı sinyal ve leke (a) üç bileşen

(b) genişlik ve faz spektrumu

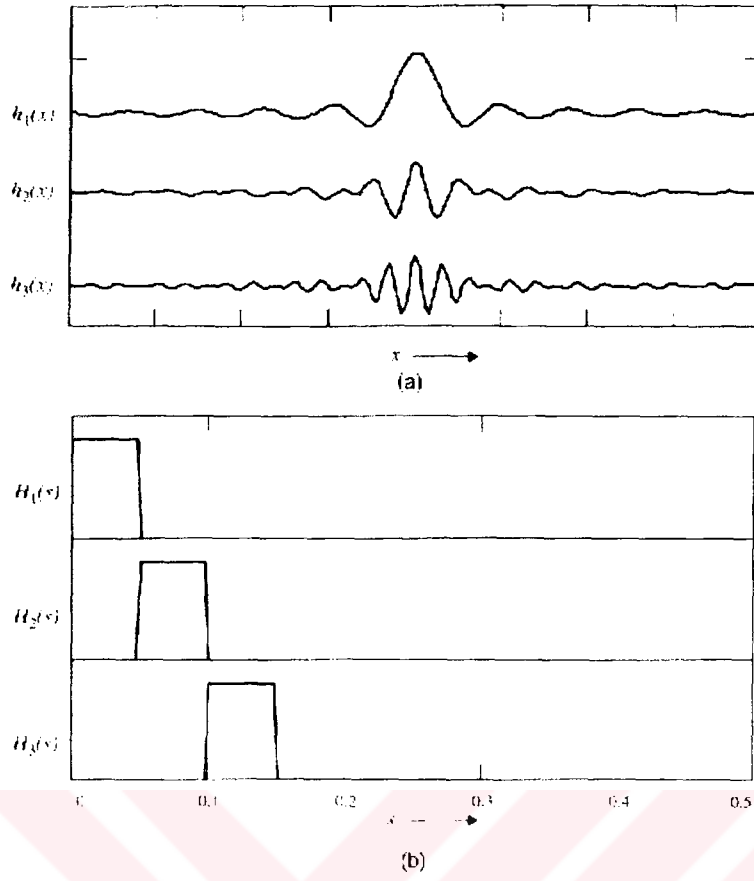
Fourier dönüşümü tabii ki sinyalin tüm içeriğini ifade edecektir, fakat, kolayca açıklamak için iyi bir yol değildir. Örneğin pozisyon bilgisi, faz spekturumunda zor bir yol ile kodlanmıştır. Genişlik spekturumu herbir geçici sinyal bileşeni nedeniyle ayrık tepeleri gösterebildiği zaman bu, spekturuma hakim olarak genişlik ve süre zarfında yeterince büyük olan bu bileşenlerin sadece geçici taramaları için güvenilirdir. Şekil 6.2b, iki tonlu patlama frekanslarında ayrık tepeleri gösterir. Sinyalin veya sesin ilgili olmayan diğer bileşenleri, basit bir frekans analizinin sinyali bileşenlerine ayrıştırmada yetersiz olduğu bir noktada spekturumu güçleştirir.

### 6.1.1 İdeal Bant-geçirgen Filtreler

$x$  frekans bileşenlerini, aralıklar grubuna ayırdığımızı düşünelim ve bu bölünmeyi ideal bant-geçirgen filtreleri tanımlamak için kullanalım. Bu durum Şekil 6.3b'de gösterilmiştir. Benzer impulse response'lar, şekil 6.3a'da görülmektedir. Şekil 6.4 ise bir bant-geçirgen filtre bankası uygulamasını gösterir. Giriş sinyali her band geçirgen filtreyi beslemektedir. Benzer çıktılar  $g_i(x)$  olarak alınır.  $H_i(s)$  oluşturulur, bu yüzden tüm frekanslar için çıktılar 1'e toplanır.

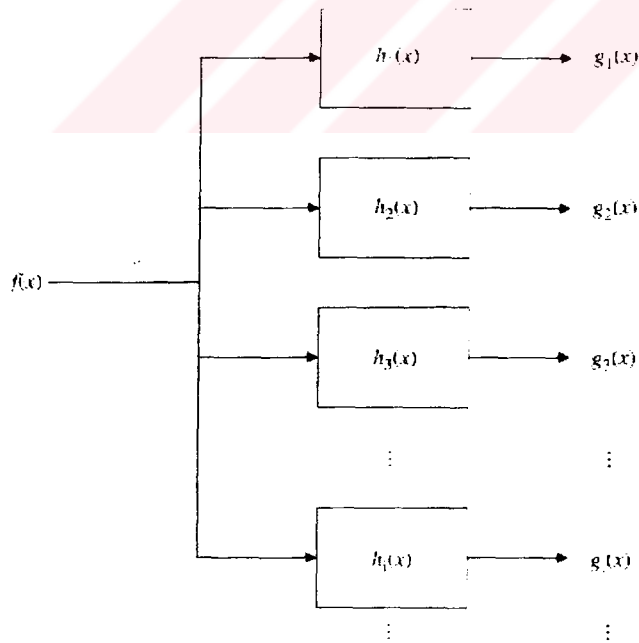
$$\sum_{i=1}^{\infty} H_i(s) = 1 \Rightarrow \sum_{i=1}^{\infty} g_i(x) = f(x) \quad (6.1)$$

Şekil 6.5 , Şekil 6.4'te gösterilen üç bant-geçirgen filtrenin çıktısını gösterir. Bu iki tonlu patlama sinyallerinin, ayrık filtrelerden çıktığını düşünelim. Üstelik, zaman eksenini boyunca konumları belli olsun. Böylece bir sinyali ayrıştırma ve bileşenleri tanımlama yaklaşımına sahip oluruz.

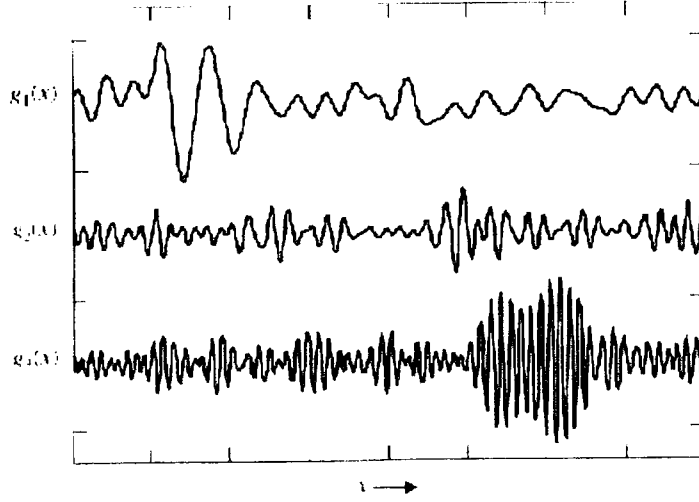


Şekil 6.3 Frekans axislerini bölerek bant-geçirgen filtreler oluşturma

(a) impulse response (b) transfer fonksiyonları



Şekil 6.4 Bir bant-geçirgen filtre bankası uygulaması



Şekil 6.5 Bant-geçirgen filtre çıktıları

Bant-geçirgen filtre çıktılarının herbiri bir bükülmeden ibarettir.

$$g_i(x) = \int_{-\infty}^{\infty} f(t)h_i(x-t)dt \quad (6.2)$$

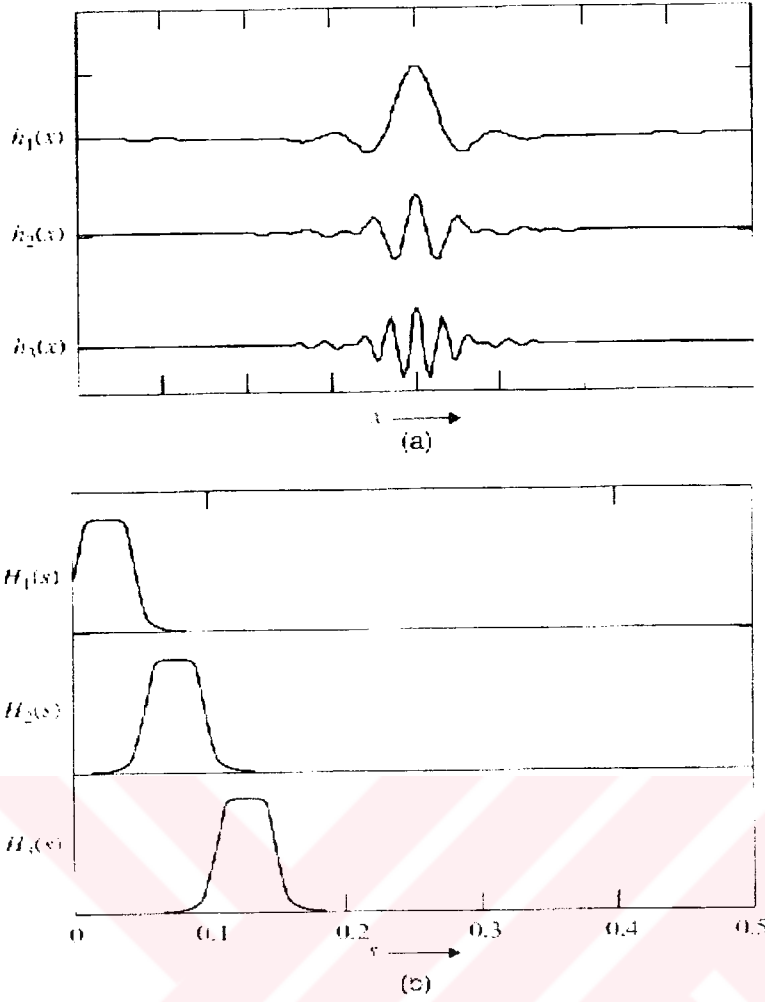
$H_i(s)$  reel ve çift olduğu için,  $h_i(x)$ 'de reel ve çift olacaktır. Bükülme integralindeki yansımanın etkisi yoktur, buradan filtre çıktıları;

$$g_i(x) = \int_{-\infty}^{\infty} f(t)h_i(t-x)dt = \langle f(t), h_i(t-x) \rangle \quad (6.3)$$

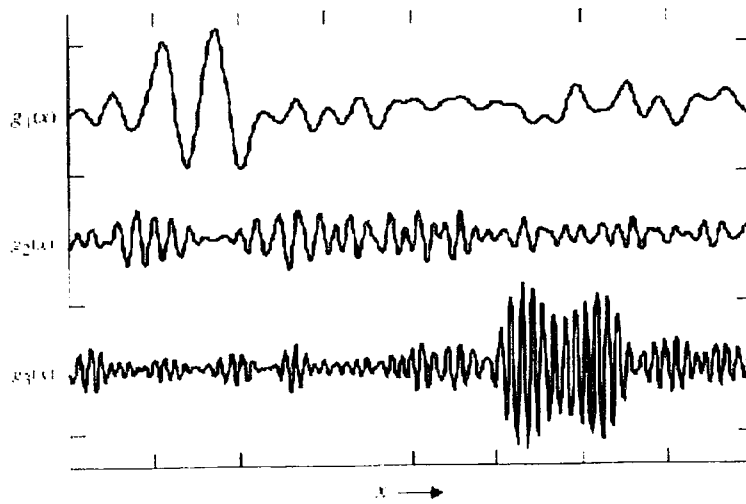
olarak yazılır. Bu nedenle  $g_i(x)$  üzerindeki her nokta,  $x$  lokasyonu tarafından değiştirilen  $h_i(t)$ 'nin bir versiyonu ile  $f(t)$ 'nin bir iç çarpımıdır. Burada  $\{g_i(x)\}$ 'i,  $\{h_i(x)\}$ 'in dalgacıklar kümesi olduğu yerde, dalgacık dönüşüm katsayılar kümesi olarak da görebiliriz. Bununla birlikte  $\{g_i(x)\}$ ,  $f(x)$ 'i tamamıyla tekrar oluşturmak için yeterlidir.

### 6.1.2 Düzgün Bant-geçirgen Filtreler

6.3a numaralı şekildeki  $h_i(x)$  fonksiyonları, iyi dalgacık temel fonksiyonlarının sahip olduğu karakteristik özelliklerinin önemli birinden yoksundur. İyi dalgacık temel fonksiyonlarının aksine, iyi konumlandırılmamışlardır. Merkez bölgelerinden çabuk kaybolmazlar. Bu da  $h_i(x-x_0)$  'in  $x_0$ 'dan ayırık yerleştirilmiş güçlü bileşenlere karşılık geldiği anlamına gelir.  $h_i(x)$ 'in arzu edilmeyen genişliğine artış veren  $H_i(s)$ 'in keskin kenarlarıdır.



