

**CURVELET DÖNÜŐÜMÜ KULLANILARAK  
SNAKE ALGORTİMASI İLE  
İMGEDA KENAR ALGILAMA**

**F.Saba AHİSHALI**

**Y.Lisans Tezi  
Elektrik ve Elektronik Mühendisliđi Anabilim Dalı  
Yrd. Doç. Dr. Emin Argun ORAL  
2012  
Her hakkı saklıdır**

ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Y.LİSANS TEZİ

CURVELET DÖNÜŞÜMÜ KULLANILARAK SNAKE ALGORTİMASI  
İLE İMGEDE KENAR ALGILAMA

F.Saba AHİSHALI

ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ERZURUM

2012

Her hakkı saklıdır.



T.C.  
ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



TEZ ONAY FORMU

Curvelet Dönüşümü Kullanılarak Snake Algoritması İle İmgede Kenar Algılama

Yrd. Doç. Dr. Emin Argun ORAL danışmanlığında, F.Saba AHISHALI tarafından hazırlanan bu çalışma 14/08/2012 tarihinde aşağıdaki jüri tarafından Elektrik-Elektronik Ana Bilim Dalı'nda Yüksek Lisans Tezi olarak oybirliği/oy çokluğu (.../...) ile kabul edilmiştir.

Başkan : Yrd. Doç. Dr. Bülent ÇAVUŞOĞLU

İmza : 

Üye : Yrd. Doç. Dr. Yenal VANGÖLÜ

İmza : 

Üye : Yrd. Doç. Dr. Emin Argun ORAL

İmza : 

Yukarıdaki sonucu onaylıyorum

Prof. Dr. İhsan EFEOĞLU  
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

## ÖZET

Y.Lisans Tezi

### CURVELET DÖNÜŞÜMÜ KULLANILARAK SNAKE ALGORTİMASI İLE İMGEDE KENAR ALGILAMA

F.Saba AHISHALI

Atatürk Üniversitesi  
Mühendislik Fakültesi  
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Emin Argun ORAL

Çoklu çözünürlük analizi, görüntü temel doku yapısı üzerinde yoğunlaşma sağlamaktadır. Çoklu çözünürlük analizinin son on yılda ulaştığı nokta ise Curvelet dönüşümüdür. Curvelet dönüşümü diğer çoklu çözünürlük dönüşümlerinden farklı olarak görüntüyü ölçek, konum ve açı parametreleri ile inceleyerek görüntü hakkında daha fazla bilgi edinilmesini sağlamaktadır.

Görüntü bölütleme tekniklerinden geleneksel Snake algoritması; kenar bulma, stereo görüntü işleme işlemlerinde kullanılmaktadır. Dudak okuma, trafikte hız kontrolü, bir görüntünün içindeki nesnenin sınırlarının bulunması gibi uygulamalarda kullanılabilen bir algoritmadır. Yakaladığı noktanın izini sürebilme özelliği vardır. Ancak bu algoritmanın doğru çalışabilmesi için algoritmaya referans olarak verilen başlangıç noktalarının görüntüdeki ilgilenilen nesnenin sınırları dışından başlatılması gerekmektedir. Ayrıca algoritma içbükey noktalarda yetersiz çalışmaktadır.

Bu çalışmada, önce Curvelet dönüşümü kullanılarak görüntülerin Curvelet katsayıları hesaplandı. Sonra en yüksek ölçekte bu katsayılardan görüntünün kenarlarına ait olan katsayılar belirlendi. Son olarak bu katsayılar farklı snake algoritmalarını işletmede başlangıç noktalarını otomatik olarak belirlemek ve geleneksel Snake algoritmasının performansını geliştirmek için kullanıldı. Uygulanılan yöntem dört ayrı imge üzerinde test edildi. Ayrıca geliştirilen yöntemin performansı geleneksel Snake algoritması ve kenar izleme konusunda oldukça başarılı Gradyan Vektör Akış algoritmasıyla karşılaştırıldı.

**2012, 87 sayfa**

**Anahtar Kelimeler:** Dalgacık, Ridgelet, Curvelet, Snake, GVF

## **ABSTRACT**

MS Thesis

### **IMAGE EDGE DETECTION WITH SNAKE ALGORITHM BY USING CURVELET TRANSFORM**

F.Saba AHISHALI

Atatürk University  
Faculty of Engineering  
Department of Electric and Electronic Engineering

Supervisor: Asst. Prof. Dr. Emin Argun ORAL

Multi Resolution Analysis (MRA) enables concentration on the fundamental structure of an image. Its final progress in the last decade is the Curvelet Transformation. Curvelet Transformation gives more detailed information about an image by analyzing it not only in terms of scale and translation but also rotational angle.

Traditional Snake algorithm, one of the available image segmentation techniques, is used in edge detection, stereo image processing procedures. It is an algorithm that can be used in applications such as lip-reading, speed control at traffic, edge detection of an object with in an image and etc. It is capable of tracing of a point that is captured. But it is necessary to start the algorithm with the initial points that are all located at the outside of the object of interest for the algorithm to perform correctly. Furthermore, its algorithm is insufficient at the concave geometries.

In this study, Curvelet coefficients of images are first calculated by using Curvelet Transformation. Then, ones at the highest scale corresponding to the edges in the images are located. Finally, they are used to define the initial starting points in an automatic manner to run various Snake algorithms and increase the performance of the Traditional Snake algorithm. This new method has been tested on four different test images. Also, its performance has been compared with the Traditional Snake algorithm and Gradient Vector Flow (GVF) algorithm, a very effective procedure in edge detection.

**2012, 87 pages**

**Keywords:** Wavelet, Ridgelet, Curvelet, Snake, GVF

## TEŐEKKÜR

Yüksek lisans eğitimim ve bu tez çalışmamda beni her konuda yönlendiren, ilgisini ve desteğini esirgemeyen değerli danışmanım Sayın Yrd. Doç. Dr. Emin Argun ORAL'a en içten teşekkür ve saygılarımı sunarım.

Yüksek Lisans eğitim boyunca burs desteği sağlayan TÜBİTAK'a, yardımlarını esirgemeyen dostlarıma ve beni bu günlere getiren aileme teşekkür ederim.

F.Saba AHISHALI

Ağustos, 2012

## İÇİNDEKİLER

ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
SİMGELER ve KISALTMALAR DİZİNİ .....	vi
ŞEKİLLER DİZİNİ .....	viii
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KURAMSAL TEMELLER.....</b>	<b>3</b>
2.1. Dalgacık Dönüşümü .....	3
2.1.1. Fourier dönüşümü.....	3
2.1.2. Kısa-zaman fourier dönüşümü .....	4
2.1.3. Dalgacık dönüşümü .....	5
2.1.3.a. Sürekli dalgacık dönüşümü .....	7
2.1.3.b. Ayrık dalgacık dönüşümü .....	9
2.1.3.c. Dalgacık paket analizi .....	12
2.1.3.d. Dalgacık özellikleri ve seçimi .....	12
2.1.3.e. Çok kullanılan ana dalgacık çeşitleri ve formülleri .....	14
2.1.3.f. İki boyutlu dalgacık dönüşümü .....	19
2.1.3.g. Çoklu-dalgacık dönüşümü.....	21
2.2. Literatürde Seyrek Geometrik Görüntü Gösterimi .....	22
2.2.1. Contourlet .....	22
2.2.2. Surfacelets .....	23
2.2.3. Directionlets .....	23
2.2.4. Bandlets .....	24
2.2.5. Grouplets .....	25
2.2.6. Shearlets .....	25
2.3. Ridgelet Dönüşümü .....	25
2.3.1. Sürekli ridgelet dönüşümü.....	26
2.3.2. Sonlu ridgelet dönüşümü .....	29
2.4. Curvelet Dönüşümü.....	31

2.4.1. Birinci nesil curvelet dönüşümü.....	33
2.4.1.a. Alt bant ayrıştırma .....	34
2.4.1.b. Yumuşak bölümlenme.....	36
2.4.1.c. Normalizasyon.....	38
2.4.1.d. Ridgelet analizi.....	38
2.4.2. Birinci nesil ters curvelet dönüşümü .....	40
2.4.2.a. Ridgelet sentezi.....	40
2.4.2.b. Normalizasyon.....	40
2.4.2.c. Smooth tümleştirme.....	40
2.4.2.d. Alt Bant birleştirme .....	41
2.4.3. İkinci nesil curvelet dönüşümü.....	41
2.4.3.a. 2-D curvelet dönüşümü .....	42
2.4.3.b. 2-D ayrık curvelet dönüşümü .....	45
2.5. Görüntü Bölütlenme .....	47
2.5.1. Görüntü bölütlenme teknikleri .....	48
2.5.1.a. Eşikleme .....	48
2.5.1.b. Bileşen etiketleme .....	50
2.5.1.c. Nokta süreksizliklerinin belirlenmesi .....	51
2.5.1.d. Çizgi süreksizliklerinin belirlenmesi.....	52
2.5.1.e. Kenar süreksizliklerinin belirlenmesi .....	53
2.5.1.f. Bölge tabanlı bölütlenme .....	54
2.5.1.g. Snake modeli .....	55
2.5.1.h. Gradyent vektör akışı .....	56
<b>3. MATERYAL ve YÖNTEM.....</b>	<b>58</b>
<b>4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....</b>	<b>61</b>
<b>5. SONUÇ .....</b>	<b>85</b>
KAYNAKLAR .....	86
ÖZGEÇMİŞ .....	88



## SİMGELER ve KISALTMALAR DİZİNİ

$\alpha$	Gerilim (Tension) parametresi
$\beta$	Katılık-sertlik (Rigidity) parametresi
$c$	Çoklu çözünürlük katsayıları
$E$	Enerji
$g[n]$	Yüksek geçiren filtre
$h[n]$	Açak geçiren filtre
$j$	Ölçek parametresi
$k$	Öteleme parametresi
$l$	Rotasyon parametresi
$\psi$	Dalgacık fonksiyonu
$\phi$	Curvelet fonksiyonu
$P$	Histogram
$\mathcal{R}_f$	Radon dönüşümü
$R_\theta$	Rotasyon matrisi
$S_\theta$	Shear (Dilimleme) matrisi

### Kısaltmalar

A	Yaklaşımlar (Approximations)
ADD	Ayrık Dalgacık Dönüşümü
CRT	Sürekli Ridgelet Dönüşümü
CurveletG1	Birinci Nesil Curvelet Dönüşümü
CurveletG2	İkinci Nesil Curvelet Dönüşümü
CWT	Sürekli Dalgacık Dönüşümü (Continuous Wavelet Transform)
D	Detaylar (Details)
DCTG1	Birinci Nesil Ayrık Curvelet Dönüşümü
DCTG2	İkinci Nesil Ayrık Curvelet Dönüşümü
DWT	Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform)

FFT	Hızlı Fourier Dönüşümü
IFFT	Ters hızlı Fourier Dönüşümü
FRAT	Sonlu Radon Dönüşümü
FRIT	Sonlu Ridgelet Dönüşümü
GVF	Gradyent Vektör Akışı (Gradient Vector Flow)
KZFD	Kısa Zaman Fourier Dönüşümü
M-DIR	Çok Yönlü (Multi-Directional)
MRA	Çoklu Çözünürlük Analizi (Multi-Resolution Analysis)
SDD	Sürekli Dalgacık Dönüşümü
SRD	Sürekli Ridgelet Dönüşümü
WT	Dalgacık Dönüşümü