Şekil 6.6 Düzgün bant-geçirgen filtreler (a) impulse response  
(b) transfer fonksiyonları



Şekil 6.7 Düzgün bant-geçirgen filtre bankası çıktıları

Daha düzgün kenarları olan  $H_i(s)$  fonksiyonları yaratmak,  $h_i(x)$ 'in genişliğini küçültecektir.  $H_i(s)$  hala birleşmeyi toplamak zorunda olacağı için, bant-geçirgen transfer fonksiyonlarının sonucu, kenarlarında üst üste gelecektir. Bu, Şekil 6.6'da görülebilir. Burada, geçirgen bant kenarları, cosinüsün yükselmiş yarım bir dairesidir.

Şekil 6.7 , giriş ve düzgün bant-geçirgen filtreler olarak, Şekil 6.2a'daki sinyalin filtre bankası çıktılarını gösterir.

## 6.2 Çoklu Çözüm Analizi

Dalgacık analizinden önce gelen bir çok gelişme genellikle çoklu çözüm analiz adı verilen alanda gerçekleşmiştir. Bu gelişmeler, Fourier dönüşüm sınırlamaları ile mücadele için tasarlanmıştır. Şimdi bunu, modern dalgacık analizi için bir temel kaynak olarak özetleyeceğiz.

Filtre bankası teorisi, müzikteki notalar ve ton patlamaları gibi titreşimli bileşenlerden ibaret olan sinyallerin uygun ortalamalarını gösterir. Bu bileşenler süreleri zarfında salınımın birtakım turlarını kapsamaktadır. Bununla birlikte görüntü analizinde, lokalize olmuş bileşenler tamamıyla titreşimli değildir. Bu kapsamdakiler sadece bir turu kapsar.

Görüntüdeki objeler, farklı boyutlarda var olmak için gözlenirler. Örneğin bir kenar, siyahtan beyaza keskin bir geçiş de olabilir, önemli bir mesafe üzerinde azar azar da oluşabilir. Genelde bir çoklu çözüm, görüntü tasvirine yaklaşıp veya analiz bu yaklaşımı kullanır.

Haritacılık bu yaklaşımı örneklerle açıklamaktadır. Haritalar çoğunlukla farklı ölçeklerde hazırlanır. Bir haritanın ölçeği, harita üzerindeki gösterilen asıl arazi parçasının boyut oranıdır. Büyük ölçeklerde, dünya üzerindeki kara parçaları ve okyanuslar gibi temel bileşenler görünür, şehir sokakları gibi detayların harita üzerindeki yoğunluğu azalır durumdadır. Daha küçük ölçekli haritalarda ise, detaylar görülebilir, büyük bileşenler harita üzerinde yer almazlar. Böylece, ayrı bir lokasyonda bir yer araştırmak için farklı ölçeklerde harita çizimlerine ihtiyaç duyulur.

Dalgacık dönüşümleri bu çoklu çözüm yollar ile gelişmiştir. Zaman frekans analizinde bir sinyal iki boyutlu olarak gösterilir. Fakat, burada dikey eksen, frekanstan daha çok kullanılan bir ölçektir. Ölçek, bir temel fonksiyonlar kümesi oluşturmak için temel dalgacığı genişleterek ve yapılandırarak elde edilir.

Temel dalgacık  $\psi(x)$ , bir temel fonksiyonlar kümesi oluşturmak için  $\psi(x/a)$  olarak ölçeklenir. Büyük ölçekte, genişletilmiş temel fonksiyonlar, büyük bileşenleri inceler, küçük olan ise detay üzerinde çalışır.

### 6.2.1 Piramit Algoritmalar

1024 x 1024 pixel dijital bir görüntü, 10 adet ortalama 2 x 2 pixel bloklar halinde ek görüntüler, her seferinde saniyede bir atan pixel satır ve sütunları oluşturduğumuzu düşünelim. Görüntüleri 512 x 512, 256 x 256 gibi birer birer küçültelim. Eğer kenar tarama yapsaydık, gerçek görüntü içinde küçük kenarlar bulurduk.

Haar dönüşümü bu yaklaşımı hemen hemen bir yüzyıl önceden beri göstermektedir. Temel görüntülerinde, incelenen görüntünün bütünü farklı ölçeklerin kenar tarayıcıları ile görürüz. İkili genişleme prensibi yeterince açıktır.

Küçük ve büyük, 1024 x 1024 pixellik gerçek görüntüde görünen ve bunları yerleştirmek için talep edilen yoğunluk değişikliği olmayan bütün kenarlar araştırılmaya çalışılır. Buradaki problem geniş kenarların uygun komşu kenarlar bulma işlemi için zor olmasıdır. Büyük kenarları bulmak, operatörlerin ölçeğini arttırabilir, fakat görüntü ölçeğini düşürmek daha yararlıdır. Yüksek yoğunluklu büyük kenarlı büyük bir operatör kullanmak, aşırı zara vericidir.

Bazı çoklu çözüm analiz formları yıllardır farklı isimler ile telafüz edilmiştir. Yalnızca son yıllarda, çoklu çözüm analizi ve filtre bankası yaklaşımları arasındaki benzerlik farkedilmiş, bu da dalgacık dönüşümleri başlığı altında birleştirilmiştir.

### 6.2.2 Laplace Piramit Kodlama

Burt ve Adelson, temeli Gauss fonksiyonuna bağlı bir piramit kodlama sistemi üzerinde çalıştılar. Görüntü, bir Gauss impulse response ile alçak-geçirgen filtrelendir ve sonuç gerçek görüntüden çıkartılır. Görüntüdeki yüksek frekanslı detay, farklı bir görüntü içinde saklanır. Alçak-geçirgen filtrelenmiş görüntü, kayıpsız olarak örneklenebilir. Bu işlemler aşağıdaki gibi örneklenebilir.

$f_0(i, j)$  gerçek görüntü,  $g(i, j)$  bir Gauss formunda alçak-geçirgen filtre impulse response'u olsun. Kodlamanın her aşamasında görüntü, yarım çözünürlüklü düşük frekanslı ve tam çözünürlüklü yüksek frekanslı bileşenlere ayrılır. Bunlar  $f_1(i, j)$  ve  $h_1(i, j)$  olarak belirtilir.

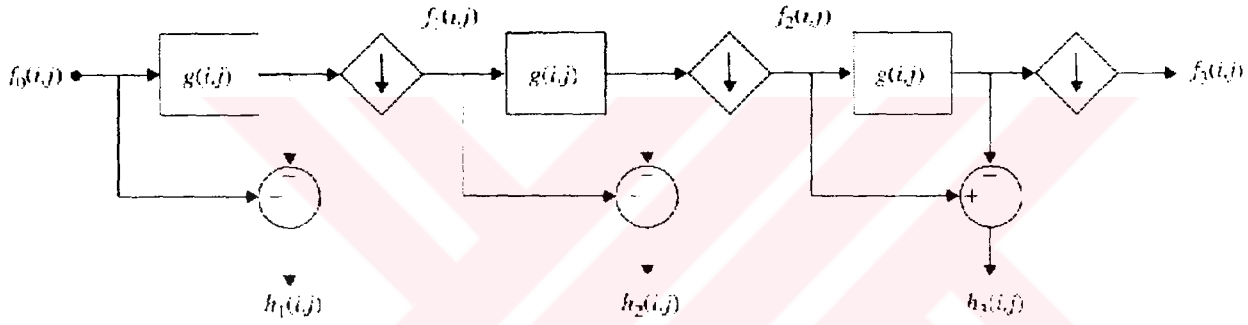
$$f_1(i, j) = [f_0 * g](2i, 2j)$$

$$\text{ve } h_1(i, j) = f_0(i, j) - [f_0 * g](i, j)$$

Bu işlem örnek görüntü üzerinde birçok kez tekrarlanır.  $n$  iterasyon sonra görüntü,  $N = 2^n$  olduğu yerde,  $f_n(i, j)$  tek bir noktadır. Kodlanmış görüntü piramiti,  $h_k(i, j)$  ve sonuç düşük frekanslı görüntü  $f_n(i, j)$ 'yi içermektedir. Bu işlem Şekil 6.8'de gösterilmektedir.

Görüntü kodlama ters açıdan da yapılır. Örnekleme, örnek noktalar arasında sıfırlama işlemidir. Her bir alt örneklenmiş görüntü  $f_k(i, j)$ , son bileşenle başlayan  $f_n(i, j)$ ,  $g(i, j)$  ile enterpole edilir. Sonuç bir sonraki görüntü  $f_{k-1}(i, j)$ 'e eklenir. Bu işlem, son görüntüye kadar tekrar edilir. Böyle gerçek görüntü kayıpsız bir şekilde elde edilir.

Her bir  $h_k(i, j)$ , çift genişlikli ve single Gauss ile bir single görüntü tarafından elde edilmiş iki görüntünün farkıdır.



Şekil 6.8 Laplace piramit kodlama şeması

Laplace piramit kodlama, sistemin pixel sayısını %33 arttırmasına rağmen, yine de görüntü sıkıştırmanın önemli bir derecesini başarıyla sonuçlandırır.  $h_k(i, j)$  görüntüler büyük miktarda korelasyon ve dinamik dağılımı küçültürler. Bununla birlikte, Laplace piramitinin oluşum şekli, daha sonra ayrık dalgacık dönüşümüne temel olacaktır.

### 6.3 Alt-bant Kodlama

Ayrık dalgacık dönüşümüne yol gösteren başka bir zemin olarak, bir zaman frekans analiz tekniği olan Alt-bant Kodlama'yı tanımlayalım. Başlangıçta alt-bant kodlama bir sinyali dar bantlı bileşenlere ayırıştırma üzerinde uygulanmıştır. Bu yol ile sinyali kayıpsız olarak tekrar elde etmek mümkündür.

$f(t)$  bant limitli bir sinyal olsun.

$$\mathfrak{F}\{f(t)\} = F(s) = 0 \quad , \quad |s| \geq s_{\max} \quad (6.5)$$

Sinyali uygun bir aralıkta örnekleyebiliriz.

$$f(i\Delta t) \quad i = 0, 1, \dots, N-1 \quad s_{\max} \leq |s| = \frac{1}{2\Delta t} \quad (6.6)$$

$s_N$  Nyquist frekanstır.

Frekans eksenlerini birbirinden bağımsız alt aralıklara bölerek analiz işlemine başlarız. Herhangi bir alt aralık uzunluğu kullanılabildiği zaman, Şekil 6.9b'de gösterildiği gibi  $s_N/2$ 'yi seçeriz. Burada  $F(s)$ ,  $2s_N$  dilimi ile periyodiktir.

### 6.3.1 Daha Düşük Yarı-bant

Şekil 6.9b , ideal bir yarı-bant alçak-geçirgen filreyi gösterir. Bu  $h_0(i\Delta t)$  ile gösterilmektedir. Frekans bantı  $[-s_N/2, s_N/2]$  aralığındadır. Impulse response ve  $h_0$  transfer fonksiyonu;

$$h_0(t) = \sin c\left(\pi \frac{t}{2\Delta t}\right) \text{ ve } H_0(s) = \pi \left(\frac{S}{S_N}\right) \quad (6.7)$$

dikdörtgen ton

$$\pi(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \quad (6.8)$$

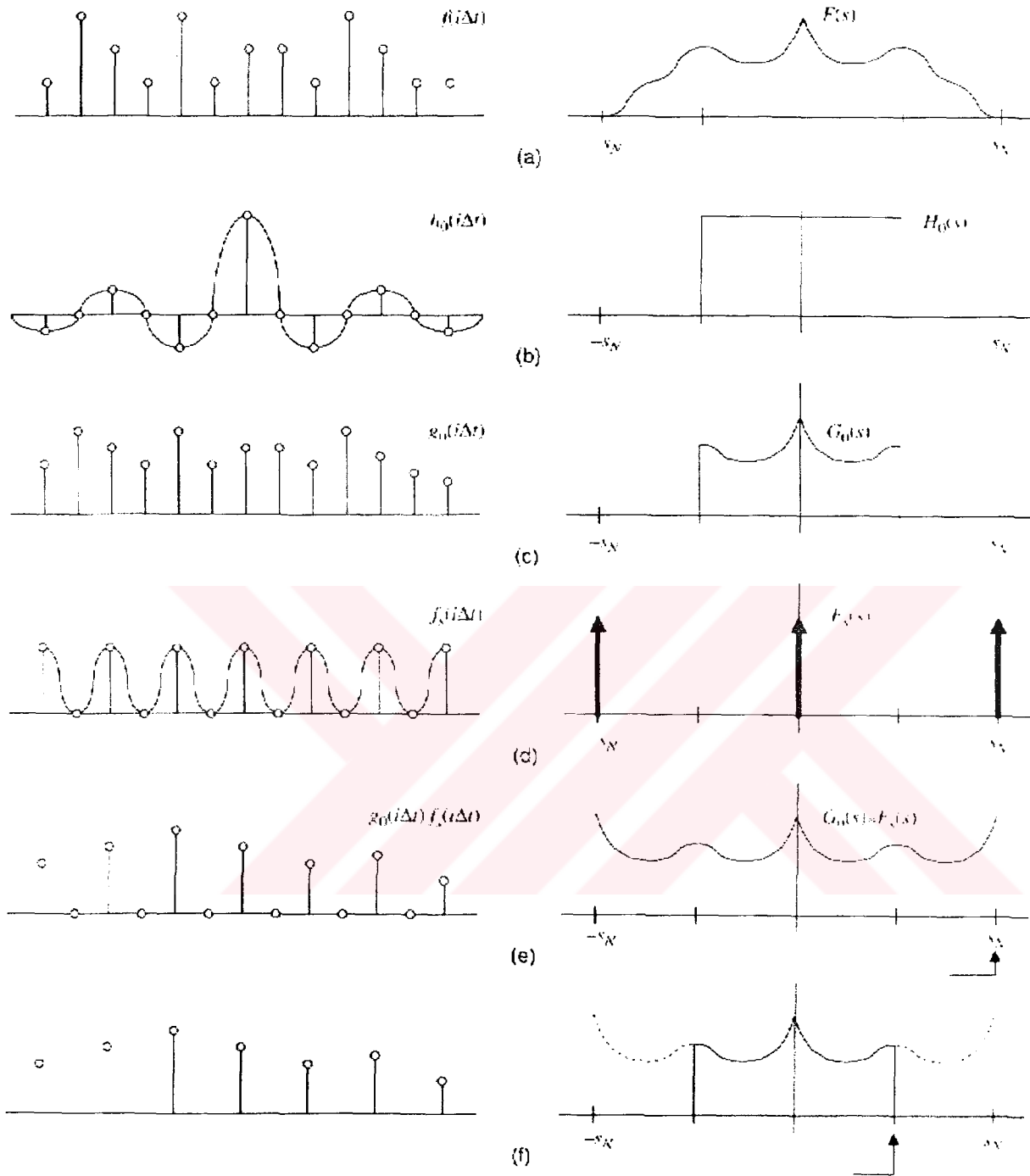
ve

$$\sin c(x) = \frac{\sin(x)}{x} \quad (6.9)$$

şeklinde gösterilir.

Bu filtre  $f(i\Delta t)$ 'ye uygulanır ve  $g_0(i\Delta t)$ ,  $s = s_N/2$ 'de bant limitlidir. Bu  $f(i\Delta t)$ 'nin düşük yoğunluklu bir versiyonudur.  $f(i\Delta t)$ 'nin temel şekline sahiptir, fakat kayıpları vardır.  $g_0(i\Delta t)$ ,  $s_N/2$ 'nin üzerinde bir enerjiye sahip olmadığı için en azından  $2\Delta t$  kadar büyük örnek aralık ile modellenebilir. Aslında, herbir modeli ortadan kaldırabiliriz. Bu

işlem alt örnekleme ve kesirleme olarak adlandırılır.



Şekil 6.9 Alt-bant kodlama, daha düşük yarı-bant ile

Bir subsampling fonksiyon

$$f_s(i\Delta t) = \frac{1}{2} [1 + \cos(2\pi s_N i\Delta t)] \quad (6.10)$$

ve spektrumu

$$F_s(s) = \frac{1}{2} [\delta(s) + \delta(s - s_N) + \delta(s + s_N)] \quad (6.11)$$

eşitlikleri Şekil 6.9'de belirtilmiştir.

$g_0(i\Delta t)$  sinyali,  $f_s(i\Delta t)$  kadar çoğaltırsak, spektrumunu  $F_s(s)$  ile katlandırabiliriz. Sonuç, periyodu  $2s_N$ 'den  $s_N$ 'e indirilerek simetrik bir spektrum oluşturmaktır (Şekil 6.9e). Genişlik yarıya indirilir ve aşağıdaki gibi yazılır.

$$F_s(s) * G_0(s) = \frac{1}{2} G_0(s) + \frac{1}{2} G_0(s + s_N) + \frac{1}{2} G_0(s - s_N) \quad (6.12)$$

Açıkçası, şu andan itibaren bilgi kaybı olmasızın tek sayı model noktalarını işlemekten çıkarabiliriz (6.9f). Bu işlem kıvrımlı frekansı  $s_N/2$ 'ye indirir.

Ayrık spektrumunu hesaplayarak,  $G_0(s)$ 'yi yeniden oluşturmayı, bunu  $s_N/2$ 'den  $s_N$ 'e kadar doldurmayı ve  $g_0(i\Delta t)$ 'yi elde etmek için  $G_0(s)$ 'nin ters ayrık Fourier Dönüşümünü alarak;  $g_0(i\Delta t)$ 'yi alt örneklenmiş bir sinyalden elde edebileceğimizi düşünelim (Şekil 6.9f). Bu metod tercih edilmediği zaman, alt modellemenin kayıpsız bilgi yarattığı kanıtlanmaya çalışılır.

$g_0(i\Delta t)$ 'yi elde etmenin daha kolay bir yolu, Şekil 6.9'da da görülmektedir. İlk önce kodlanmış alçak bit sinyali modelleyeceğiz. Sonra bu sinyali ideal yarı-bant filtresi  $2h_0(i\Delta t)$  ile filtreleyeceğiz. Bu, spektrumu yeniden oluşturacaktır.

Frekans alanında aşağıdaki eşitlik yazılır.

$$F_s(s) * G_0(s) \times H_0(s) = \left[ \frac{1}{2} G_0(s) + \frac{1}{2} G_0(s + s_N) + \frac{1}{2} G_0(s - s_N) \right] \times \pi \left( \frac{s}{s_N} \right) = \frac{1}{2} G_0(s) \quad (6.13)$$

### 6.3.2 Daha Yüksek Yarı-bant

$f(i\Delta t)$ 'nin daha yüksek yarı-bantını göz önünde bulundurursak, enerjiyi burada ideal bir yarı-bant bant-geçirgen filtre ile izole edebiliriz. Bu filtrenin impulse response'u ve transfer fonksiyonu, bahsedilen sıraya göre;

$$h_1(t) = \delta(t) - \sin c\left(\pi \frac{1}{2\Delta t}\right) \text{ ve } H_1(s) = 1 - \pi\left(\frac{s}{s_N}\right) \quad (6.14)$$

olarak yazılabilir. Burada  $\pi(x)$ , (6.8) numaralı eşitlikten elde edilir.

Filtreler sadece daha yüksek yarı-bant içinde sıfır olmayan spektrumun bir sinyalini,  $g_1(i\Delta t)$ 'yi üretir. Bu sinyal Şekil 6.9b'deki alçak-geçirgen filtre ile  $f(i\Delta t)$ 'den elimine edilen, tamamıyla yüksek frekanslı bilgi içerir. Bu sonuçla,  $g_0(i\Delta t)$  ve  $g_1(i\Delta t)$ , beraber alınır, gerçek sinyal  $f(i\Delta t)$ 'yi gösteren tüm bilgiyi içerir. Buradan

$$f(i\Delta t) = g_0(i\Delta t) + g_1(i\Delta t) = f(i\Delta t) * h_0(i\Delta t) + f(i\Delta t) * h_1(i\Delta t) \quad (6.15)$$

eşitliği göz önünde bulundurulursa

$$H_0(s) + H_1(s) = 1 \quad (6.16)$$

olduğu ortaya çıkar. Şekil 6.10d,  $f_s(i\Delta t)$ 'nin analizde nasıl kullanıldığını göstermektedir.  $g_1(i\Delta t)$ ,  $f_s(i\Delta t)$  tarafından alt modellenirse, spektrumu  $F_s(s)$  ile oluşturulur. Bu  $[-s_N/2, s_N/2]$  aralığını spektrumunun bir kopyası ile doldurur ve spektrumu Şekil 6.10e'de görüldüğü hale dönüştürür. Eşitlik aşağıdaki gibi olur.

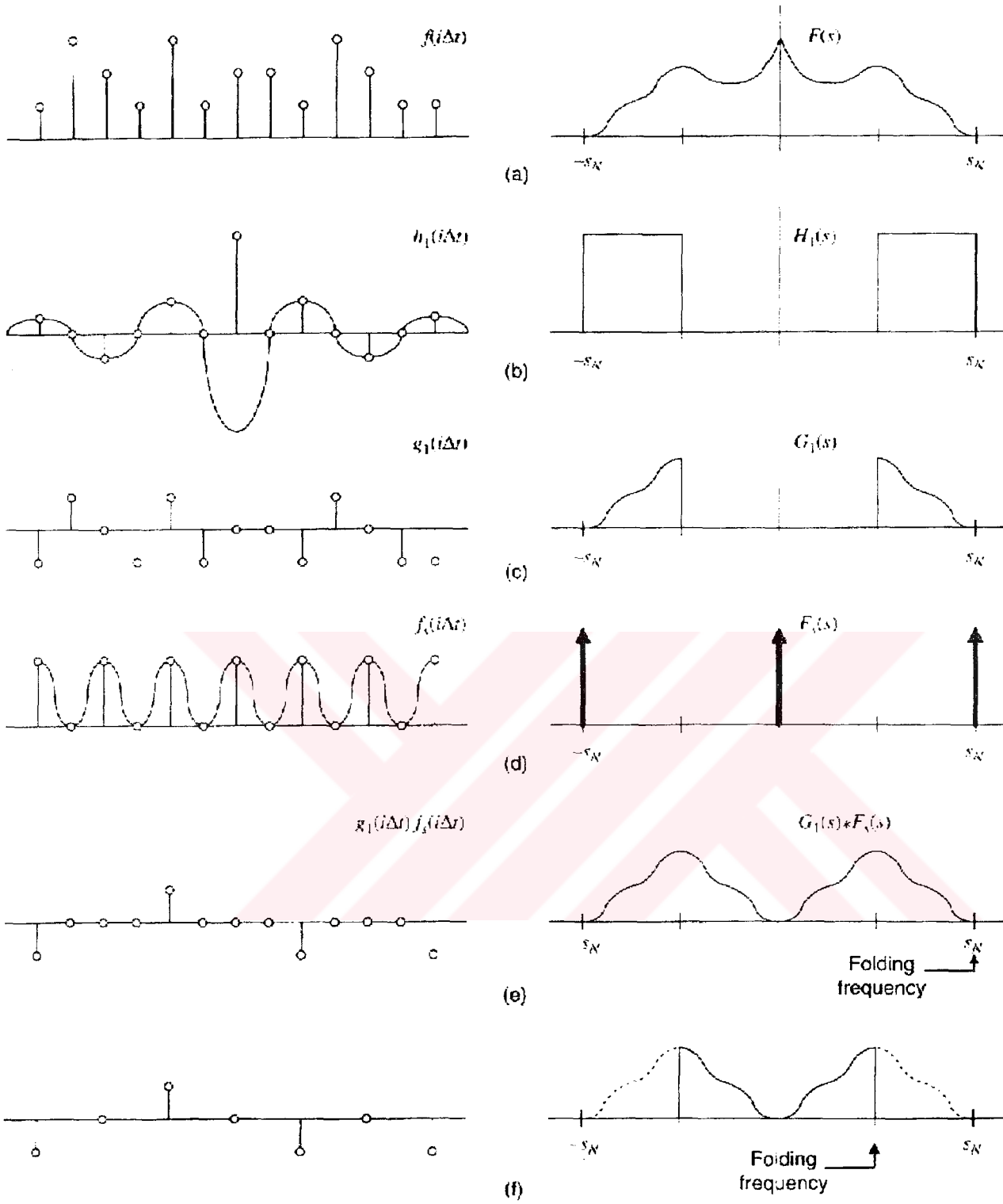
$$F_s(s) * G_1(s) = \frac{1}{2}G_1(s) + \frac{1}{2}G_1(s + s_N) + \frac{1}{2}G_1(s - s_N) \quad (6.17)$$

Bu spectrum  $s_N/2$ 'lik bir periyoda sahiptir. Böylece daha düşük yarı-bant olarak sınırlandırılan başka bir sinyale sahip oluruz.

Bu işlem iki  $N/2$  sinyali içine kodlanmış,  $N$  nokta sinyal bırakır.  $G_1(s)$ 'yi elde edebiliriz.

$$F_s(s) * G_1(s) \times H_1(s) = \left[ \frac{1}{2}G_1(s) + \frac{1}{2}G_1(s + s_N) + \frac{1}{2}G_1(s - s_N) \right] \times \left[ 1 - \pi\left(\frac{s}{s_N}\right) \right] = \frac{1}{2}G_1(s) \quad (6.18)$$

Bu sonuçla, iki kanallı alt-bant kodlamada, iki alt modellenmiş ayrık filtre çıktısının süreleri içinde sinyalin tersi alınabilir bir gösterimini elde etmiş oluruz.