## ŞEKİLLER DİZİNİ

Şekil 2.1. Fourier dönüşümü .....	4
Şekil 2.2. Kısa-zaman Fourier dönüşümü.....	5
Şekil 2.3. Dalgacık dönüşümü .....	6
Şekil 2.4. Dalgacık dönüşümünün temel teorisi .....	9
Şekil 2.5. ADD uygulamasının alt bantlara ayrışımı .....	10
Şekil 2.6. Standart dalgacık ayrıştırma ağacı.....	11
Şekil 2.7. Dalgacık paket analizi .....	12
Şekil 2.8. Meksika şapkası dalgacığı fonksiyonu .....	14
Şekil 2.9. Meyer dalgacığı fonksiyonu .....	15
Şekil 2.10. Morlet dalgacığı fonksiyonu.....	16
Şekil 2.11. Gauss dalgacığı fonksiyonu .....	16
Şekil 2.12. Haar dalgacığı fonksiyonu .....	17
Şekil 2.13. Daubechies 3 dalgacığı fonksiyonu .....	18
Şekil 2.14. Daubechies 4 dalgacığı fonksiyonu .....	18
Şekil 2.15. Daubechies 5 dalgacığı fonksiyonu .....	19
Şekil 2.16. İki boyutlu Dalgacık dönüşümü.....	21
Şekil 2.17. İki boyutlu çoklu-Dalgacık dönüşümü ayrışımı .....	21
Şekil 2.18. Koordinat düzleminden Radon bölgesine geçiş .....	28
Şekil 2.19. FRIT diyagramı .....	31
Şekil 2.20. Dalgacık ve Curvelet dönüşümleri arasındaki karşılaştırılması .....	32
Şekil 2.21. Bir tek alt bandın uzamsal ayrışması .....	33
Şekil 2.22. Birinci nesil ayırık Curvelet dönüşümünün akış diyagramı .....	34
Şekil 2.23. Alt bant ayrıştırması .....	36
Şekil 2.24. Smooth bölümlene .....	37
Şekil 2.25. Birinci nesil Curvelet dönüşümü adımları.....	39
Şekil 2.26. Birinci nesil ters Curvelet dönüşümü adımları .....	41
Şekil 2.27. Frekans bölgesinde curvelet yapısı.....	44
Şekil 2.28. Frekans düzleminde ayırık Curvelet dönüşümü .....	45
Şekil 2.29. $\phi_{4,k,l}$ fonksiyonunun desteklediği alan .....	47

<b>Şekil 2.30.</b> İkili eşikleme .....	49
<b>Şekil 2.31.</b> Çoklu eşikleme .....	50
<b>Şekil 2.32.</b> Nokta arama maskesi .....	52
<b>Şekil 2.33.</b> Çizgi arama maskeleri .....	52
<b>Şekil 2.34.</b> En çok kullanılan filtreler; Roberts, Prewitt, Sobel .....	53
<b>Şekil 4.1.</b> Testimge_1 orijinal görüntü (480×480) .....	63
<b>Şekil 4.2.</b> Testimge_1 geleneksel Snake algoritmasında Snake hareketi .....	64
<b>Şekil 4.3.</b> Testimge_1 geleneksel Snake algoritması gösterimi .....	64
<b>Şekil 4.4.</b> Testimge_1 GVF algoritmasında Snake hareketi .....	65
<b>Şekil 4.5.</b> Testimge_1 GVF algoritması gösterimi .....	65
<b>Şekil 4.6.</b> Testimge_1 Curvelet katsayıları .....	66
<b>Şekil 4.7.</b> Testimge_1 Curvelet katsayıları uygulandığında Snake hareketi .....	66
<b>Şekil 4.8.</b> Testimge_1 Curvelet katsayıları ile Snake algoritması gösterimi .....	67
<b>Şekil 4.9.</b> Testimge_1 iterasyon adımları için kenar algılama hata grafiği .....	67
<b>Şekil 4.10.</b> Testimge_2 orijinal görüntü (319×309) .....	68
<b>Şekil 4.11.</b> Testimge_2 Geleneksel Snake algoritmasında Snake hareketi .....	68
<b>Şekil 4.12.</b> Testimge_2 Geleneksel Snake algoritması gösterimi .....	69
<b>Şekil 4.13.</b> Testimge_2 GVF algoritmasında Snake hareketi .....	69
<b>Şekil 4.14.</b> Testimge_2 GVF algoritması gösterimi .....	70
<b>Şekil 4.15.</b> Testimge_2 Curvelet katsayıları .....	70
<b>Şekil 4.16.</b> Testimge_2 Curvelet katsayıları uygulandığında Snake hareketi .....	71
<b>Şekil 4.17.</b> Testimge_2 Curvelet katsayıları ile Snake algoritması gösterimi .....	71
<b>Şekil 4.18.</b> Testimge_2 iterasyon adımları için kenar algılama hata grafiği .....	72
<b>Şekil 4.19.</b> Testimge_3 orijinal görüntü (512×512) .....	72
<b>Şekil 4.20.</b> Testimge_3 geleneksel Snake algoritmasında Snake hareketi .....	73
<b>Şekil 4.21.</b> Testimge_3 geleneksel Snake algoritması gösterimi .....	73
<b>Şekil 4.22.</b> Testimge_3 GVF algoritmasında Snake hareketi .....	74
<b>Şekil 4.23.</b> Testimge_3 GVF algoritması gösterimi .....	74
<b>Şekil 4.24.</b> Testimge_3 Curvelet katsayıları .....	75
<b>Şekil 4.25.</b> Testimge_3 Curvelet katsayıları uygulandığında Snake hareketi .....	75
<b>Şekil 4.26.</b> Testimge_3 Curvelet katsayıları ile Snake algoritması gösterimi .....	76
<b>Şekil 4.27.</b> Testimge_3 iterasyon adımları için kenar algılama hata grafiği .....	76

<b>Şekil 4.28.</b> Testimge_4 orijinal görüntü (512×512) .....	77
<b>Şekil 4.29.</b> Testimge_4 geleneksel Snake algoritmasında Snake hareketi .....	77
<b>Şekil 4.30.</b> Testimge_4 geleneksel Snake algoritması gösterimi.....	78
<b>Şekil 4.31.</b> Testimge_4 GVF algoritmasında Snake hareketi .....	78
<b>Şekil 4.32.</b> Testimge_4 GVF algoritması gösterimi .....	79
<b>Şekil 4.33.</b> Testimge_4 Curvelet katsayıları .....	79
<b>Şekil 4.34.</b> Testimge_4 Curvelet katsayıları uygulandığında Snake hareketi .....	80
<b>Şekil 4.35.</b> Testimge_4 Curvelet katsayıları ile Snake algoritması gösterimi .....	80
<b>Şekil 4.36.</b> Testimge_4 iterasyon adımları için kenar algılama hata grafiği .....	81
<b>Şekil 4.37.</b> Testimge_4 geleneksel Snake algoritmasında Snake hareketi .....	81
<b>Şekil 4.38.</b> Testimge_4 geleneksel Snake algoritması gösterimi.....	82
<b>Şekil 4.39.</b> Testimge_4 GVF algoritmasında Snake hareketi .....	82
<b>Şekil 4.40.</b> Testimge_4 GVF algoritması gösterimi .....	83
<b>Şekil 4.41.</b> Testimge_4 Curvelet katsayıları uygulandığında Snake hareketi .....	83
<b>Şekil 4.42.</b> Testimge_4 Curvelet katsayıları ile Snake algoritması gösterimi .....	84
<b>Şekil 4.43.</b> Testimge_4 iterasyon adımları için kenar algılama hata grafiği .....	84

## 1. GİRİŞ

Görüntü bir nesnenin bilgilerinin görüntüsel olarak saklanması ve sunumudur. Görüntü işleme bir görüntünün özelliklerinin değiştirilmesi sonucunda yeni bir görüntü elde etmek için kullanılan teknikler bütünüdür. Görüntü işleme teknikleri görüntünün insan veya bilgisayar tarafından anlaşılabilmesi veya yorumlanabilmesini sağlamak için kullanılmaktadır.

Görüntü işlemede dönüşüm kavramı, bir sinyalin bir bölgede belirgin olmayan özelliklerini ortaya çıkarmak için başka bir bölgeye taşınması işlemidir. Birçok sinyal zamanın bir fonksiyonu ile ifade edilmektedir. Zaman tanım alanında verilmiş bir sinyal, herhangi bir matematiksel dönüşüm uygulanarak işlenebilmektedir. Fakat çoğu sinyal işleme uygulamaları için zaman bölgesindeki gösterim yeterli olmamaktadır. Çünkü çoğu durumda sinyaldeki önemli bilgi sinyalin frekans içeriğinde saklanmaktadır. Bu yüzden sinyal işleme açısından sinyallerin frekans spektrumunu elde etmek önemlidir. Frekans spektrumu, sinyalin içerdiği farklı frekans bileşenlerini ve bu frekanslara ait genlik büyüklüklerini içermektedir. Dolayısıyla sinyal işlemede dönüşümün amacı zaman bölgesinde verilmiş sinyali frekans bölgesine çevirmektir (Yakar 2006). Örneğin; bir imgeler frekans uzayında parlaklık veya renk değişimlerine bağlı olarak gösterilirler.

Farklı frekanslar için farklı çözünürlüklerde analiz fikrine çoklu çözünürlük analizi (Multi-resolution analysis - MRA) adı verilmektedir. Çoklu çözünürlük analizi bir görüntünün çözünürlüğünün veya bulanıklığının belirli seviyelerine göre korunmasını sağlamaktadır. Genel anlamda, çok çözünürlüklü analiz temel doku yapısı üzerinde yoğunlaşma (zoom in out ) sağlamaktadır. Bu nedenle, doku çıkartılmasıyla komşu pikselin boyutu etkilenmez (Dettori and Semler 2007).

Görüntü işlemede özellikle görüntü bölütlemeye MRA başarıyla kullanılmaktadır. Dalgacık tabanlı özellikler görüntü sıkıştırma, gürültü giderme (denosing) ve sınıflandırma gibi çeşitli uygulamalarda kullanılmaktadırlar. Son zamanlarda, sonlu Ridgelet ve Curvelet dönüşümleri yüksek boyutlu MRA aracı olarak literatürde tanıtılmaktadır.

Bu çalışmada, kenar algılamada etkili bir yöntem geliştirilmeye çalışıldı. Bu amaçla kenar algılama işlemi çoklu çözünürlük analizi ve bir bölütleme tekniği kullanılarak iki aşamada gerçekleştirildi. Bu amaçla, bir çoklu çözünürlük analizi olan Curvelet dönüşümü bir görüntü bölütleme tekniği olan snake algoritmasıyla birleştirildi. Bu yeni yaklaşımda Snake algoritmasının performansı artırıldı ve kenarlar başarıyla belirlendi.

## 2. KURAMSAL TEMELLER

### 2.1. Dalgacık Dönüşümü

#### 2.1.1. Fourier dönüşümü

Sinyal analizinde en çok kullanılan ve bilinen yöntem, Fransız bilim adamı olan Jean Baptise Joseph Fourier tarafından bulunan Fourier dönüşümüdür. Fourier dönüşümü (Şekil 2.1) bir sinyali farklı frekanslardaki sinüs bileşenlerine ayrıştırmaktadır. Bir  $f(t)$  giriş sinyalinin Sürekli-zaman Fourier dönüşümü denklem 2.1. ile tanımlanmaktadır.

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.1.)$$

Ters Sürekli-zaman Fourier dönüşümü denklem 2.2. ile tanımlanmaktadır.

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega)e^{j\omega t} d\omega \quad (2.2.)$$

Birçok sinyal için Fourier dönüşümü oldukça kullanışlıdır. Ancak, dönüşümün ciddi bir sakıncası vardır. Fourier dönüşümü sadece frekans bilgisini sağlar, yani sinyal frekans ortamına dönüştürüldüğünde zaman bilgisi kaybolmaktadır. Bir sinyalin Fourier dönüşümüne bakıldığında, bir olayın ne zaman gerçekleştiğine dair bir şey söylemek imkansızdır. Eğer sinyalin özelliği zamanla değişmiyorsa, yani sinyal durağan (stationary) ise zaman bilgisinin kaybolması önemli olmamaktadır. Ancak sinyal durağan olmayan ve keskin geçişlerin olduğu karakterdeyse, ani geçişler ile olayın başlangıç ve bitiş yerleri sinyalin en önemli bilgilerini içermektedir ve Fourier dönüşümü bunları algılamak için uygun olmamaktadır.