Şekil 6.10 Alt-bant kodlama, daha yüksek yarı-bant ile

### 6.3.3 İki Bantlı Alt-bant Kodlama ve Kodu Çözme

İki kanallı alt-bant kodlama, sadece  $f(i\Delta t)$ 'yi,  $h_0(i\Delta t)$  ve  $h_1(i\Delta t)$  ile filtrelemede kullanır. Filtre çıktıları aşağıdaki gibi ifade edilir.

$$g_0(k\Delta t) = \sum_i f(i\Delta t) h_0((-i + 2k)\Delta t) \quad (6.19)$$

ve

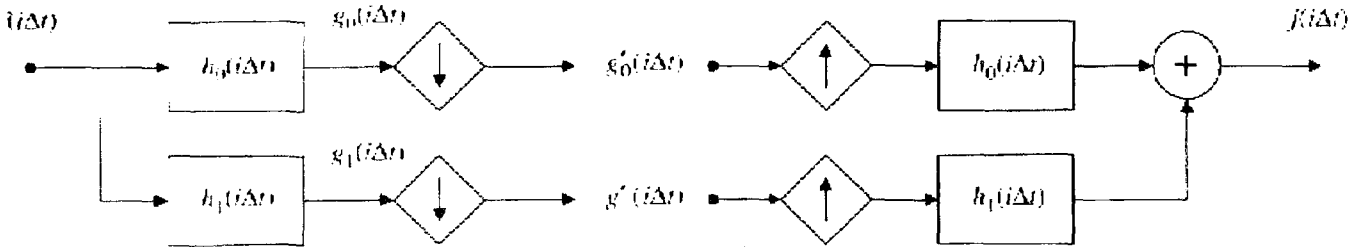
$$g_1(k\Delta t) = \sum_i f(i\Delta t) h_1((-i + 2k)\Delta t) \quad (6.20)$$

Yeniden oluşturma işlemi, daha düşük ve daha yüksek alt-bant sinyalleri tarafından üst modellenerek sonuçlandırılır.  $2h_0(i\Delta t)$  ve  $2h_1(i\Delta t)$  ile iç değerlendirme yapılır ve sırasıyla bunlara eklenir.

$$f(i\Delta t) = 2 \sum_k [g_0(k\Delta t) h_0((-i + 2k)\Delta t) + g_1(k\Delta t) h_1((-i + 2k)\Delta t)] \quad (6.21)$$

eşitliği Şekil 6.11'de örnekle açıklanmıştır. Kodlama ve kodu çözme işlemi  $f(i\Delta t)$ 'yi, birer defa  $h_0(i\Delta t)$  ve  $h_1(i\Delta t)$  ile olmak üzere, iki kere filtrelemeyi gerektirdiğinden, orta frekans noktası  $s = s_N/2$ 'de ufak bir problem vardır. Bu noktalarda  $H_0(s_N/2) = 1/2$  ve  $H_1(s_N/2) = 1/2$  eşitlikleri gerçekleşir. Bu problem (6.8) numaralı eşitlikte

$\pi(\pm \frac{1}{2}) = \sqrt{\frac{1}{2}}$  alınması ile giderilebilir.



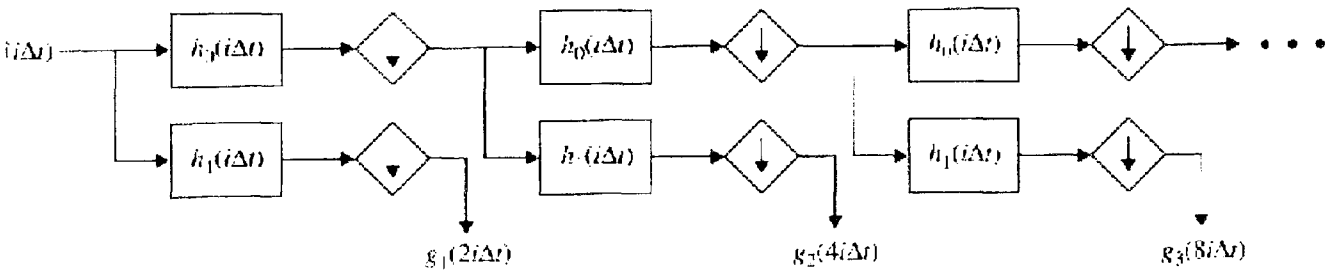
Şekil 6.11 İki bantlı alt-bant kodlama ve yeniden oluşturma

#### 6.4 Hızlı Dalgacık Dönüşüm Algoritması

Mallat, iç çarpımların tam bir kümesini hesaplamaktan daha etkili bir ayrık dalgacık dönüşüm algoritması tanımlamıştır. Bu algoritma, ardışık hareket içinde iki bantlı alt-bant kodlama işlemini gerçekleştirir ve temelden dalgacık dönüşümünü oluşturur.

Alt-bant kodlamanın ilk aşamasından sonra, daha düşük alt-bant sinyali,  $g_0(i\Delta t)$ , bir kez daha yarı-bant alt-bant kodlama işlemine maruz kalır. Bu bize  $[0, s_N]$  aralığının birinci ve ikinci çeyreklerine tekabül eden,  $N/2$  noktası daha yüksek yarı-bant sinyalini ve iki  $N/4$  noktası alt-bant sinyalini verir.

Herbir aşamada daha yüksek bir yarı-bant sinyale sahip olarak ve farklı düşük yarı-bant sinyali kodlanarak, bu işlem devam ettirilir, ta ki bir noktalı düşük sinyal elde edilinceye kadar. Dönüşüm katsayıları bu durumda alçak-bant noktası ve alt-bant kodlama yüksek yarı-bant sinyallerinin toplamı olacaktır. Şekil 6.12'de gösterilmektedir. İlk  $N/2$  katsayılar,  $F(s)$ 'nin yüksek yarı-bantından gelmektedir, diğer  $N/4$  noktalar ikinci çeyrek-banttan gelir vb. gibi.

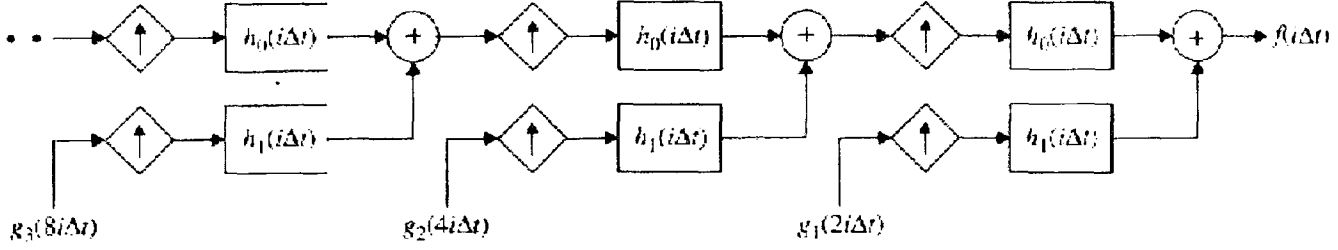


Şekil 6.12 Ayrık dalgacık dönüşümü algoritması

Impulse response,  $h_j$ , her iterasyonda ölçeği ikiler. Böylece, dikey bir dalgacık dönüşümü elde ederiz. Temel dalgacık  $h(t) = \delta(t) - \sin c(at)$ , ve temel fonksiyonlar  $\{2^{-j/2} h(2^j t - n)\}$  olur. Bu suretle, alt-bant kodlama, temel olarak bir zaman frekans dönüşüm tekniğidir, bir zaman-ölçek dalgacık dönüşümünü tanımlamak için kullanılmıştır.

Yukarıda sözü edilen bu algoritma, bazen hızlı dalgacık dönüşümü olarak, bazen de

görünümünden dolayı Mallat'ın herring balığı algoritması olarak adlandırılır. Ters dönüşüm Şekil 6.13'te görüldüğü üzere ters işlem yapılarak elde edilir.



Şekil 6.13 Ters ayrık dalgacık dönüşümü algoritması

#### 6.4.1 Temel Fonksiyonlar

Şekil 6.12'de, dönüşüm katsayılarının her birinin  $f(i\Delta t)$ 'nin, tekrar tekrar  $h_0(i\Delta t)$  ve sonrasında  $h_1(i\Delta t)$  ile convolve edilerek oluşturulduğunu görüyoruz. Bu suretle, bu dalgacık dönüşümünün temel fonksiyonları  $h_1(i\Delta t)$  ve diğer fonksiyonlardır.

#### 6.5 Ayrık Dalgacık Dönüşüm Tasarımı

Şimdi, bir ayrık dalgacık dönüşümü içinde kullanılan temel bir dalgacık tasarımı oluşturmak için hazırlanalım. Şunu biliyoruz ki, bir filtre bankası uygulamasında ideal alçak-geçirgen ve yarı-geçirgen olmak, filtreleri gerekli kılmaz. Benzer şekilde, ayrık dalgacık dönüşümü için, (6.21) numaralı eşitliğe uyan herhangi bir alt-bant kodlama çiftini kullanabiliriz.

Frekans alanındaki (6.21) numaralı eşitliği yazarsak,

$$F(s) = 2 \left[ \frac{1}{2} G_0(s) H_0(s) + \frac{1}{2} G_1(s) H_1(s) \right] = 2 \left[ \frac{1}{2} F(s) H_0(s) H_0(s) + \frac{1}{2} F(s) H_1(s) H_1(s) \right] \quad (6.22)$$

bulunur. Buradan;

$$F(s) = F(s) [H_0^2(s) + H_1^2(s)] \quad (6.23)$$

sağlanır ve iki filtre transfer fonksiyonu aşağıdaki şartı sağlamak zorundadır.

$$H_0^2(s) + H_1^2(s) = 1 \quad \text{ve} \quad 0 \leq |s| \leq s_N \quad (6.24)$$

Transfer fonksiyonlarının, burada kareleri alınmıştır. Çünkü  $f(t)$ , her filtrede bir kere kodlamada, bir kere de kodu çözme sırasında, iki kez convolve edilmiştir.

$H_0(s)$ 'nin dalgacık dönüşümünde kullanmak istediğimiz bir düz kenarlı alçak-geçirgen transfer fonksiyonu olduğunu varsayalım. Bu bilgi doğrultusunda,  $H_1(s)$  aşağıdaki gibi bulunur.

$$H_1^2(s) = 1 - H_0^2(s) \quad (6.25)$$

Böylece, iyi seçimli bir alçak-geçirgen filtre ayrık bir dalgacık dönüşümü tasarlamak için tamamıyla gereklidir.

### 6.5.1 Ayna Filtreler

Şekil 6.9b ve 6.10b'yi karşılaştırsak, şunu görürüz ki ideal bant-geçirgen filtre durumu için  $h_1(i\Delta t)$ ,  $h_0(i\Delta t)$  olarak görülebilir. Bu yer değiştirme teoremine göre,

$$\mathfrak{I}^{-1}\{H(s-a)\} = e^{j2\pi at} h(t) \Rightarrow \mathfrak{I}^{-1}\{H(s-s_N)\} = e^{j2\pi\left(\frac{1}{2\Delta t}\right)i\Delta t} h(i\Delta t) = (-1)^i h(i\Delta t) \quad (6.26)$$

işlemi göz önünde bulundurulur. Spektrumun yarım periyotlu bir değişimi, sinyalin rakamlı örneklerinin işaretini değiştirerek kolayca sonuçlandırılabilir.

Bu yaklaşımı, ,alt-bant filtrelerinden daha genel tasarım içinde kullanabiliriz.  $h_1(i\Delta t)$ 'yi seçersek,

$$h_1((N-1-i)\Delta t) = (-1)^i h_0(i\Delta t) \quad (6.27)$$

eşitliği görülmektedir. Burada  $N$ ,  $h_0(i\Delta t)$ 'nin uzunluğudur. Buradan, benzer yüksek-geçirgen filtreyi elde ederiz.  $h_1(i\Delta t)$  filtresi,  $h_0(i\Delta t)$ 'nin ayna filtresi olarak adlandırılır. Eğer  $h_0(i\Delta t)$  kısa aralıklı ise,  $h_1(i\Delta t)$ 'nin olabildiğince kısa olduğuna inandırılabiliriz. Simetri özelliği,

(6.25) numaralı eşitlikte olduğu gibi değerlendirilecektir.

$$H_0^2\left(\frac{s_N}{2} + s\right) = 1 - H_0^2\left(\frac{s_N}{2} - s\right) \quad (6.28)$$

### 6.5.2 Ölçek Vektör

Ayrık bir dalgacık dönüşümünü geliştirmek için, sadece kesin şartlara ras gelen bir alçak-geçirgen filtre impulse responsu'na,  $h_0(k)$  'ya ihtiyacımız vardır. Bu impulse response, bazen

ölçek vektör olarak adlandırılır.

$h_0(k)$  'dan, bağımlı bir fonksiyon elde edebiliriz,  $\phi(t)$ . Buna ölçek vektör denir. Aynı zamanda  $h_1(k)$  'yı, bundan  $\phi(t)$  'yi ve temel dalgacık  $\psi(t)$  'yi elde edebiliriz. Eğer ölçek vektör, sıfır olmayan girişlerin sadece bir sonlu değeri ise, buradan  $\phi(t)$ ,  $\psi(t)$  ve sonuç fonksiyonları tamamıyla compact support'a sahip olacaklardır. Bu onların t ekseninde oldukça kısa bir aralık dışında sıfır olacaklarını gösterir.

Hakikaten, eğer hem  $h_0(k)$ , hem de  $\phi(t)$  'ye sahipsek, bunu diğerini elde etmek için kullanabiliriz. Genellikle  $h_0(k)$  ile başlamak daha kolaydır.

$$\sum_k h_0(k) = \sqrt{2} \quad \text{ve} \quad \sum_k h_0(k)h_0(k+2l) = \delta(l) \quad (6.29)$$

Buradan bir ölçek fonksiyon oluşur.

$$\phi(t) = \sum_k h_0(k)\phi(2t-k) \quad (6.30)$$

$\phi(t)$ , diktörgegen biçiminde ton fonksiyonlarının ölçekli versiyonları ile  $h_0(k)$  bükümü tekrar edilerek, nümerik olarak da hesaplanabilir. Bu;

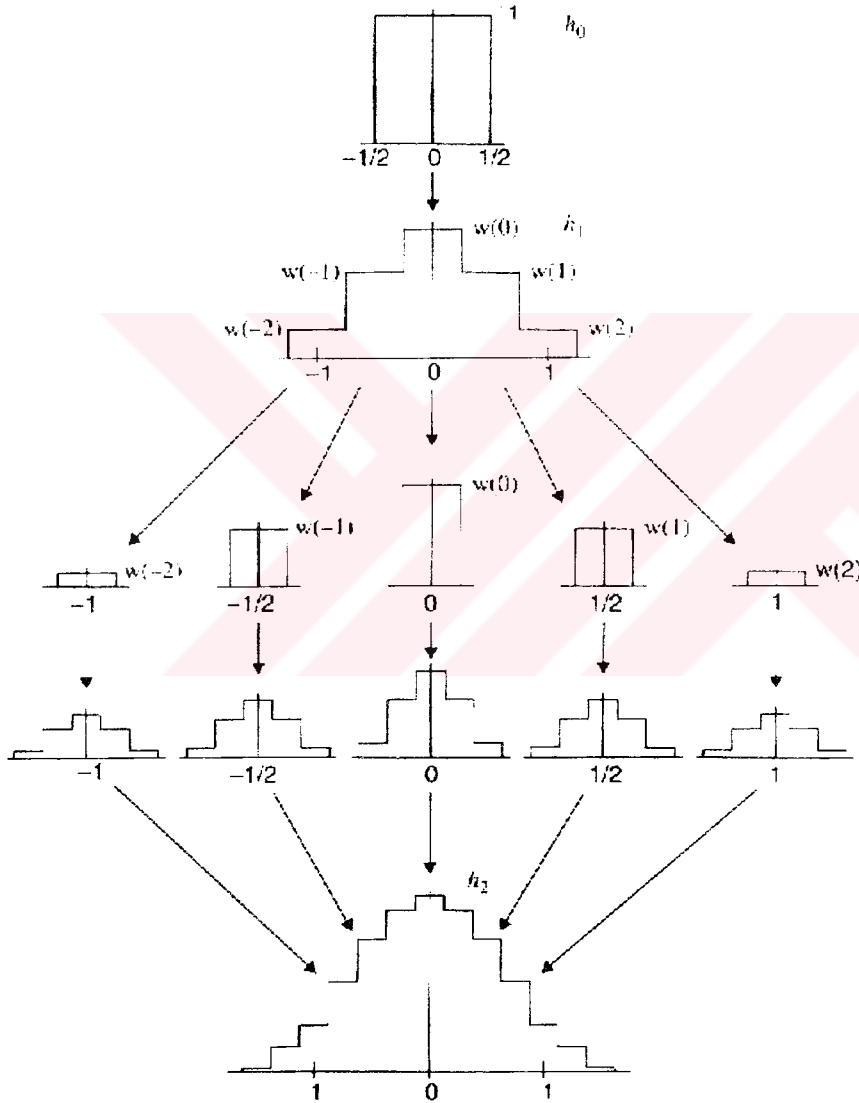
$$\phi(x) = \lim_{i \rightarrow \infty} \eta_i(x) \quad (6.31)$$

$$\eta_i(x) = \sqrt{2} \sum_n h_0(n)\eta_{i-1}(2x'-n) \quad (6.32)$$

olduğu yerde  $\phi(t)$ 'ye titreşimli bir yaklaşımdır ve

$$\eta_0(x) = \pi(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \quad (6.33)$$

şeklinde gösterilir.



Şekil 6.14 Ölçek fonksiyonu oluşturma

Burada ilk iterasyonun,  $h_0(k)$  değerlerine sahip titreşimli bir fonksiyon yarattığını göz önünde bulunduralım. Bunun yanında yaklaşım yoğunluğu ikilenir ve yaklaşım her bir iterasyon sonrasında daha düzgün bir hal alır. Örneğin dokuz iterasyon, bir 1,024 nokta model fonksiyonu içinde dikdörtgenimsi bir dört nokta alacaktır. Ölçek fonksiyon  $\phi(t)$ , ayrıık alçak-

geçirgen filtre impulse response  $h_0(k)$  olarak, aynı genel şekle sahip sürekli bir fonksiyondur.

Diğer taraftan, eğer ölçek fonksiyon  $\phi(t)$  ile başlarsak, birim değişiklikler altında dikey olmak zorundadır. Bu da;

$$\langle \phi(t-m), \phi(t-n) \rangle = \delta_{m,n} \quad (6.34)$$

eşitliği ile gösterilmektedir. Buradan  $h_0(k)$  ;

$$h_0(k) = \langle \phi_{1,0}(t), \phi_{0,k}(t) \rangle \quad (6.35)$$

ile bulunur.

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \quad j = 0,1,\dots \quad k = 0,1,\dots, 2^j - 1 \quad (6.36)$$

Ölçek vektör, sıfır olmayan çıktıların sadece sonlu bir sayısına sahipse, sonuç vektörler compact support'a sahip olacaktır.

Eğer arzu edilen bir ölçek fonksiyon,  $\hat{\phi}(t)$ , dikey değil ise, spektrumunun gerçek normalizasyonu kadar dikey olan genel hali ile kullanılabilir. Bu ;

$$\Phi(s) = \frac{C\hat{\Phi}(s)}{\sqrt{\sum_{n=-\infty}^{\infty} 0\hat{\Phi}(s-2\pi n)}} \quad (6.37)$$

eşitliği ile gösterilir.  $C$  bir titreşimdir.

### 6.5.3 Dalgacık Vektörü

$h_0(k)$  ve  $\phi(t)$  tanımlarını elde ettiğimize göre, bir ayrık yüksek geçirgenli impulse response tanımlayarak teorimizi geliştirmeye devam edelim. Bu gelişim sonucu dalgacık vektörüne ulaşırız. Dalgacık vektörü;

$$h_1(k) = (-1)^k h_0(-k+1) \quad (6.38)$$

ile elde edilir. Buradan temel dalgacık;

$$\psi(t) = \sum_k h_1(k) \phi(2t-k) \quad (6.39)$$

dikey bir dalgacık kümesindenden;

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (6.40)$$

elde edilir.

#### 6.5.4 Dalgacık Dönüşümü Hesaplama

Dikey dalgacıklar verilsin, bant limitli sürekli fonksiyon  $f(t)$  'nin dalgacık seri yayılımı;

$$c_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt \quad \text{ve} \quad f(t) = \sum_{j,k} c_{j,k} \psi_{j,k}(t) \quad (6.41)$$

ve örnek fonksiyonun ayırık dalgacık dönüşümü;

$$c_{j,k} = \sum_i f(i\Delta t) \psi_{j,k}(i\Delta t) \quad \text{ve} \quad f(i\Delta t) = \sum_{j,k} c_{j,k} \psi_{j,k}(i\Delta t) \quad (6.42)$$

ile gösterilir. Katsayılar ve toplamlar, tek bir integer değişken olan  $n$  ile sıralanırlar.

$$n = 0, 1, \dots, N-1.$$

$$n = 2^j + k, \quad j = 0, 1, \dots, \log_2(N) - 1, \quad k = 0, 1, \dots, 2^j - 1 \quad (6.43)$$

İlk olarak büyük ölçekli katsayılar elde ettiği için, biz bundan top-down algoritma olarak söz edeceğiz. Mukayase edersek, Mallat'ın herring balığı algoritması ilk önce küçük ölçekli katsayılar elde etmektedir.

Bu tasarım, ilk önce  $h_0(k)$  'yı bulmayı ve sonra da sırasıyla benzer ölçek fonksiyonu oluşturmayı veya dikey bir ölçek fonksiyon seçmeyi ve (6.35) numaralı eşitlikten  $h_0(k)$  'yı determine etmeyi içermektedir. Ölçek fonksiyon (6.37) numaralı eşitlik ile dikey yapılabilir. Buradan dalgacık vektörü,  $h_1(k)$ , (6.38) numaralı eşitlik ile, temel dalgacık (6.39) numaralı eşitlik ile elde edilir.

Ayrık dalgacık dönüşümü, (6.42) numaralı eşitlik veya hızlı dalgacık dönüşümü hering balığı algoritması ile doğrudan tamamlanabilir. Sonraki ölçek fonksiyonunun ve dalgacığın açık yapısına gerek duymaz ve hesaplama yönünden daha kullanışlıdır.

Matematiksel olarak kusursuz olmak için, (6.29) numaralı eşitlikteki şartlar sıkı bir pencere oluşturan  $\{\psi_{j,k}(t)\}$  dalgacıkları meydana getirir ve böylece tersi alınabilir bir dönüşümü destekler.

### 6.5.5 Örnekler

Bir dalgacık dönüşümü oluşumunu üç örnek ile ifade edelim.

#### Örnek 1

İdeal alçak-geçirgen ve bant-geçirgen filtreler kullanalım.

$$h_0(k) = \frac{1}{\sqrt{2}} \sin c\left(\pi \frac{k}{2}\right) \quad \text{ve} \quad h_1(k) = \sqrt{2} \delta(k) - h_0(k) \quad (6.44)$$

ve

$$\phi(t) = \sin c(\pi t) \quad \text{ve} \quad \psi(t) = 2\phi(2t) - \phi(t) \quad (6.45)$$

Bu eşitlik bize, sinüs dalgacıklarına bağlı bir ayrık dalgacık dönüşümünü verir. Biz bu dalgacıkların compact support'a sahip olmadıklarını farzedelim.

#### Örnek 2

Aşağıdaki eşitlik göz önünde bulundurulursa,

$$h_0(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0,1 \\ 0 & \text{otherwise} \end{cases} \quad \text{ve} \quad h_1(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ -\frac{1}{\sqrt{2}} & k = 1 \\ 0 & \text{otherwise} \end{cases} \quad (6.46)$$

$\psi(t)$  Haar fonksiyonudur ve Haar dönüşümünü uygular. Bu ölçek vektörün iki tane sıfır olmayan girişi vardır. Bunun yanında Haar dönüşümünün compact support'u yoktur.