**Şekil 2.1.** Fourier dönüşümü (Günel 2008)

### 2.1.2. Kısa-zaman fourier dönüşümü

Fourier dönüşümünün zaman bilgisi eksikliğini gidermek için, Dennis Gabor (1946) sinyalin zaman alanında küçük pencereler halinde analiz edebileceği fikrini ortaya atmıştır. Bu yönteme, Kısa-zaman Fourier dönüşümü (KZFD - Short-time Fourier transform - STFT) denir. KZFD bir sinyalin zaman ve frekans bölgesinde eşzamanlı bilgisini vermektedir. Yani sinyalin ne zaman ve hangi frekansla oluştuğu hakkında bilgi vermektedir (Polikar 1999).

Bir  $f(t)$  sinyalinin Kısa-zaman Fourier dönüşümü denklem 2.3. ile tanımlanmaktadır.

$$f(\tau, \omega) = \int_{-\infty}^{\infty} f(t)w(t - \tau)e^{-j\omega t} dt \quad (2.3.)$$

Burada  $w(t)$  uygun seçilmiş bir pencere fonksiyonu olup, sonlu enerjiye sahip olması ve integralinin alınabilmesi gerekmektedir. Pencere fonksiyonu ile giriş sinyali belirli zaman bölgelerine ayrılmakta ve her bölge kendi frekans içerikleri için ayrı olarak analiz edilmektedir. Ayrıca eğer sinyal keskin geçişlere sahip ise giriş sinyalini sonlanma noktalarında sıfır olacak şekilde pencerelemektedir. Bu pencereleme sonlanma noktalarını ortadaki noktalardan daha az vurgulayacak bir ağırlık fonksiyonuyla gerçekleştirilmektedir.



**Şekil 2.2.** Kısa-sürekli Fourier dönüşümü (Günel 2008)

KZFD'nin en önemli sorunu sabit pencere boyutuna sahip olmasıdır (Şekil 2.2). Bu durumda elde edilen bilgiler sınırlı doğruluktur. Çünkü doğruluk pencerenin boyutuyla ilgilidir. KZFD'nin sinyalin zaman ve frekans bilgilerini içermesi önemini artırmasına rağmen ama sabit zaman-frekans çözünürlüğü ile yapılan analizde sinyallerin, yerel noktalardaki analizi, çözünürlüğü sınırlı olmaktadır. Durağan olmayan sinyaller bazı zaman dilimlerinde daha hassas yaklaşıma dolayısıyla değişen pencereleme tekniklerine ihtiyaç duymaktadır.

### 2.1.3. Dalgacık dönüşümü

Dalgacık dönüşümü (Wavelet transform), durağan olmayan sinyaller için gerekli olan hassasiyeti sağlamak için kullanılan bir yaklaşımdır. Fourier dönüşümünde olduğu gibi sinyali küçük parçalara ayırmaktadır. Ancak, Fourier dönüşümü ayrıştırma işlemi için sonsuz uzunlukta olduğu varsayılan ve değişik frekanslardaki düzenli sinüs dalgalarını kullanmaktayken, Dalgacık dönüşümü “ana dalgacık” adı verilen sınırlı süreli, düzensiz ve asimetrik sinyallerin, ölçeklenmiş ve kaydırılmış tekrarlarını kullanmaktadır. Sinyallerdeki kısa süreli ve keskin değişiklikler; pürüzsüz ve düzgün bir sinüs dalgasına göre düzensiz bir dalgacık ile daha iyi analiz edilmektedir (Mallat 1999).

Bir  $f(t)$  sinyalinin Dalgacık dönüşümü denklem 2.4. ile tanımlanmaktadır.

$$C(\text{ölçek}, \text{konum}) = \int_{-\infty}^{\infty} f(t)\psi(\text{ölçek}, \text{konum}) dt \quad (2.4.)$$

Burada, “ $\psi$ ” dalgacık fonksiyonunu, “C” buna karşılık gelen dalgacık katsayılarını ifade etmektedir.

Dalgacık dönüşümü; sinyali farklı frekans bileşenlerine ayıran ve her bir bileşeni o ölçekteki çözünürlüğüyle inceleyen bir dönüşüm tekniğidir. Zaman analizi için, örnek dalgacığın sıkıştırılmış, yüksek frekans şekli uygulanmaktadır. Frekans analizi için, aynı dalgacığın genişletilmiş, alçak frekans şekli uygulanmaktadır. Ana dalgacığa uygulanan kaydırma ve ölçekleme işlemleri, sinyalin yerel bölgeleri ile dalgacık arasındaki karşılıklı ilişkiyi temsil edecek olan katsayıları belirlemede kullanılmaktadır. Eğer veriye uygun en iyi dalgacıklar seçilirse veya bir başlangıç değerinin altında kalan katsayılar atılırsa, veri kısmen temsil edilmiş olmaktadır. Bu kısmi kodlama, veri sıkıştırma alanında dalgacıkları mükemmel bir araç yapmaktadır.



**Şekil 2.3.** Dalgacık dönüşümü (Günel 2008)

Dalgacık dönüşümü (Şekil 2.3), farklı pencereleme fonksiyonları (ölçeklenebilir pencereler) ve asimetrik dalgacıklar kullandığı için sinyali tam olarak betimleyebilmektedir. Böylece sinyal hakkında ayrıntılı bilgiye sahip olunmaktadır. Pencere sinyal boyunca kaydırılarak her pozisyonu için spektrumu yeniden hesaplanmaktadır. Ve bu işlem daha uzun veya kısa pencerelerle pozisyon döngüleri için tekrarlanmaktadır. Sonuçta dönüşüm, zaman-frekans temsillerinin toplamı olarak bulunmaktadır. Bu temsillerin toplamı, farklı uzunluktaki pencerelerden oluştuğundan çoklu çözünürlük analizi olarak adlandırılmaktadır.

Dalgacık dönüşümünde sinyal, zaman-frekans yerine zaman-ölçek ortamında incelenir ve en önemli üstünlüklerinden biri; sinyali belirli bir bölgesinin analizinin mümkün olmasıdır. Bu şekilde tanımlı ölçek tabanlı analizin, gürültüye karşı daha az hassasiyet göstermesi de ayrı bir avantaj sağlamaktadır. Dalgacık dönüşümü, başka sinyal işleme yöntemlerinin yakalayamadığı bozulma noktalarını, süreksizlikleri ve benzerlikleri çıkarmada böylece başarılı olmaktadır.

Dalgacık dönüşümü yüksek frekanslarda iyi zaman çözünürlüğü ve kötü frekans çözünürlüğü, alçak frekanslarda ise iyi frekans çözünürlüğü ve kötü zaman çözünürlüğü elde edilecek şekilde tasarlanmaktadır. Bunun için pencere genişlikleri yüksek frekanslarda dar, alçak frekanslarda ise geniş tutulmaktadır.

Öte yandan tarihsel olarak, dalgacıklar matematik, kuantum fiziği, elektrik mühendisliği ve sismik jeoloji alanlarında bağımsız olarak geliştirilmekteydi. Farklı disiplinlerin bir araya gelmesi ve bu alanlardaki bilgi değiş tokuşuyla; astronomi, alt bant kodlama, sinyal ve görüntü işleme, resim sıkıştırma (FBI'da ki parmak izlerinin sıkıştırılması vb), biyomedikal, ses ayırma, fraktaller, türbülans, deprem tahmini gibi yeni dalgacık uygulamalarını ortaya çıkarmaktadır (Polikar 1999; Selesnick 2007). Dalgacık dönüşümünde, sinyal, dalgacık ölçekleri ile gösterilir, bu ölçek ile gösterim daha küçük olup veri sıkıştırma, hızlı veri iletimi, gürültüden arındırma, sinyal onarımı ve hızlı sayısal algoritma uygulamaları için anahtar teşkil etmektedir (Jung Jun *et al.* 1999).

### 2.1.3.a. Sürekli dalgacık dönüşümü

Sürekli Dalgacık dönüşümü (SDD - Continuous Wavelet transform - CWT) denklem 2.5. ile gösterildiği şekilde sinyalin her ölçek ve pozisyonundaki dalgacık fonksiyonu ile çarpımlarının toplanması olarak tanımlanmaktadır.

$$C(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (2.5.)$$

$\psi(t)$  ana dalgacık (mother wavelet) fonksiyonu,  $a$  ölçek ve  $b$  öteleme parametresi olmak üzere; olmak üzere,  $\psi(\frac{t-b}{a})$  ana dalgacıktan türetilen dalgacık fonksiyonunu,  $C$  ölçek ve konumuna bağlı bulunan Dalgacık dönüşümü katsayıları temsil etmektedir.

$a$  ve  $b$  parametrelerinin değerlerine bağlı olarak ana dalgacık fonksiyonu;

- $a > 1$  olduğunda, zaman ekseninde genişler (stretch) ve genliği düşer.
- $a < 1$  olduğunda, zaman ekseninde daralır (squeeze) ve genlik büyür.
- $a < 0$  olduğunda,  $t=0$  noktasına göre simetriği alınır.
- $b > 0$  olduğunda, zaman ekseninde sağa doğru ötelemektir.
- $b < 0$  olduğunda, zaman ekseninde sola doğru ötelemektir.

Dönüşümde kullanılacak olan fonksiyonlar ana dalgacık tarafından üretildiği için, Dalgacık dönüşümünün karakteristiğini ana dalgacık belirlemektedir. Bu yüzden uygulamalarda ana dalgacık türünün seçilmesi büyük önem taşımaktadır.

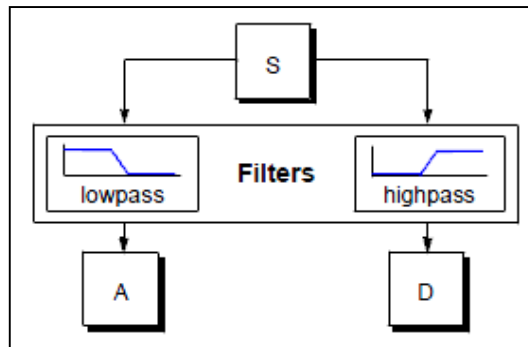
Düşük ölçeklerde dalgacık daralır ve yüksek frekanslar (ani değişimler) yakalanırken, yüksek ölçeklerde ise dalgacık genişleyerek düşük frekanslar yakalanmaktadır. SDD, bilgisayar ile hesaplama yapılırken doğal olarak ayırık işlenebilmektedir.

### 2.1.3.b. Ayrık dalgacık dönüşümü

Dalgacık dönüşümü tüm ölçek aralığında yapılırsa çok büyük veri yığınları oluşmakta ve çok fazla işlem yapmak gerekmektedir. Bu istenmeyen bir durum olup, belirli ölçek grupları tespit edilir ve bu aralıklarda analizler yapılırsa, bu analize ayrık Dalgacık dönüşümü (ADD - Discrete Wavelet transform - DWT) denir (Misiti *et al.* 1996,1997). Matematiksel olarak SDD'deki gibi ifade edilmekle birlikte, farkı ölçeğin ve kaymanın ayrık olmasıdır.

ADD, 1988 yılında Mallat tarafından filtreler kullanılarak geliştirilmiştir. Mallat algoritması olarak da bilinen bu yöntem, iki kanal alt bant kodlayıcısı ile yapılan hızlı bir Dalgacık dönüşümü filtreleme algoritmasıdır (Demir 2008).