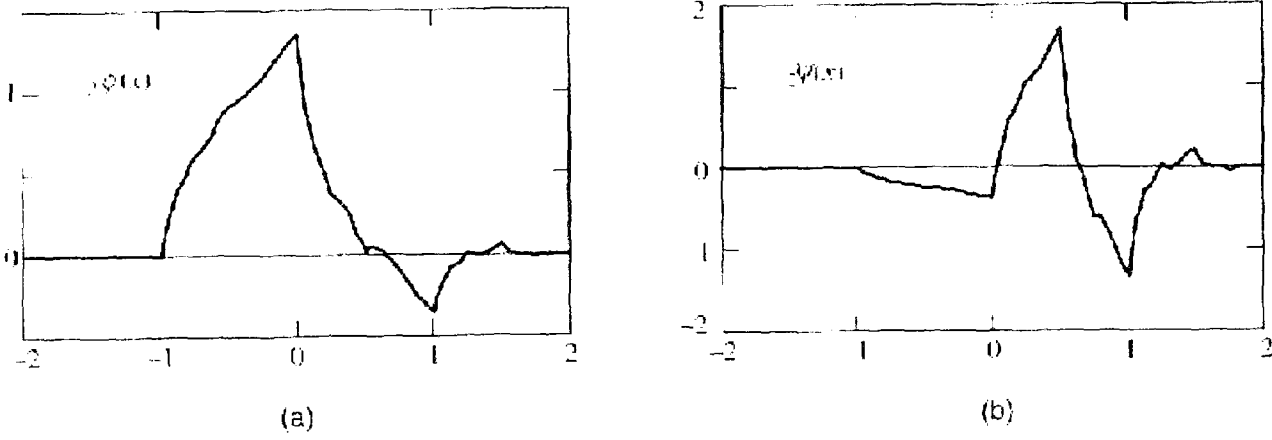
#### Örnek 3

$h_0(k)$ , dört sıfır olmayan elementi vardır.

$$4\sqrt{2}h_0(k) = \begin{cases} (1+\sqrt{3}) & k = 0 \\ (3+\sqrt{3}) & k = 1 \\ (3-\sqrt{3}) & k = 2 \\ (1-\sqrt{3}) & k = 3 \\ 0 & \text{otherwise} \end{cases} \quad (6.47)$$

(6.28) numaralı eşitliğe uygunluk gösterir ve bu yüzden bir ölçek vektördür. Ölçek vektörü ve

dalgacık, Şekil 6.15'te görüldüğü gibi, Şekil 6.14'te de belirlenen yapı tarafından oluşturulur.



Şekil 6.15 Daubechies'in (a) ölçek fonksiyonu (b) dalgacı

### 6.5.6 Compact Support'a Sahip Dikey Dalgacıklar

Daubechies compact support'a sahip dikey dalgacıkların bir ailesini,  $\{\psi(x)\}$ 'i yarattı.  $r$  indeksinin her bir integer değeri için dalgacık kümesi;

$$\{\psi_{j,k}(x)\} = \{2^{j/2} \psi(2^j x - k)\} \quad (6.48)$$

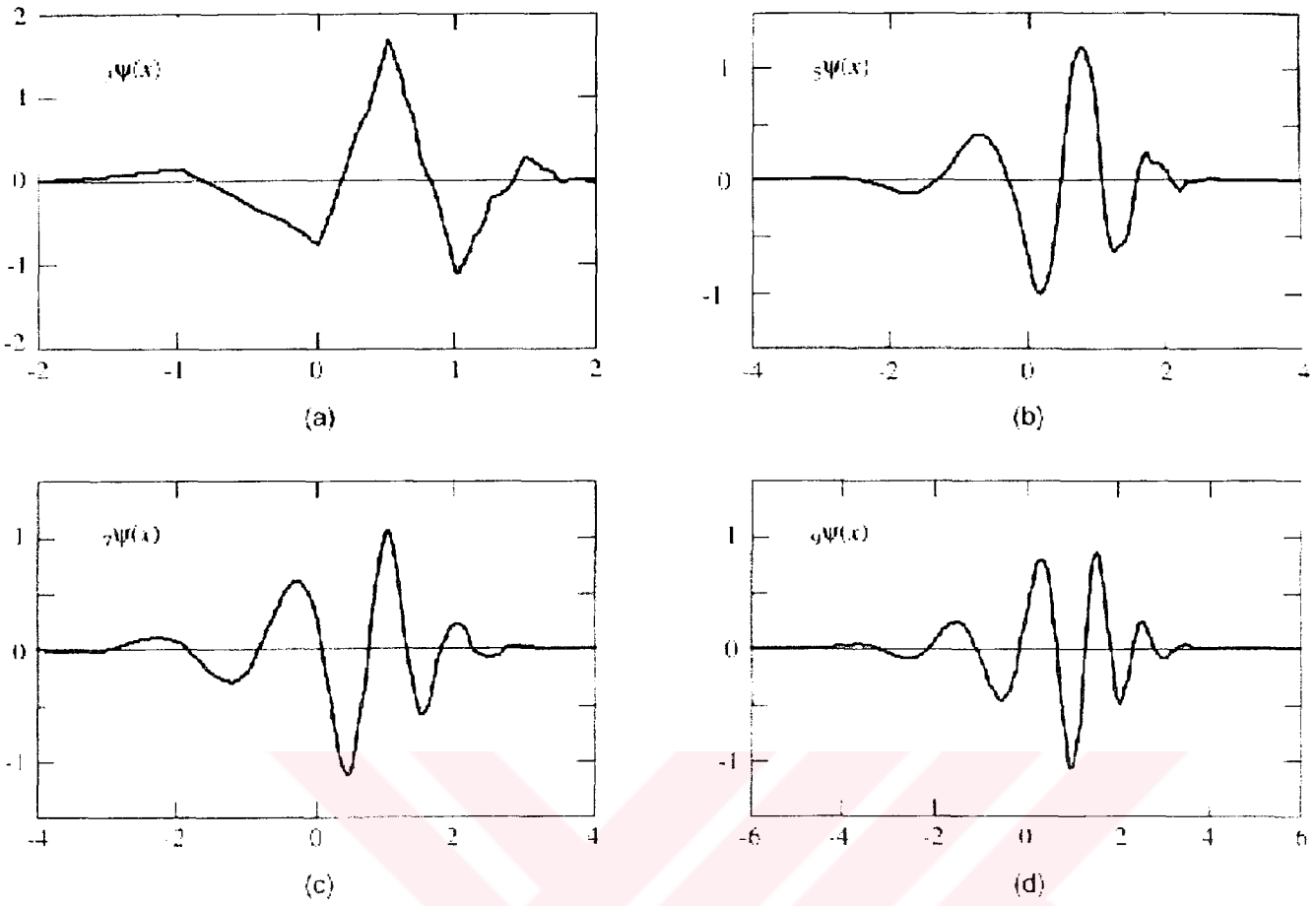
olur.  $j$  ve  $k$  integer değerlerdir. Bununla birlikte,  $\psi(x)$ ,  $[0, 2r-1]$  aralığı dışında sıfırdır. İlk  $r$  momenti ihmal edilir, buradan

$$\int_{-\infty}^{\infty} x^n \psi(x) dx = 0 \quad n = 0, 1, \dots, r \quad (6.49)$$

elde edilir ve sürekli türevlerinin sayısı yaklaşık olarak  $r/5$  kadardır. Bu oldukça iyi bir durumu tanımlar. İlginç olan,  $\psi(x)$ 'in Haar dönüşümünün temel dalgacı olmasıdır.

Tablo 6.1,  $r=3, 5, 7$  ve  $9$  için dikey dalgacıkları oluşturan  $h_0(k)$  dizilerini göstermektedir.

Şekil 6.16, benzer dalgacıkların işaretlerini gösterir. Bu fonksiyonların her ikisinin de, daha geniş ve düzenli olduğunu göz önünde bulunduralım.



Şekil 6.16  $r = 3, r = 5, r = 7$  ve  $r = 9$  için dikey dalgacıklar

Çizelge 6.1 Şekil 6.16'daki dikey dalgacıkların ayrık filtre dizileri

( $r=3,5,7$  ve  $9$ )

|        |        |        |        |        |        |        |        |        |       |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| .3327  | .8069  | .4599  | -.1350 | -.0854 | .0352  | .      | .      | .      | .     |
| .1601  | .6083  | .7243  | .1384  | -.2423 | -.0322 | .0776  | -.0062 | -.0126 | .0033 |
| .0779  | .3965  | .7291  | .4698  | -.1439 | -.2240 | .0713  |        |        |       |
| .0806  | -.0380 | -.0166 | .0126  | .0004  | -.0018 | .0004  |        |        |       |
| .0381  | .2438  | .6048  | .6573  | .1332  | -.2933 | -.0968 | .1485  | .0307  |       |
| -.0676 | .0003  | .0224  | -.0047 | -.0043 | .0018  | .0002  | -.0003 | .0000  |       |

## 7. İKİ BOYUTLU AYRIK DALGACIK DÖNÜŞÜMÜ

Birimsel görüntü dönüşümleri ile, iki boyutlu ölçek fonksiyonun ayrıştırılabilir olduğu bir durum göz önünde bulunduralım. Bu  $\phi(x)$  'in tek boyutlu bir ölçek fonksiyon olduğu yerde;

$$\phi(x, y) = \phi(x)\phi(y) \quad (7.1)$$

şeklinde ifade edilir. Eğer  $\psi(x)$ , eş dalgacı ise, sonrasında üç adet iki boyutlu temel dalgacıklar

$$\psi^1(x, y) = \phi(x)\psi(y) \quad \psi^2(x, y) = \psi(x)\phi(y) \quad \psi^3(x, y) = \psi(x)\psi(y) \quad (7.2)$$

iki boyutlu dalgacıkların temel yapısını oluştururlar. Üst olarak yazılan değer bir üstel değerden çok, burada index değeri olarak kullanılır. Fonksiyonlar kümesi  $L^2(R^2)$  için bir dikey temeldir.

$$\{\psi^l_{j,m,n}(x, y)\} = \{2^j \psi^l(x - 2^j m, y - 2^j n)\} \quad j \geq 0 \quad l = 1, 2, 3 \quad (7.3)$$

$j, l, m$  ve  $n$  integer değerlerdir.

### 7.1 İleri Dönüşüm

$N \times N$  bir görüntü ile başlayalım.  $j = 0$  için ölçek  $2^j = 2^0 = 1$  'dir ve bu gerçek görüntünün ölçeğidir.  $j$  'nin her bir daha büyük integer değeri, ölçeği çiftler ve yoğunluğu ikiye böler. Bu konuda yapılan çalışmaların bazılarında,  $j$  değeri ölçekten daha çok yoğunluğu indexlemek için kullanılır. Görüntü, iki boyutlu dalgacıkların süreleri içinde genişletilebilir. Dönüşümüm her aşamasında, görüntü dört parçaya ayrılır (Şekil 7.1). Dört görüntüden herbiri dalgacık temel görüntülerinden biri ile iç çarpımları kullanılarak oluşturulur.  $j = 1$  olarak, ilk aşamada

$$f_2^0(m, n) = \langle f_1(x, y), \phi(x - 2m, y - 2n) \rangle$$

$$f_2^1(m, n) = \langle f_1(x, y), \psi^1(x - 2m, y - 2n) \rangle \quad (7.4)$$

$$f_2^2(m, n) = \langle f_1(x, y), \psi^2(x - 2m, y - 2n) \rangle$$

$$f_2^3(m, n) = \langle f_1(x, y), \psi^3(x - 2m, y - 2n) \rangle$$

Diğer aşamalar için  $j > 1$  için,  $f_{2^j}^o(x, y)$  tamamıyla aynı yollar kullanılarak  $2^{j+1}$  ölçeğe, dört daha küçük görüntüye bölünecektir. Şekil 7.1c'de bu işlem görülmektedir. Sonuçta Şekil 7.1d'de görüldüğü üzere, Haar dönüşümünününe benzer bir düzen oluşacaktır.

Kat olarak iç çarpımları yazarsak, aşağıdaki eşitliklere ulaşırız.

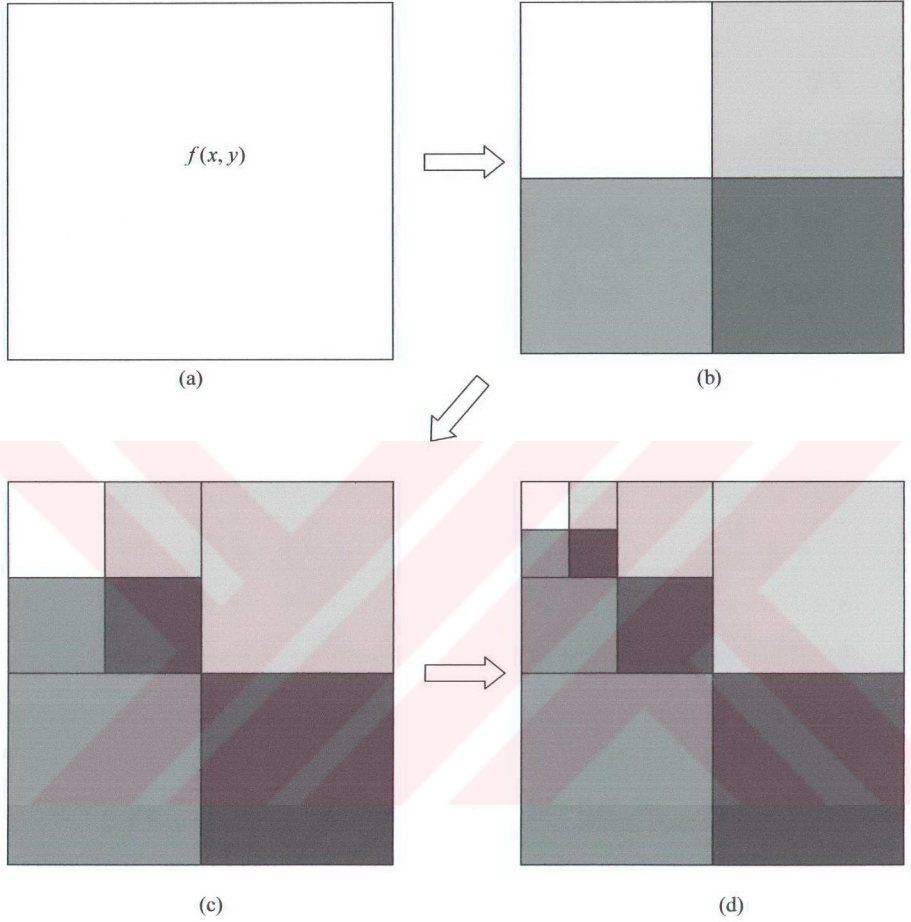
$$\begin{aligned}
 f_{2^{j+1}}^o(m, n) &= \{f_{2^j}^o(x, y) * \phi(-x, -y)\}_{2m, 2n} \\
 f_{2^{j+1}}^1(m, n) &= \{f_{2^j}^1(x, y) * \psi^1(-x, -y)\}_{2m, 2n} \\
 f_{2^{j+1}}^2(m, n) &= \{f_{2^j}^2(x, y) * \psi^2(-x, -y)\}_{2m, 2n} \\
 f_{2^{j+1}}^3(m, n) &= \{f_{2^j}^3(x, y) * \psi^3(-x, -y)\}_{2m, 2n}
 \end{aligned} \tag{7.5}$$

Her bir aşamada aynı dört filtre örneği kullanılır.

Ölçek ve dalgacık fonksiyonları ayrıştırılabilir olduklarından, her kat  $f_{2^j}^o(x, y)$ 'nin satır ve sütunlarında bir boyutlu katlara ayrılmaktadır. Şekil 7.2'de bu diagram düzeninde gösterilmektedir.

Birinci aşamada,  $f_1(x, y)$  görüntüsün satırlarını  $h_0(-x)$  ve  $h_1(-x)$  ile katlıyoruz, sonra iki sonuç boyutun tek sayılı sütunlarını ihmal ediyoruz.  $N/2$ 'ye  $N$  boyutlarından herbir sütun,  $h_0(-x)$  ve  $h_1(-x)$  ile oluşturulur. Tek sayılı satırlar ihmal edilir. Sonuçta  $N/2$ 'ye  $N/2$  boyutta görüntü elde edilmiş olur.

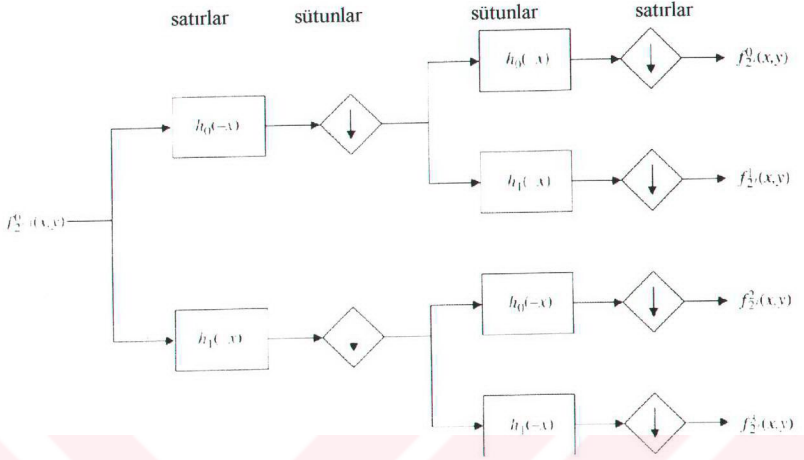
İki boyutlu ayrıştırılabilir dalgacık dönüşümü böylece hızlı bir şekilde hesaplanabilir. Dönüşüm işlemi,  $J$  kadar iterasyon devam ettirilir. Eğer dönüşüm katsayıları birikmiş nokta tamlığı ile hesaplanırsa, ters dönüşüm küçük bir kayıp ile gerçek görüntüyü tekrar oluşturabilir.



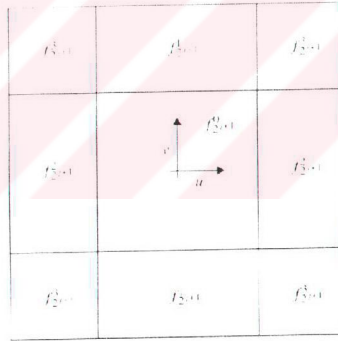
Şekil 7.1 İki boyutlu ayrık dalgacık dönüşümü

(a) gerçek görüntü (b) Birinci aşama (c) İkinci aşama (d) Üçüncü aşama

Şekil 7.3 , dört adet birbirini takip eden daha yüksek ölçek görüntülerin nereden geldiğini göstermektedir. Her ölçekte,  $f_{2'}^0(x, y)$  önceki aşamadan gelen düşük frekanslı bilgi içerir,  $f_{2'}^1(x, y)$ ,  $f_{2'}^2(x, y)$  ve  $f_{2'}^3(x, y)$  sırasıyla yatay, dikey ve çapraz kenar bilgilerini içerir.



Şekil 7.2 Ayrık dalgacık dönüşümü görüntü ayrıştırma işlemi



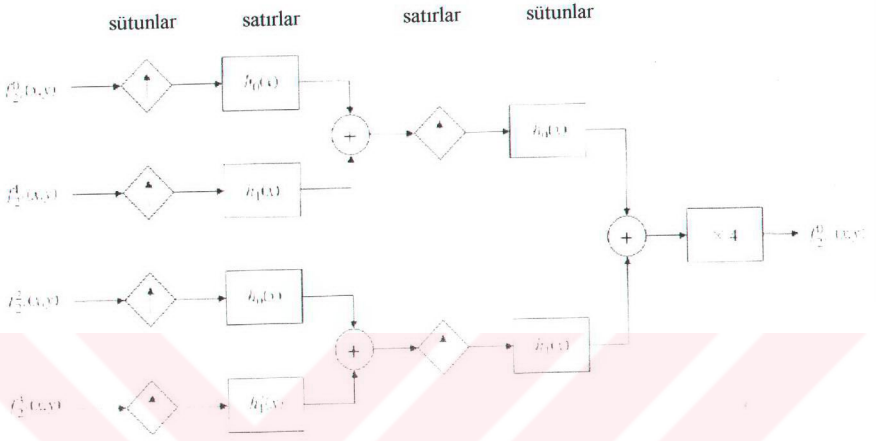
Şekil 7.3 Frekans alanında ayrık dalgacık dönüşümü

## 7.2 Ters Dönüşüm

Dönüşümün tersi, ileri dönüşüme benzer bir işlem ile gerçekleşir. Bu işlemler Şekil 7.4'te algoritma olarak gösterilmektedir.

Her aşamada, her kolonun soluna sıfırlar kolonu eklenerek örnekleme yapılır. Daha sonra

satırları  $h_0(x)$  ve  $h_1(x)$  ile katlıyoruz ve  $N/2$ 'ye  $N$  boyutta eklenir. İki sonuç boyut, her satırın üzerine sıfırlar satırı eklenerek  $N$ 'e  $N$  boyutta getirilir.



Şekil 7.4 Ters ayrık dalgacık dönüşümü yeniden görüntü oluşturma işlemi

### 7.3 Örnekler

Şekil 7.5 ,iki boyutlu ayrık bir dalgacık dönüşümünün ilk aşamasını hesaplayan sayısal bir örnektir. Şekil 7.6'da aynı görüntünün ters dalgacık dönüşümünün son aşamasını gösterir.

$$f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 4 & 4 & 2 & 0 & 0 \\ 0 & 1 & 4 & 8 & 8 & 4 & 1 & 0 \\ 0 & 1 & 4 & 8 & 8 & 4 & 1 & 0 \\ 0 & 0 & 2 & 4 & 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{1,0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 5 & 4 & 2 & 0 & 0 \\ 1 & 3 & 8 & 11 & 8 & 3 & 1 & 0 \\ 1 & 3 & 8 & 11 & 8 & 3 & 1 & 0 \\ 0 & 2 & 4 & 5 & 4 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 & 0 & 0 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 & 0 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{2,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 1 & 8 & 8 & 1 \\ 1 & 8 & 8 & 1 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{2,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 3 & *3 & -1 \\ 1 & 3 & -3 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{3,0} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 1 & 9 & 9 & 1 \\ 1 & 11 & 11 & 1 \\ 1 & 9 & 9 & 1 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{3,1} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -3 & -3 & 0 \\ 0 & -2 & -2 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{3,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{4,0} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 9 & 9 & 1 \\ 1 & 9 & 9 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$f_{4,1} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & -3 & 0 \\ 0 & -1 & -1 & 0 \end{bmatrix}$$

$$f_{4,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 3 & -3 & -1 \\ 1 & 3 & -3 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_{4,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Şekil 7.5 İki boyutlu ayrık dalgacık dönüşümünün hesaplamasına ilişkin örnek



## 8. BİDİKEY DALGACIK DÖNÜŞÜMLERİ

Compact support ile dikey dalgacıklar haline gelen fonksiyonlar, gerçekleşmesi istenen simetri özelliklerinden yoksundur. Eğer  $\psi(t)$ , tek veya çift bir fonksiyon olsaydı, simetri özellikler sağlanmış olurdu. İki farklı dalgacık yapısı kullanarak,  $\psi(x)$  (ayırıştırma işlemi için) ve  $\tilde{\psi}(x)$  (yeniden oluşturma işlemi için), compact support'a sahip simetrik dalgacıklar oluşturabiliriz. İki dalgacık her biri için eşlidir ve dalgacık kümeleri  $\{\psi_{jk}(x)\}$  ve  $\{\tilde{\psi}_{jk}(x)\}$ , bidikeydirler.

$$\langle \psi_{j,k}, \tilde{\psi}_{l,m} \rangle = \delta_{j,l} \delta_{k,m} \quad (8.1)$$

Bu işlemden sonra ayırıştırma işlemi için

$$c_{j,k} = \langle f(x), \tilde{\psi}_{j,k}(x) \rangle \quad \text{ve} \quad d_{j,k} = \langle f(x), \psi_{j,k}(x) \rangle \quad (8.2)$$

ve yeniden oluşturma işlemi için

$$f(x) = \sum_{j,k} c_{j,k} \psi_{j,k}(x) = \sum_{j,k} d_{j,k} \tilde{\psi}_{j,k}(x) \quad (8.3)$$

eşitlikleri göz önünde bulundurulur. Dalgacıklardan biri ayırıştırma için, diğeri yeniden oluşturma işlemi için kullanılabilir. Bidikey dalgacık dönüşümü, compact support'a sahip simetrik (tek veya çift) dalgacıklar kullanımına olanak sağlar.

### 8.1 Uygulama

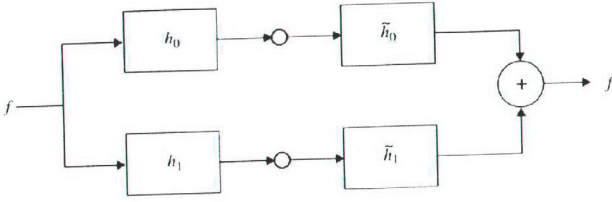
Bir boyutlu bidikey dalgacık dönüşümü, dört adet ayrık filtreye gereksinim duyar. Transfer fonksiyonları  $h_0(n)$  ve  $\tilde{h}_0(n)$  olan, iki tane alçak-geçirgen filtre seçmek zorundayız.

$$H_0(0) = \tilde{H}_0(0) = 1 \quad \text{ve} \quad H_0(s_N) = \tilde{H}_0(s_N) = 0 \quad (8.4)$$

$s_N = 1/2\Delta x$  frekanstır. Bunlardan, iki bant geçirgenli filtre meydana getiririz. Transfer fonksiyonlar;

$$h_1(n) = (-1)^n h_0(1-n) \quad \tilde{h}_1(n) = (-1)^n \tilde{h}_0(1-n) \quad (8.5)$$

elde edilir. Bu dört filtreyi kullanarak şimdi, İleri Dalgacık Dönüşümü Hering balığı algoritmasını tamamlayabiliriz. Şekil 8.1.