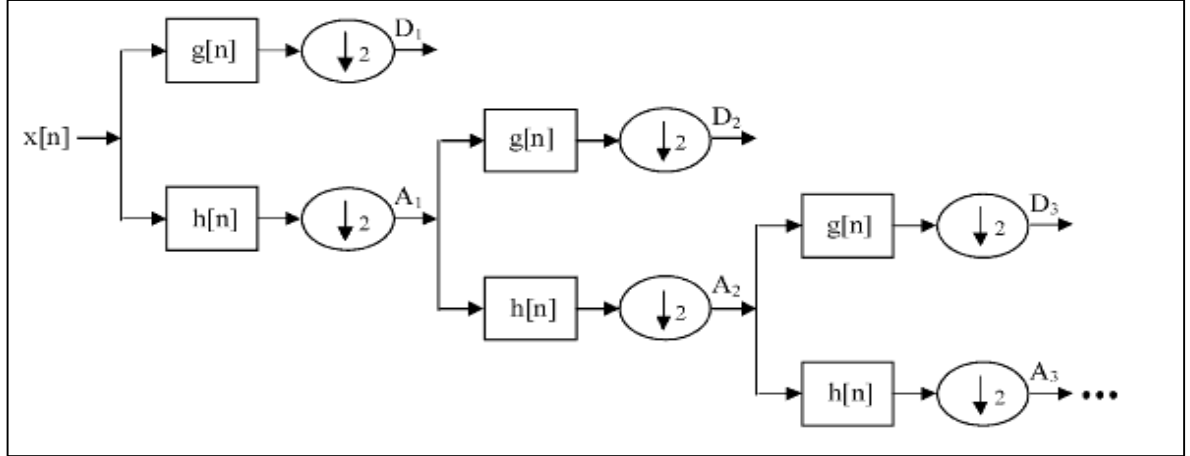
Ayrık Dalgacık dönüşümü, sinyalleri, eşit sayıda alt frekans bileşenine sahip iki sinyal olarak ayrıştırmaktır. Şekil 2.4'de görüldüğü üzere bir S sinyalinin alçak frekans bileşeni yaklaşım (Approximation=A) ve yüksek frekans bileşeni detay (Detail=D) olarak tanımlanır.



Şekil 2.4. Dalgacık dönüşümünün temel teorisi (Demir 2008)

Dalgacık dönüşümü ile sinyal belli sayıda ölçeklere ayrılmaktadır. Çoklu çözünürlük ayrışımı olarak isimlendirilen bu işlem  $x[n]$  sinyali için Şekil 2.5'de (Cvetkovic *et al.*

2008) gösterilmektedir. Bu işleme aynı zamanda alt bantlara ayırışım işlemi de denilmektedir.  $g[n]$  yüksek geçiren filtre,  $h[n]$  alçak geçiren filtre temsil etektedir.



**Şekil 2.5.** ADD uygulamasının alt bantlara ayırışımı

Dalgacık dönüşümü denklem 2.6. ile verilen şartı sağlayan alçak geçiren filtre  $h[n]$  ile belirlenebilmektedir.

$$H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 1 \quad (2.6.)$$

$H(z)$ ,  $h[n]$  filtresinin  $Z$  dönüşümü olarak gösterilmektedir. Bu filtrenin tamamlayıcı yüksek geçiren filtresi  $g[n]$  denklem 2.7. ile tanımlanabilmektedir.

$$G(z) = zH(-z^{-1}) \quad (2.7.)$$

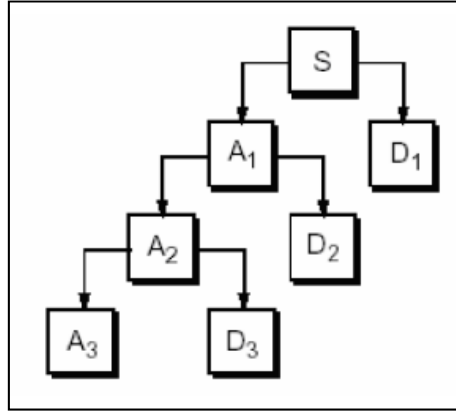
Artan uzunluklar ile filtre dizisi ( $i$  indeksi ile) denklem 2.8. gibi elde edilmektedir.

$$H_{i+1}(z) = H(z^{2^i})H_{i(z)} , \quad G_{i+1}(z) = G(z^{2^i})H_{i(z)} , \quad i = 0, \dots, I-1 \quad (2.8.)$$

Burada başlangıç şartı  $h_0(z) = 1$ 'dir. Bu, zaman bölgesinde denklem (2.9.) ile ifade edilmektedir.

$$h_{i+1}(k) = [h]_{\uparrow 2^i} * h_i(k), \quad g_{i+1}(k) = [g]_{\uparrow 2^i} * h_i(k) \quad (2.9.)$$

Burada,  $[.]_{\uparrow m}$  indeksi m faktörü ile yukarı örnekleme ve k eşit olarak örneklenmiş ayrık zamanı ifade etmektedir (Cvetkovic *et al.* 2008).



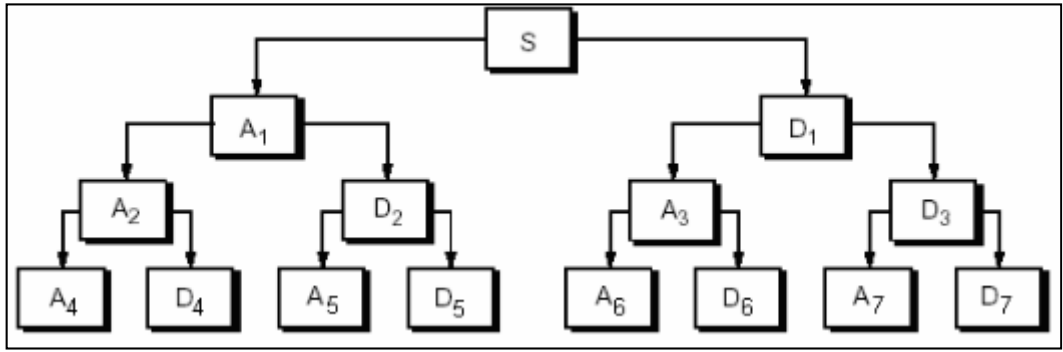
**Şekil 2.6.** Standart dalgacık ayrıştırma ağacı;  $S = A_3 + D_3 + D_2 + D_1$  (Günel 2008)

Ayrık Dalgacık dönüşümü uygulanan sinyal (S) kayba uğramaz, sadece ana dalgacıkla çarpılarak, bileşenlerine ayrıştırılır. İlk ayrışım sonucu elde edilen yaklaşım tekrar ikinci seviye yaklaşım ve detayına ayrıştırılır. Bu işlem istenilen sonuç elde edilene kadar tekrarlanabilir. Yaklaşımlar (A); yüksek ağırlıklı, düşük frekanslı bileşenler iken Detaylar (D); düşük ağırlıklı, yüksek frekanslı bileşenlerdir. Ayrık Dalgacık dönüşümü, yüksek frekanslarda yüksek zaman çözünürlüğü ve düşük frekans çözünürlüğü, alçak frekanslarda ise yüksek frekans çözünürlüğü ve düşük zaman çözünürlüğünü sağlamaktadır.



### 2.1.3.c. Dalgacık paket analizi

Her yaklaşım (A) ve detay (D) için aynı aritmetik düzenin kullanılmasıyla tam alt bant ayrışımı elde edilmektedir. Böylece bu işlemle, düşük frekans bileşenlerinin yanı sıra yüksek frekans bileşenlerinde de taşınan önemli bilgileri olan sinyallerin analizinde oldukça fayda sağlamaktadır.



Şekil 2.7. Dalgacık paket analizi (Günel 2008)

### 2.1.3.d. Dalgacık özellikleri ve seçimi

Bir fonksiyonun dalgacık olabilmesi için denklem 2.10. ve denklem 2.11. sağlaması gerekmektedir.

Teorik olarak sıfır ortalamalı;

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (2.10.)$$

ve sonlu enerjiye;

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \quad (2.11.)$$

sahip olmalıdır.

Dalgacık dönüşümünde öncelik, ana dalgacık fonksiyonunu seçmektir. Çünkü orijinal sinyal yayılmış bir dalgacığın katsayıları (dalgacık fonksiyonlarının lineer birleşimlerinin katsayıları) cinsinden ifade edilebilmektedir. Veriye iyi uyarlanmış bir dalgacık seçildiğinde veya belli bir eşik değerinin altındaki katsayıları atıldığında veri çok yaklaşık ancak kısmen temsil edilmektedir. Bu şekildeki kısmi kodlama dalgacıkları veri sıkıştırma alanında çok kullanışlı bir araç yapmaktadır.

Ancak, dalgacığı seçmek için birçok kriter vardır. Dalgacığın, zaman ve frekans ortamındaki sönümlenmesi önemlidir. Zaman ve frekansta iyi lokalize olabilmek için dalgacık zaman ve frekans ortamında hızlı sönümlenmelidir (Tüfekci and Gowdy 2000). Önemli kriterlerden diğerleri ise şunlardır(Aygün 2006).

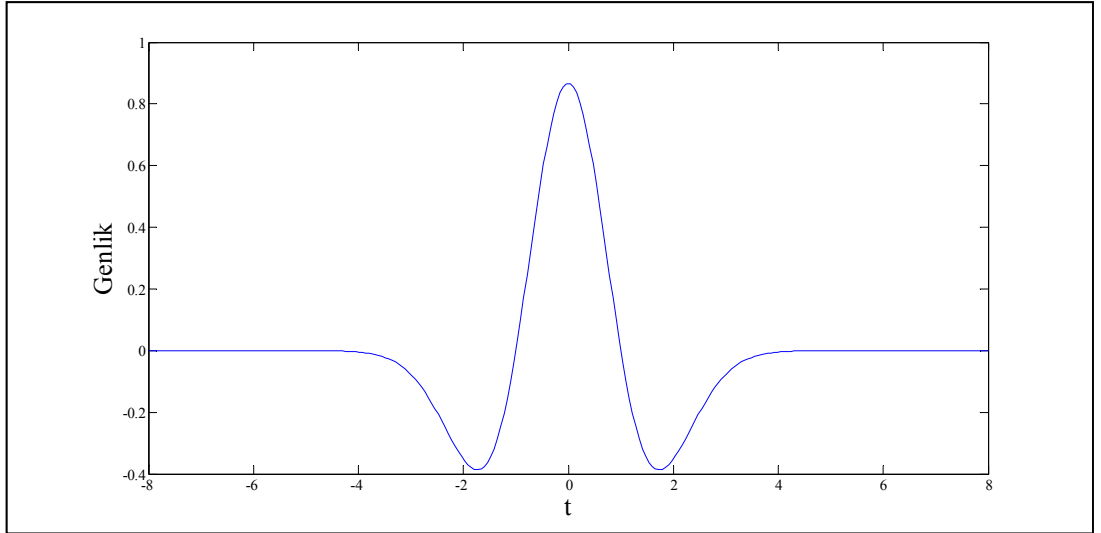
- Simetri: Özellikle görüntü işlemede defazdan kaçınmak için önemlidir.
- Kaybolma anlarının sayısı: Sıkıştırma amaçlı olarak kullanılacağı zaman önemlidir.
- Ortogonallik ve biortogonallik: Sonuç analizlerinde kolaylık sağlamaktadır.
- Ölçeklendirme fonksiyonu
- Düzenlilik (Regularity): Yeniden düzenlenen bir görüntünün daha ince özelliklerin çıkarılması için önemlidir.

### 2.1.3.e. Çok kullanılan ana dalgacık çeşitleri ve formülleri

Ayrık Dalgacık dönüşümünde birbirinden farklı dalgacık çeşitleri kullanılmaktadır. Bunlardan bazıları aşağıda verilmiştir.

Meksika şapkası (Mexican Hat) dalgacığı:

Meksika şapkası dalgacığı fonksiyonu (Şekil 2.8) denklem 2.12. ile tanımlanmaktadır.