Şekil 8.1 Bidikey dalgacık dönüşümünün ayrıştırma ve yeniden oluşturma işlemi

## 8.2 Bidikey Dalgacıklar

Bidikey dalgacık filtreleri üzerine şartlar;

$$\sum_n h_o(n) = \sum_n \tilde{h}_o(n) = \sqrt{2} \quad \text{ve} \quad \sum_n h_1(n) = \sum_n \tilde{h}_1(n) = 0 \quad (8.6)$$

ve etkili yeniden oluşturma özelliği ile aşağıdaki eşitliği gerekli kılar.

$$H_0(s)\tilde{H}_0(s) + H_1(s)\tilde{H}_1(s) = H_0(s)\tilde{H}_0(s) + H_0(s-s_N)\tilde{H}_0(s-s_N) = 1 \quad (8.7)$$

İki ölçek fonksiyon verilir.

$$\Phi(2s) = \tilde{H}_0(s)\Phi(s) = \prod_{n=0}^{\infty} \tilde{H}_0(s/2^n) \quad \text{ve} \quad \tilde{\Phi}(2s) = H_0(s)\tilde{\Phi}(s) = \prod_{n=0}^{\infty} H_0(s/2^n) \quad (8.8)$$

ve dalgacıklar

$$\psi(x) = \sqrt{2} \sum_n h_1(n+1)\phi(2x-n) \quad \text{ve} \quad \tilde{\psi}(x) = \sqrt{2} \sum_n \tilde{h}_1(n+1)\tilde{\phi}(2x-n) \quad (8.9)$$

eşitlikleri ile elde edilir.

## 8.3 Bidikey Dalgacıklar Oluşturma

Bidikey dalgacık tasarımı, ayrıık impulse response'ları geliştirmeyi ,zorunlu kılmaktadır. Bu araştırmanın hareketli yanadır ve bazı bilim adamları bu gibi filtreleleri ve benzer bidikey dalgacıkları bir katalog haline getirmişlerdir.

Çizelge 8.1 Şekil 8.1’de gösterilen bidikey

dalgacıklar için ayrık dalgacık değerleri

Laplace analizi filtresi

$$h_0 = \sqrt{2}[-.05 \quad .25 \quad .6 \quad .25 \quad -.05]'$$

Laplace sentezi filtresi

$$h_0 = \sqrt{2}[-.0107 \quad -.0536 \quad .2607 \quad .6071 \quad .2607 \quad -.0536 \quad -.0107]'$$

Laplace sentezi filtresi

$$h_0 = \sqrt{2}[-.0107 \quad -.0536 \quad .2607 \quad .6071 \quad .2607 \quad -.0536 \quad -.0107]'$$

Spline 2 filtresi

$$h_0 = \sqrt{2} [.25 \quad .5 \quad .25]'$$

Spline 4 filtresi

$$\tilde{h}_0 = \frac{\sqrt{2}}{128} [3 \quad -6 \quad -16 \quad 38 \quad 90 \quad 38 \quad -16 \quad -6 \quad 3]'$$

18-nokta analizi filtresi

$$h_0 = \sqrt{2} [.0012 \quad -.0007 \quad -.0118 \quad .0117 \quad .0713 \quad -.0310 \quad -.2263 \quad .0693 \quad .7318 \\ .7318 \quad .0693 \quad -.2263 \quad -.0310 \quad .0713 \quad .0117 \quad -.0118 \quad .0007 \quad .0012]'$$

18-nokta sentezi filtresi

$$h_0 = \sqrt{2} [.0012 \quad .0007 \quad -.0113 \quad -.0114 \quad .0235 \quad .0017 \quad -.0444 \quad .2044 \quad .6479 \\ .6479 \quad .2044 \quad -.0444 \quad .0017 \quad .0235 \quad -.0114 \quad -.0113 \quad .0007 \quad .0012]'$$

Örneğin Cohen, Daubechies ve Feauveau gibi araştırmacılar,  $\phi(x)$ 'i bir B-spline fonksiyonu olarak seçmişler,  $H_0(s)$ 'i  $\cos(s)$  içinde bir polinom olarak geliştirmişlerdir. Vetterli ve Herley gibi araştırmacılar ise, diaphantine eşitlikler ve sürekli kesirler teorilerini esas alan yaklaşımlar üzerinde çalışmışlardır. Genellikle, daha uzun impulse response kullanmak daha

düzenli dalgacıklar, daha büyük bir türev sayıları verir. Çizelge 8.1 , üç çift ölçek vektör çiftini göstermektedir.

#### 8.4 İki Boyutlu Bidikey Dalgacıklar

İleri iki boyutlu dönüşüm için bidikey dalgacıklar, (7.2) numaralı eşitlik ile gösterilmiştir. Ters dönüşüm ise

$$\tilde{\psi}^1(x, y) = \tilde{\phi}(x)\tilde{\psi}(y) \quad \tilde{\psi}^2(x, y) = \tilde{\psi}(x)\tilde{\phi}(y) \quad \tilde{\psi}^3(x, y) = \tilde{\phi}(x)\tilde{\psi}(y) \quad (8.10)$$

eşitlikleri ile gösterilir. İki boyutlu dikey ileri dalgacık dönüşümü uygulaması, dikey halin açık bir genişlemesidir.

## 9. SONUÇLAR VE ÖNERİLER

### 9.1 Dalgacık Seçimi

Binary ölçekli fonksiyonların dyadic çevirilerinin dikey olduğu durumlarda, ideal temel dalgacık, kısa süreli titreşimli bir fonksiyon olur. Haar fonksiyonu bu olayı açıklamaktadır.

Bir temel dalgacık seçimi, genellikle uygulama ile belirlenir. Örneğin kayıpsız sıkıştırma işleminde, amaç fonksiyonu tamamen göstermek olduğu için, dikey ve bidikey temel gerekebilir. Öte yandan eğer hedeflenen kayıpsız sıkıştırma, bir görüntü içindeki kenarlar gibi özel bileşenlerin bulunması ise bileşenler ile aynı bir dalgacık seçmek daha önemlidir.

Dalgacık dönüşümleri, compact gösterim beklentisini ve seçilen dalgacığın dalga şekline uygun görüntü bileşenlerinin bulunmasını sağlar. Dikey dalgacık dönüşümü, doğal olarak compacttır. Çünkü görüntü bileşenlerinin ufak değişimleri altında iyi hareket etmez. Dalgacığa uyumlu bir görüntü, dönüşüm içinde tam olarak görülecektir. Bu nedenle, ortanormal olmayan dönüşümler, ortaya çıkarma işlemine daha iyi uyum gösterirler.

### 9.2 Sonuçlar ve Uygulama

Uygulamada da görülebileceği gibi dalgacık dönüşümü kayıplı bir sıkıştırma tekniğidir. Görüntünün yoğunluğuna göre kayıp oranı değişmektedir. Tabii ki uygulanan iterasyon sayısı ve sıkıştırma oranı da, yeniden oluşturma işlemi sonrasında ortaya çıkan görüntüdeki kaybı etkilemektedir.

Tüm bunlara rağmen veri ve görüntü saklama çalışmalarında, sıkıştırma esnasında gerçekleşen bu kayıplar, ihmal edilebilecek düzeyde gerçekleşebilir. Görüntünün yoğunluğu ve dosya uzantısına göre uygulanacak teknikler üzerinde karar verilmelidir.



(a)



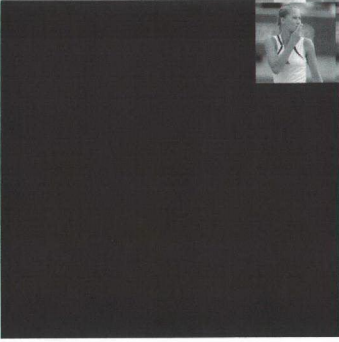
(b)



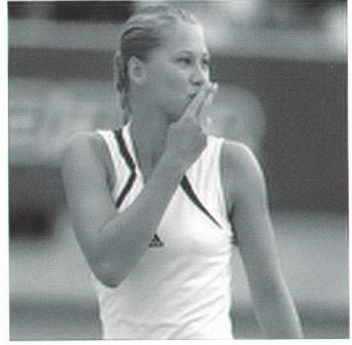
(c)



(d)



(e)



(f)

Şekil 9.1 Görüntü sıkıştırma örneği, Anna Kournikova

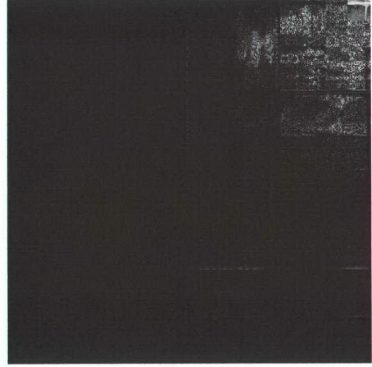
(a) 512 x 512 BMP orjinal görüntü (b) İki iterasyon ile ileri dönüşüm

(c) %25 sıkıştırma oranı ile (d) Ters dönüşüm

(e) %6.25 sıkıştırma oranı ile (f) Ters dönüşüm



(a)



(b)



(c)



(d)



(e)



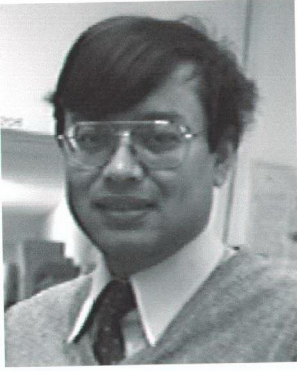
(f)

Şekil 9.2 Görüntü sıkıştırma örneği, parmak izi

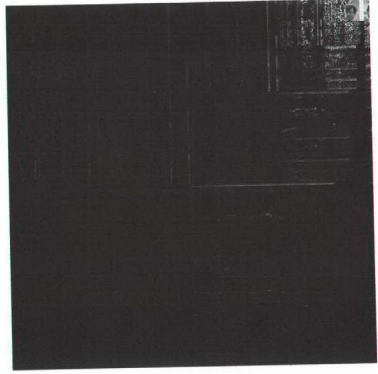
(a) 512 x 512 BMP orjinal görüntü (b) Dört iterasyon ile ileri dönüşüm

(c) %25 sıkıştırma oranı ile (d) Ters dönüşüm

(e) %6.25 sıkıştırma oranı ile ters dönüşüm (f) %3,82 sıkıştırma oranı ile ters dönüşüm



(a)



(b)



(c)



(d)

Şekil 9.3 Görüntü sıkıştırma örneği, Akhan

(a) 512 x 512 BMP orjinal görüntü (b) Dört iterasyon ile ileri dönüşüm

(c) %25 sıkıştırma oranı ile (d) Ters dönüşüm

**KAYNAKLAR**

- Marr, D., (1982) "In Vision", W.H.Freeman and Company
- Grossman, A., Morlet, J., (1984), "Decomposition of Hard Functions into Square Integrable Wavelets of Constant Shape", Siam J.Math, 723-736
- Daubechies, I., (1996), "Orthonormal Basis of Compactly Supported Wavelets", In Pure and Applied Math.,böl 7, s 909-996
- Castleman, Kenneth R., (1996), "Digital Image Processing", 303-350, 401-446
- Bradley, J., Brislawn, C., (1992) "The Wavelet/Scalar Quantization Compression Standart for Digital Fingerprint Images", Technical Report LA-UR-94-4-1409
- Mallat, S., Zhong, S., (1991), "Wavelet Transform Maxima and Multiscale Edges", Technical Report, University of Pennsylvania
- Canny, J., (1986), "A Computational Approach to Edge Detection", IEEE Transaction on Pattern Analysis and Machine Intelligence
- Youla, D., Webb, H., (1982), "Image Restoration by the Method of Convex Projections", IEEE Trans. On Medical Imaging
- Lim, J.S., (1990), "Two Dimensional Signal and Image Processing", Prentice Hall
- L Mallat, S., (1989), "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans.
- Burt, P.J., Adelson E.H., (1983), "The Laplacian Pyramid as a Compact Image Code", IEEE Trans.
- Woods, J.W., O'Neill, S.D., (1986), "Subband Coding of Images", IEEE Trans.
- Esteban, D., Galand, C., (1977), "Application of Quarature Mirror Filters to Split-Band Voice Coding Systems", International Conf. On Acoustics, Speech and Signal Processing

**ÖZGEÇMİŞ**

|               |            |                                                                                     |
|---------------|------------|-------------------------------------------------------------------------------------|
| Doğum tarihi  | 11.03.1976 |                                                                                     |
| Doğum yeri    | İzmir      |                                                                                     |
| Lise          | 1989-1992  | Gölcük Barbaros Hayrettin Lisesi                                                    |
| Lisans        | 1992-1996  | Yıldız Teknik Üniversitesi Mühendislik Fak.<br>Matematik Mühendisliği Bölümü        |
| Yüksek Lisans | 1996-      | Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü<br>Matematik Müh. Ana Bilim Dalı |

**Çalıştığı kurum**

1998-Devam ediyor Bluechip Turizm ve Organizasyon