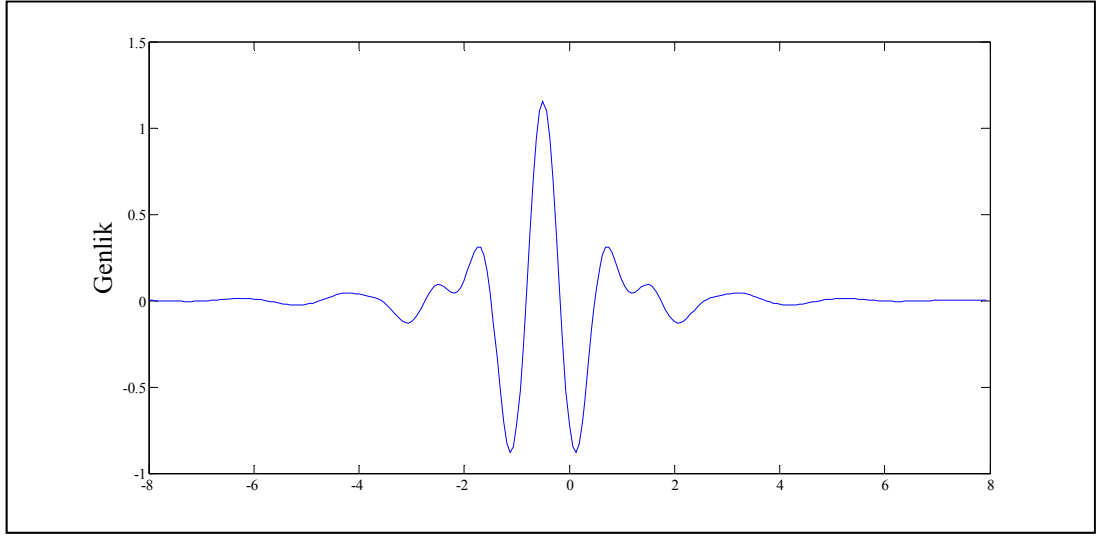


Şekil 2.8. Meksika şapkası dalgacığı fonksiyonu

$$\psi(t) = (1 - t^2)e^{-t^2/2} \quad (2.12.)$$

Meyer dalgacığı:

Meyer dalgacığı fonksiyonu (Şekil 2.9) denklem 2.13. ile tanımlanmaktadır.

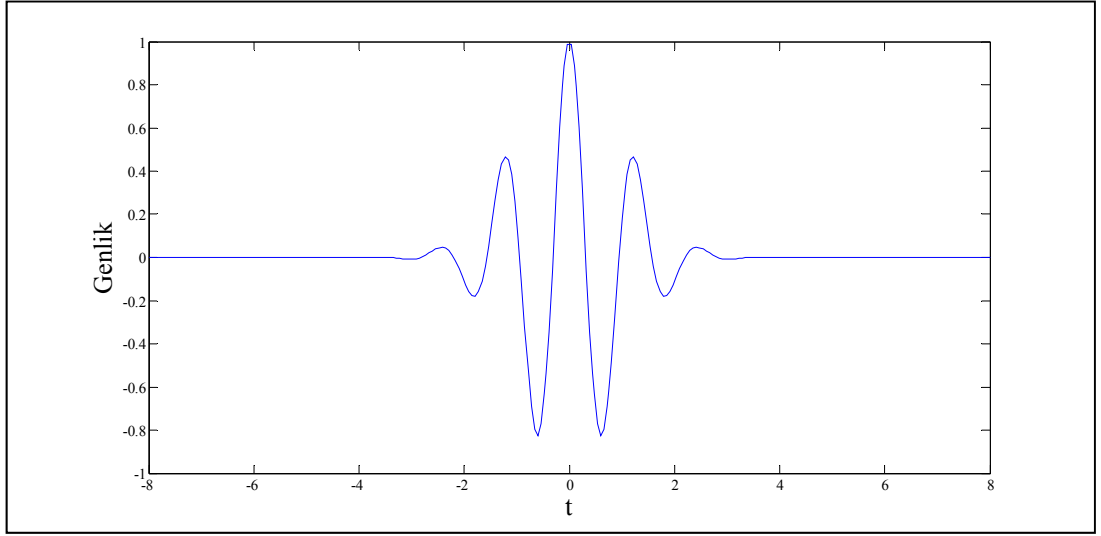


**Şekil 2.9.** Meyer dalgacıđı fonksiyonu

$$\bar{\psi}(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \sin\left(\frac{\pi}{2} v\left(\frac{3}{2\pi} |\omega| - 1\right)\right) \quad (2.13.)$$

Morlet dalgacıđı:

Morlet dalgacıđı fonksiyonu (Şekil 2.10) denklem 2.14. ile tanımlanmaktadır.

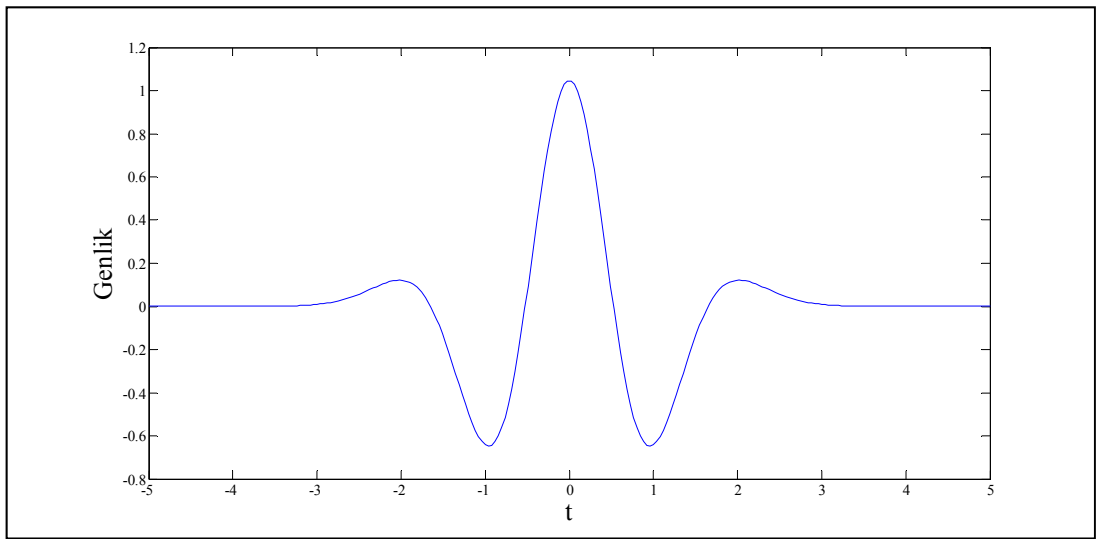


Şekil 2.10. Morlet dalgacıđı fonksiyonu

$$\psi(t) = (e^{-i\omega_0 t} e^{-\omega^2/2}) e^{-t^2/2} \quad (2.14.)$$

Gaus dalgacıđı:

Gauss dalgacıđı fonksiyonu (Şekil 2.11) denklem 2.15. ile tanımlanmaktadır.

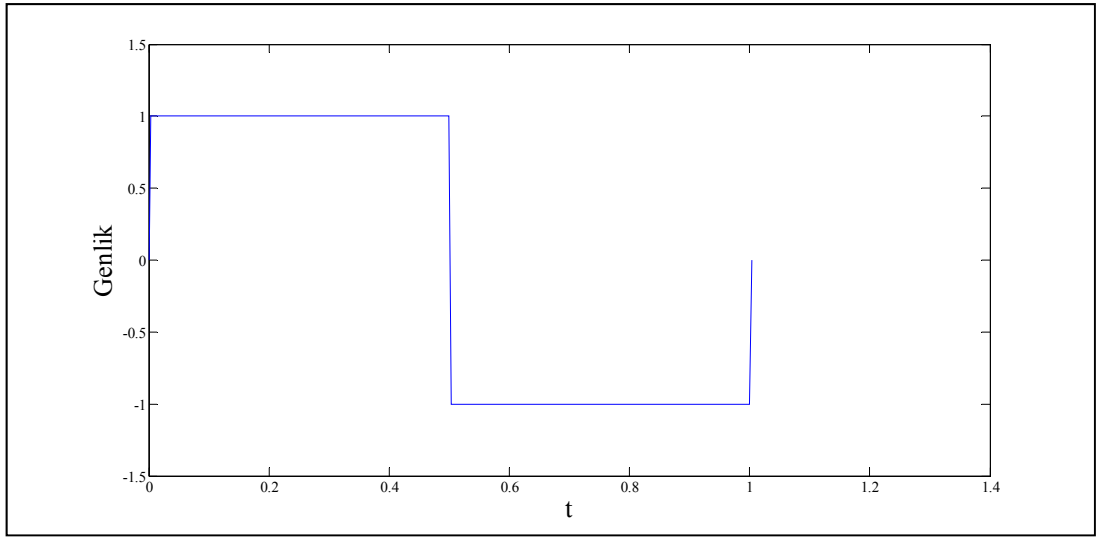


Şekil 2.11. Gauss dalgacıđı fonksiyonu

$$\psi(t) = C_p e^{t^2} \quad (2.15.)$$

Haar dalgacıđı:

Haar dalgacıđı ilk bilinen dalgacıktır. 1909 yılında Alfred Haar tarafından bulunmuştur. Mevcut dalgacık türleri içerisinde en basit olanıdır. En büyük eksikliđi süreksiz olmasıdır. Haar dalgacıđı fonksiyonu (Şekil 2.12) denklem 2.16. ile tanımlanmaktadır.



Şekil 2.12. Haar dalgacıđı fonksiyonu

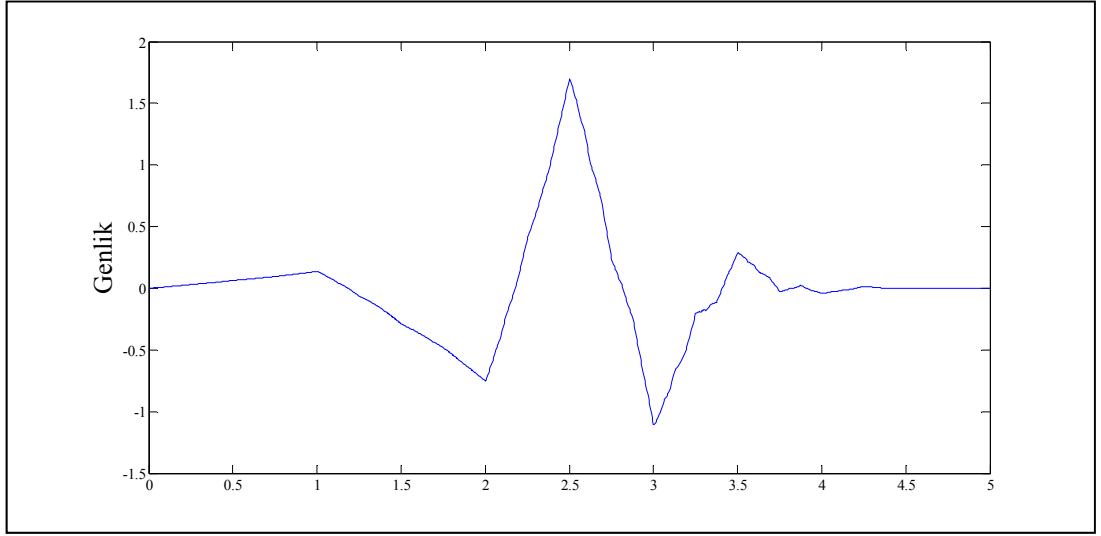
$$\psi(t) = 1 \quad , 0 \leq t < 1/2$$

$$\psi(t) = -1 \quad , 1/2 \leq t < 1$$

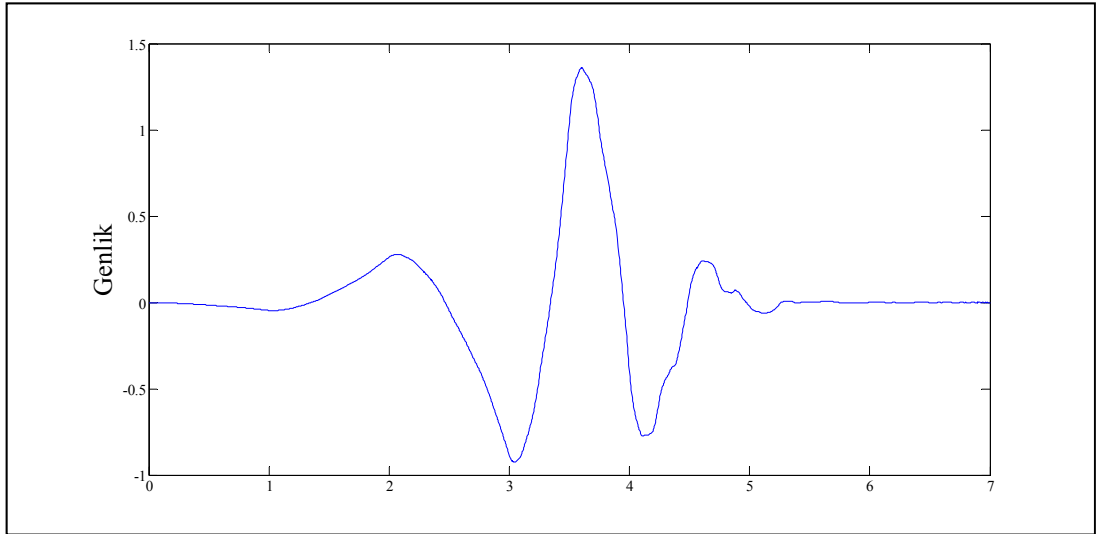
$$\psi(t) = 0 \quad , \text{aksi halde} \quad (2.16.)$$

Daubechies dalgacıđı:

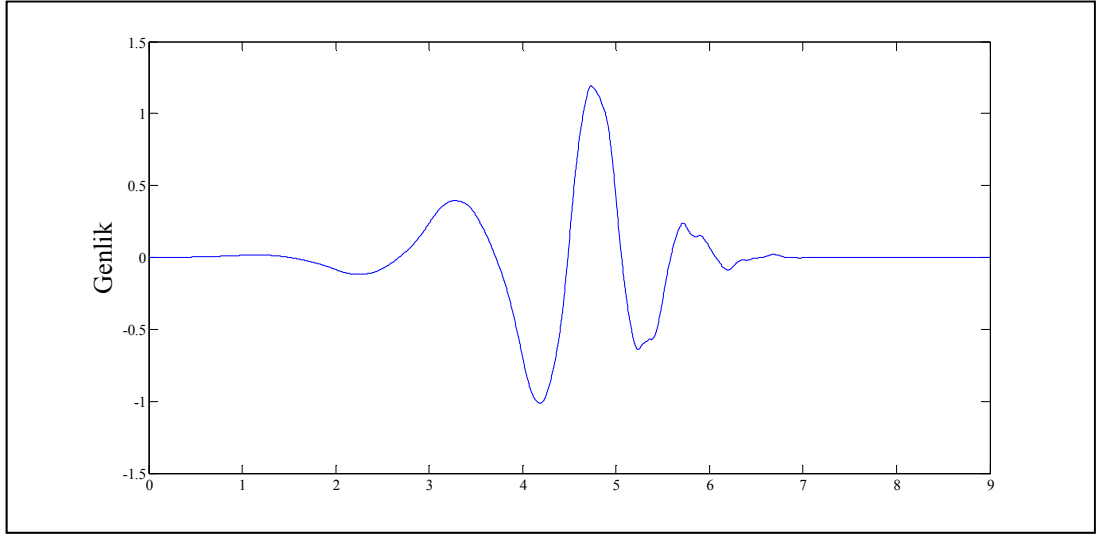
Daubechies dalgacıkları fonksiyonları (Şekil 2.13, Şekil 2.14, Şekil 2.15) denklem 2.17. ile tanımlanmaktadır.



Şekil 2.13. Daubechies 3 dalgacığı fonksiyonu



Şekil 2.14. Daubechies 4 dalgacığı fonksiyonu



**Şekil 2.15.** Daubechies 5 dalgacığı fonksiyonu

$$c(n) = h_0x(2n) + h_1x(2n + 1) + h_2x(2n + 2) + h_3x(2n + 3) \quad (2.17.)$$

### 2.1.3.f. İki boyutlu dalgacık dönüşümü

Çoklu çözünürlük analizi kullanılarak geliştirilen bir boyutlu Dalgacık dönüşümüne benzer şekilde iki boyutlu Dalgacık dönüşümü elde edilebilmektedir. Burada Dalgacık dönüşümü görüntüye uygulanırken, görüntü iki boyutlu bir diziymiş gibi düşünülmektedir. Önce bir boyutta (mesela yatayda) tüm satırlara alçak geçiren filtre ve yüksek geçiren filtre uygulanmaktadır. Daha sonra aynı işlem diğer boyutta da tekrarlanmaktadır.

Fakat bu şekilde dalgacıkların elde edilmesi çok zordur. Bunun için daha basit bir yaklaşımla iki tane bir boyutlu ortonormal dalgacığın tensör çarpımıyla iki boyutlu ortonormal dalgacık elde edilmektedir. (Daubechies 1990; Yakar 2006). Mallat tarafından geliştirilen bu işlem sonrasında  $x$  ve  $y$  değişkenlerine bağlı olarak bir ölçekleme fonksiyonu ve üç farklı dalgacık elde edilmektedir.



$$\Phi(x, y) = \phi(x)\phi(y) \quad (2.18.)$$

$$\Psi^1(x, y) = \phi(x)\psi(y) \quad (2.19.)$$

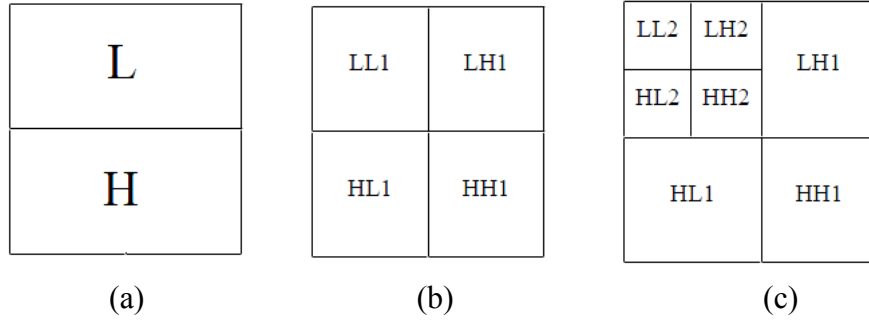
$$\Psi^2(x, y) = \psi(x)\phi(y) \quad (2.20.)$$

$$\Psi^3(x, y) = \psi(x)\psi(y) \quad (2.21.)$$

Burada  $\Phi(x, y)$ , ölçekleme fonksiyonu olup, ayrıştırılmış görüntünün yaklaşıklığını (A) temsil etmektedir.  $\Psi^1(x, y)$  yatay dalgacık,  $\Psi^2(x, y)$  dikey dalgacık ve  $\Psi^3(x, y)$  diagonal dalgacık olarak tanımlanmaktadır. Bu dalgacıklar ayrıştırılmış görüntünün detaylarını (D) temsil etmektedir. Böylece detay sinyal 3 tane alt görüntü içermektedir (Yakar 2006).

İki boyutlu Dalgacık dönüşümünde sinyal her ayrıştırma seviyesinde dört alt banda ayrıştırmaktadır. Bu bantların bir tanesi ölçekleme fonksiyonu ile ilgili sinyalin yaklaşıklık bilgisini içerir. Diğerleri yatay, dikey ve diagonal dalgacıklar ile ilgili sinyalin yatay, dikey ve diagonal detay bilgilerini içermektedir.

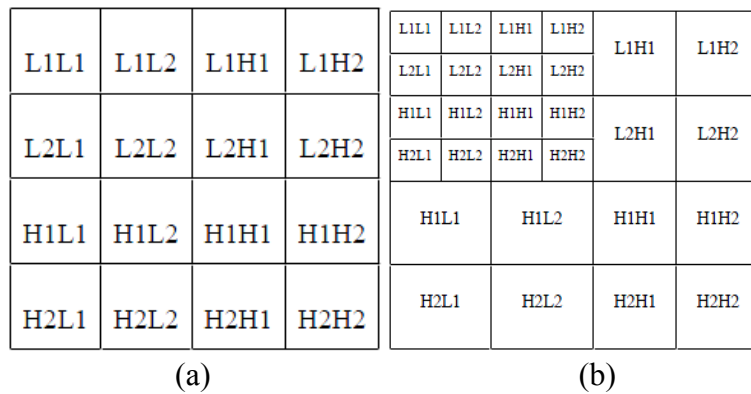
Şekil 2.16'da (a) bir boyutta ayrıştırmayı (b) 1.seviye iki boyutta ayrıştırmayı (c) 2.seviye iki boyutta ayrıştırmayı göstermektedir.



**Şekil 2.16.** İki boyutlu Dalgacık dönüşümü (Yakar 2006).

### 2.1.3.g. Çoklu-dalgacık dönüşümü

Çoklu-Dalgacık dönüşümü birden fazla ölçekleme fonksiyonu ve birden fazla dalgacık fonksiyonu kullanan Dalgacık dönüşümü olarak tanımlanmaktadır. Dalgacık dönüşümünde olduğu gibi çoklu-Dalgacık dönüşümünde çoklu çözünürlük analizi ile açıklanabilmektedir. Tek fark Dalgacık dönüşümü için tek bir ölçekleme fonksiyonu varken, çoklu-dalgacık için birden fazla ölçekleme fonksiyonu olmasıdır. Şekil 2.17’de (a) 1.seviye (b) 2.seviye çoklu-Dalgacık dönüşümünü göstermektedir.



**Şekil 2.17.** İki boyutlu çoklu-Dalgacık dönüşümü ayrışımı (Yakar 2006).

Literatürde yapılan çalışmalarda çoklu-Dalgacık dönüşümü ile yapılan her ayrıştırmanın Dalgacık dönüşümü ile yapılan 2 seviye ayrıştırmaya denk olduğu belirtilmiştir.

Örneğin, çoklu-Dalgacık dönüşümü ile yapılan 2 seviye ayrıştırma Dalgacık dönüşümü ile yapılan 4 seviye ayrıştırmaya denk gelmektedir (Yakar 2006).

## 2.2. Literatürde Seyrek Geometrik Görüntü Gösterimi

Çoklu ölçek yöntemleri, özellikle son on yıl içerisinde dalgacıkların gelişimiyle birlikte, çok popüler hale gelmiştir. Klasik dalgacığın başarısına rağmen, geleneksel dalgacıkların birden daha yüksek boyutta etkinliklerini sorgulatan bazı güçlü sınırlamalar olduğu ileri sürülmektedir. Dalgacıklar, tüm ölçeklerde ve konumlarda yön bağımsız (izotropik) elemanlardan meydana gelen bir bütüne dayanmaktadır. Dalgacıklar yüksek dereceli yön bağımlı (anizotropik) elemanları tanımlayamazlar ve ölçekten bağımsız yönlü elemanları sadece sabit bir sayıda içermektedir. Bu sebeple Ridgelet ve Curvelet gibi yeni yapılar önerilmiştir. Ridgeletler ve Curveletler, son zamanlarda hesaplamalı ve uygulamalı harmonik analiz alanında heyecan verici araştırmalara yol açan çok ölçekli yönelim seçici dönüşümler ailesinin özel üyeleridir. Bu aileye ait birçok başka yapılar da contourlets, directionlets, bandlets, grouplets, shearlets vb. literatürde incelenmiştir. Bu geometrik veya yönlü dalgacıklar genel olarak X-lets olarak adlandırılmaktadırlar. Bu konularda yapılan bazı temel çalışmalara bu bölümde yer verilmektedir (Fadili and Starck 2007).

### 2.2.1. Contourlet

Burada, görüntüler için resimsel bilginin içsel geometrik yapısını yakalayabilen yeni bir "gerçek" iki boyutlu temsilin geliştirilmesi üzerinde durulmaktadır. Do ve Vetterli tarafından önerilen Contourlets, yönlü bir filtre bankasıyla Laplace piramidini birleştiren, piramidal yönlü filtre bankası denilen, bir çift filtre bankası yapısından bahsedilmektedir. Sonuç Contourlet dönüşümüdür. Contourlet dönüşümü görüntüler için esnek çoklu çözünürlük ve yönlü ayrışma sağlamaktadır. Contourlet dönüşümü, eğriler de anizotropik ölçekleme ilişkisini karşılamak için tasarlanmıştır. Böylece, curvelet gibi örneklenen sinyalleri ayrıştıran hızlı bir yapı sunmaktadır.

Bu yeni piramidal yönlü filtre bankası yapısında, ilk olarak laplacian piramit noktasal süreksizlikler (kenarları) yakalanmakta, sonra yönlü bir filtre bankasıyla lineer yapılar içinde bu noktasal süreksizlikler bağlanmaktadır. Sonuçta, bu dönüşümü görüntülere benzer iki boyutlu parçalı yumuşak sinyaller için seyrek bir temsili sağlamaktadır (Do and Vetterli 2001).

Contourlets'in önemli bir bileşeni olan yönlü filtre bankası, dikkatlice tasarlanan filtreler tarafından örtüşmenin olmasına izin verilen ve iptal edilebilen uygun bir ağaç yapısındadır. Contourlets ve curvelets dönüşümleri arasındaki ana farklılık contourlet dönüşümünün doğrudan ayrık Kartezyen koordinatlar üzerinde tanımlı olmasıdır. Maalesef, contourlet fonksiyonları curvelet fonksiyonlarından görüntüden gürültü giderme ve görüntü sıkıştırma başta olmak üzere uygulamalarda daha az net geometrik özelliklere sahiptir (Ma and Plonka 2010).

### **2.2.2. Surfacelets**

Surfacelets, bir yüksek boyutlu yönlü filtre bankası ve bir çok ölçekli piramit ile elde edilen 2-D contourlets'in 3-D uzantılarıdır. Çok boyutlu hacimsel veri içeren biyomedikal görüntüleme, sismik görüntüleme, video işlemede verimli bir şekilde yüzey benzeri tekilliklerin gösteriminde ve yakalanmasında kullanılmaktadır. Surfacelets ve 3-D curvelets, aynı frekans bölümlenmesini amaçlamakta, fakat iki dönüşüm bu amaca farklı yaklaşımlar ile ulaşmaktadır. Surfacelets dönüşümü 3-D Curvelet dönüşümünden daha az fazlalığa sahiptir. Bu avantajı yön özelliklerinin belirli kaybıyla sağlamaktadır (Ma and Plonka 2010).

### **2.2.3. Directionlets**

Görüntü işlemede standart Dalgacık dönüşümünün (WT) başarısına rağmen, temsil verimliliği yalnızca yatay ve dikey yönlerde yerleşik temel fonksiyonların uzamsal

izotropisiyle sınırlıdır. Görüntülerde tek boyutlu (1-D) süreksizlikler (kenarlar ve konturlar), çok sayıda dalgacık temelli fonksiyonun kesiştiği ve temsilin seyrekliğinin azaldığı, görsel algılamada çok önemli unsurlardır. Verimli bir şekilde bu anizotropik geometrik yapıları yakalamak için, daha karmaşık bir çok yönlü (M-DIR) ve anizotropik dönüşümü gerekmektedir. Burada, mükemmel yeniden yapılanma sağlayan yeni bir kafes tabanı ve örneklenmiş anizotropik M-DIR Dalgacık dönüşümü sunulmaktadır.

Burada; sadece yatay ve dikeyde değil, pek çok yönde ayrılabilir filtreleme kullanarak görüntüler için yeni anizotropik bir dönüşüm olmaktadır. Bu dönüşüm, hesaplama verimliliği sağlamak ve standart Dalgacık dönüşümünden gelen filtre tasarım basitliği korumaktadır. Ayrıca bu yeni dönüşüm daha önce önerilen dönüşüm yapılarına göre görüntülerin doğrusal olmayan yaklaşımı için etkin bir araç sağladığını göstermektedir (Velisavljevic *et al.* 2006).

#### **2.2.4. Bandlets**

Görüntülerdeki kenarların geometrik düzenliliğinin avantajlarından faydalanılarak bir seyrek görüntü gösterimi tanıtılmaktadır. Foveal Dalgacık denilen, tek boyutlu dalgacık ortonormal temellerin yeni bir sınıfı olarak, teklikleri algılamak ve yeniden yapılandırmak için tanıtılmaktadır. Foveal dalgacıklar gelişigüzel eğrilerin geometrisini izleyerek iki boyutta genişletilmektedir. Bunun sonucunda iki boyutlu Bandeletler, az sayıda sıfır olmayan katsayılarla (sparse) düzenli kenarların yakın yaklaşımlarını yenileyebilen ortonormal aileleri tanımlamaktadır. Bir çift katmanlı görüntü kodlama algoritması tanımlanmaktadır. Kenarlar, nicemlenmiş bandelet katsayıları ile ve kalan düzgün şekil, standart iki boyutlu bir dalgacıkla kodlanmaktadır (Le Pennec 2000).

### 2.2.5. Grouplets

Grouplet ortogonal temeller ve dar çerçeveler, uzayda veya zamanda geometrik görüntü düzenliklerinden yararlanan grup noktaların yani alanların birlikteliğiyle oluşturulmaktadır. Bu alanlar birçok bağlantı noktasının birleştirilebildiği çok ölçekli bir geometriye sahiptir. Hızlı Grouplet dönüşümü ortogonal çok ölçekli hiyerarşik gruplamalar ile hesaplanmaktadır. Dalgacık görüntü katsayılarına uygulanan Grouplet dönüşümü bir ortogonal temel veya gruplanan Bandletlerin sıkı bir çerçevesi olarak tanımlanmaktadır. Görüntü yakınlaştırma ve gürültü giderme uygulamaları tanımlanmıştır (Mallat 2007).

### 2.2.6. Shearlets

Shearlets denilen çok boyutlu gösterim sistemlerinin yeni bir sınıfı tanımlanmıştır. Bunlar, genleşme eylemleri, kesme dönüşümü ve sabit bir fonksiyon ötelemesinin uygulanmasıyla elde edilmektedir. Geometrik ve matematiksel özellikler, örneğin; yönlülük, uzatılmış şekiller, ölçekler, salınımlar göstermektedir. Son zamanlarda seyrek görüntü işleme uygulamaları için birçok yazar tarafından savunulmaktadır. Bu sistemler, genellenen çoklu çözünürlük analizinin yapısı içinde çalışabilmektedir. Bu yaklaşım, bu sistemlerin uygulanması için, klasik kaskad algoritmasını genelleştiren özyinelemeli bir algoritma yol açmaktadır (Labate 2005).

### 2.3. Ridgelet Dönüşümü

Dalgacık dönüşümünün avantajı, mevcut dönüşümlere göre hem zaman hem frekans alanında yerleştirme sayesinde bir sinyalin çoklu çözünürlük analizi gerçekleştirmesidir. Ancak medikal görüntü bölütleme gibi iki veya daha üzeri boyutlardaki uygulamalarda Dalgacık dönüşümü kullanılırken en önemli problem dönüşümün noktasal tekillilik özelliği göstermesidir. Görüntülere Dalgacık

dönüşümünü uygulandığında süreksizlik noktası doğru algılanmasına rağmen özellikle sırtlar (düz çizgi) veya eğriler gibi yüksek boyutlu tekillikler içeren tıbbi görüntülerde dalgacıklar tekillikleri algılamakta yetersiz kalmaktadır. Bu nedenle, Dalgacık dönüşümünün başarısı bir medikal görüntü bölütleme tekniği olarak sınırlıdır. Dalgacık dönüşümünün yüksek boyutlardaki bu zayıflığının üstesinden gelmek için iki boyutta çizgisel tekilliklerde daha etkili olan Ridgelet dönüşümünü önerilmiştir.

### 2.3.1. Sürekli ridgelet dönüşümü

Tümlevleşebilir iki değişkenli fonksiyon  $\mathbf{f}(\mathbf{x})$  göz önüne alındığında,  $R^2$ 'de sürekli Ridgelet dönüşümü (SRD - Continuous Ridgelet transform - CRT) denklem 2.22. ile tanımlanmaktadır (Do and Vetterli 2003).

$$CRT_f(a, b, \theta) = \int_{R^2} \psi_{a,b,\theta}(x) f(x) dx \quad (2.22.)$$

Burada iki boyutlu (2-D)  $\psi_{a,b,\theta}(x)$  ridgelet, herhangi bir ölçekte  $a>0$ , konumda  $b \in R$  ve yönelim  $\theta \in [0, 2\pi]$  için bir boyutlu (1-D)  $\psi(x)$  ana dalgacık fonksiyonundan denklem 2.23.'deki gibi elde edilmektedir.

$$\psi_{a,b,\theta}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x_1 \cos \theta + x_2 \sin \theta - b}{a}\right) \quad (2.23.)$$

Bir ridgelet  $\theta$  açısında yöneldiği çizgiler boyunca sabittir. Yani iki boyutlu uzayda  $x_1 \cos \theta + x_2 \sin \theta = \text{sabit}$  şartını sağlayan noktalarda ridgelet de aynı değerinde sabit olmaktadır.

Karşılaştırmak için  $\mathbf{f}(\mathbf{x})$ 'in  $R^2$ 'de iki boyutlu sürekli Dalgacık dönüşümü denklem 2.24. şeklinde yazılabilmektedir.

$$CWT_f(a_1, b_1, a_2, b_2) = \int_{\mathbb{R}^2} \psi_{a_1, b_1, a_2, b_2}(x) f(x) dx \quad (2.24.)$$

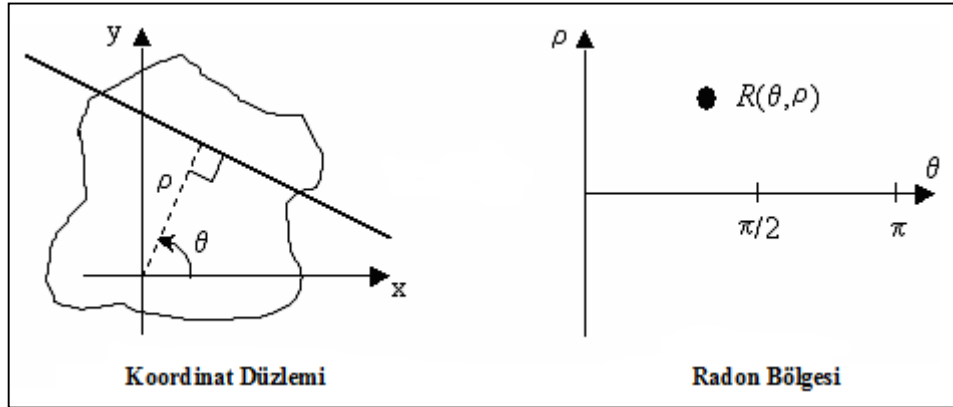
Burada  $\psi_{a_1, b_1, a_2, b_2}(x)$ , bir boyutlu  $\psi_{a, b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$  dalgacıkların tensör çarpımıyla  $\psi_{a_1, b_1, a_2, b_2}(x) = \psi_{a_1, b_1}(x_1) \psi_{a_2, b_2}(x_2)$  şeklinde iki boyutta dalgacığı ifade etmektedir. Görüldüğü gibi, iki boyutlu SDD'deki nokta parametreleri  $(b_1, b_2)$  yerine, SRD'de çizgi parametreleri olan  $(b, \theta)$  kullanılmaktadır.

Sonuç olarak, dalgacıklar izole noktasal tekillikler ile nesnelerin temsilinde çok etkiliyken, ridgeletler çizgiler boyunca olan tekillikler ile nesnelerin temsilinde etkili olmaktadır. Aslında Ridgelet dönüşümü iki boyutlu uzayda belli bir açıda sıralanmış doğrular boyunca hesaplanmış çizgi integral fonksiyonuna bir boyutlu Dalgacık dönüşümünün uygulanması olarak da düşünülebilir. Dolayısıyla, görüntü işlemede ridgeletleri kullanmak; tekillikler kenarlar veya konturlar boyunca birleştiği için cazip olmaktadır (Do and Vetterli 2003).

Radon dönüşümü iki boyutlu görüntünün özelliklerinin belirlenmesini sağlayan bir dönüşümdür. Görüntüdeki çizgileri, radon bölgesindeki noktalara dönüştürmektedir. İki boyutta, noktalar ve çizgiler Radon dönüşümü ile ilişkilendirilmekte ve böylece Dalgacık ve Ridgelet dönüşümleri Radon dönüşümüyle bağlanmaktadır.

Radon dönüşümü  $(x, y)$  Kartezyen koordinat sistemindeki noktaların  $(\theta, p)$  Polar koordinat sistemindeki iz düşümlerini hesaplamaktadır.





**Şekil 2.18.** Koordinat düzleminde Radon bölgesine geçiş (Çamaşırcıoğlu 2007)

İki boyutlu bir  $f(x,y)$  fonksiyonunun Radon dönüşümü denklem 2.25. ile tanımlanmaktadır.

$$\mathcal{R}_f(\theta, p) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - p) dx dy \quad (2.25.)$$

Bu durumda, Ridgelet dönüşümü, sabit  $\theta$  için elde edilmiş Radon dönüşümü dilimlerine bir boyutta Dalgacık dönüşümü uygulanması olarak da tanımlanabilmektedir.

$f(x,y)$  görüntüsüne öncelikle  $(\theta, t) \in [0, 2\pi) \times \mathbb{R}$  parametreleriyle Radon dönüşümü uygulandığında  $\mathcal{R}_f(\theta, t) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - t) dx dy$  elde edilmektedir. Sonra bu ifadenin bir boyuttaki Dalgacık dönüşümü hesaplandığında denklem 2.26. elde edilmektedir.

$$CRT_f(a, b, \theta) = \int_{\mathbb{R}^2} \mathcal{R}_f(\theta, t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (2.26.)$$

Burada  $\mathcal{R}_f(\theta, t)$  yerine konulursa,

$$CRT_f(a, b, \theta) = \int \{ \iint f(x, y) \delta(x \cos \theta + y \sin \theta - t) dx dy \} \frac{1}{\sqrt{a}} \psi \left( \frac{t-b}{a} \right) dt, \quad (2.27.)$$

$$CRT_f(a, b, \theta) = \iint f(x, y) \frac{1}{\sqrt{a}} \left\{ \int \delta(x \cos \theta + y \sin \theta - t) \psi \left( \frac{t-b}{a} \right) dt \right\} dx dy, \quad (2.28.)$$

$$CRT_f(a, b, \theta) = \iint f(x, y) \frac{1}{\sqrt{a}} \psi \left( \frac{x \cos \theta + y \sin \theta - b}{a} \right) dx dy, \quad (2.29.)$$

$$CRT_f(a, b, \theta) = \iint f(x, y) \psi_{a,b,\theta}(x, y) dx dy \quad (2.30.)$$

Şeklinde Ridgelet katsayıları elde edilmektedir. Yani Ridgelet dönüşümü Radon dönüşümünün çoklu çözünürlük uygulaması olarak ta değerlendirilebilmektedir.

### 2.3.2. Sonlu ridgelet dönüşümü

Ayrık Ridgelet dönüşümü ayrık Radon dönüşümü kullanılarak yapılabilmektedir. Radon dönüşümünün çok sayıda bölümlenmeleri sürekli formlere yaklaşmak için bulunmuştur. Ancak, bunların çoğu dijital görüntüler için ters çevrilebilir dönüşümler olarak tasarlanmamıştır. Alternatif olarak, sonlu Radon dönüşümü (Finite Radon transform - FRAT) teorisi, ilginç bir çözüm sağlamaktadır. İlgili dönüşüm sürekli Radon dönüşümünün periyodizasyonundan türetilmiştir.

Sonlu Ridgelet dönüşümü (Finite Ridgelet transform - FRIT); sonlu Radon dönüşümünün hesaplanması ve sonra da Dalgacık dönüşümün uygulanmasıyla iki adım da hesaplanmaktadır. Sonlu Radon dönüşümü de; görüntünün iki boyutlu Hızlı Fourier dönüşümünün (FFT) hesaplanması ve sonra bir boyutlu ters hızlı Fourier dönüşümünün (IFFT) radon izdüşümünün tüm radyal yönler üzerine uygulanmasıyla iki adımda hesaplanmaktadır. Sonra bir boyutlu Dalgacık dönüşümü, üç seviyeli ayrışma için orijinden geçmekte olan radyal yönler kısıtlama uygulamaktadır (AlZu'bi *et al.* 2011).

FRAT çizgilerin belirli bir kümesi üzerinde görüntünün piksel toplamları olarak tanımlanmaktadır. Bu çizgiler, Öklid geometrisinde sınırlı bir geometrideki sürekli Radon dönüşümündeki çizgilere benzer bir yolla tanımlanmaktadır.

Bir  $f(i,j)$  fonksiyonunun sonlu Radon dönüşümü,  $Z_p^{2,2}$  de (finite grid) denklem 2.31. ile tanımlanmaktadır.

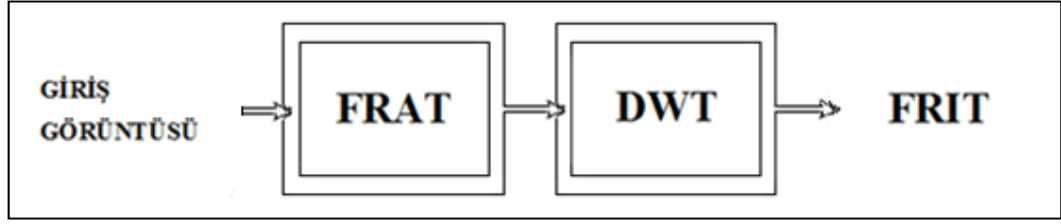
$$r_k[l] = FRAT_f(k, l) = \frac{1}{\sqrt{p}} \sum_{(i,j) \in L(k,l)} f(i, j) \quad (2.31.)$$

Burada,  $L(k,l)$   $Z_p^{2,2}$  de bir çizgiyi oluşturan noktalar kümesini göstermektedir.

$$\begin{aligned} L(k, l) &= \{(i, j) : j = k_i + l(\text{mod } p), \quad i \in Z_p\}, 0 \leq k < p, \\ L(p, l) &= \{(l, j) : j \in Z_p\} \end{aligned} \quad (2.32.)$$

K. radon izdüşümünün (yani dizideki K. satır) hesaplamak için orijinal görüntüdeki tüm piksellerin bir kez geçirilmesi gerekmekte ve satırdaki her piksel için bir P histogrammer kullanılmaktadır. Sonunda, tüm P histogram değerleri, ortalama değerler elde etmek için K ile bölünmektedir (AlZu'bi *et al.* 2011).

Dalgacık ve Radon dönüşümleri uygulandıktan sonra, Ridgelet dönüşümü basittir. Görüntüye uygulanan FRAT ile çizgisel tekillikler noktasal tekilliklerle eşleşmekte ve her FRAT izdüşümü dizisine Dalgacık dönüşümü uygulanarak radon bölgesinde noktasal tekilliklerin etkili bir şekilde algılanması ve bölütlenmesiyle bir sonlu Ridgelet dönüşümü (Finite Ridgelet transform - FRIT) elde edilmektedir (Şekil 2.19).



**Şekil 2.19.** FRIT diyagramı

Tıbbi görüntüler Radon dönüşümü uygulandıktan sonra da noktasal tekillikleri olmayan eğrilerden oluşmaktadır. Bu yüzden Dalgacık dönüşümü bu tekillikleri doğru algılayamamaktadır. Bu durum, Ridgelet dönüşümünün çoğu tıbbi görüntünün bölütlenmesi için uygun olmadığını açıklamaktadır.

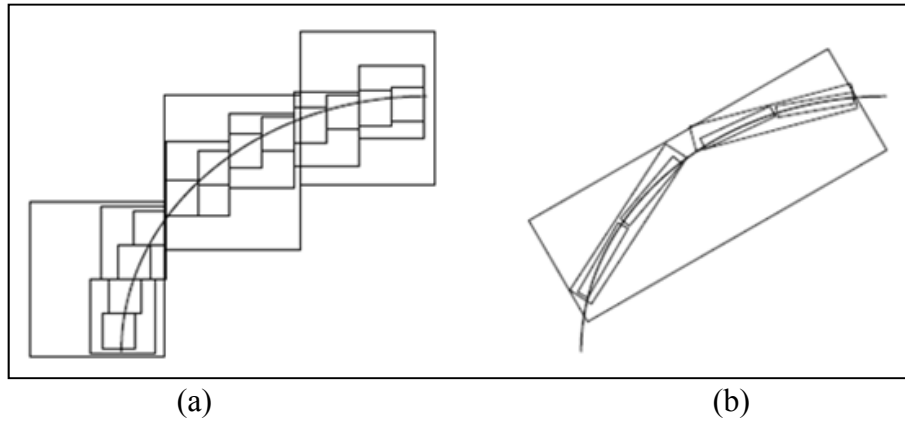
Ridgelet dönüşümü kenarlar ve düz çizgiler içeren görüntülerde kullanılabilir. Curvelet dönüşümü bu sorunu çözmek için geliştirilmiştir. Dalgacık ve Ridgelet dönüşümleriyle karşılaştırıldığında daha yüksek tekillikleri algılayabilmektedir (AlZu'bi *et al.* 2011) ve MRA kullanarak yapılan medikal görüntü bölütlemesi için daha güvenilir ve uygundur.

#### **2.4. Curvelet Dönüşümü**

Curvelet dönüşümü, görüntü gürültü giderme, görüntü ayrışma, doku sınıflandırma, astronomik görüntüleme ve kontrast geliştirme gibi alanlarda etkili bir araç olarak kullanılan yeni çoklu çözünürlük dönüşümüdür.

Dalgacık ile karşılaştırıldığında Curvelet, görüntüdeki her hangi bir süreksizliği çok az sayıda (“non-zero” sıfır olmayan) katsayı ile daha etkin bir şekilde temsil edebilmektedir. Dalgacık dönüşümü yatay, dikey ve çapraz olmak üzere üç yönelimdeki süreksizlikler için (sıfır olmayan) değer oluşturmaktadır. Yani Dalgacık sadece sınırlı olarak yön bilgisi yakalayabilmektedir. Her ne kadar Dalgacıklar noktasal süreksizliği

KSFD'den daha iyi işleyebilse de, eğriler için en uygun değildir. Çünkü Dalgacıklar yön bağımsızken eğriler yöne sahiptir, ve bu yüzden kenarları için çok fazla sayıda katsayı ile tanımlanmaktadır (Şekil 2.20.a). Curvelet dönüşümü yönelim ile yerelleşebildiği için eğrisel süreksizlikleri daha hassas bir şekilde işleyebilmektedir (Şekil 2.20.b).



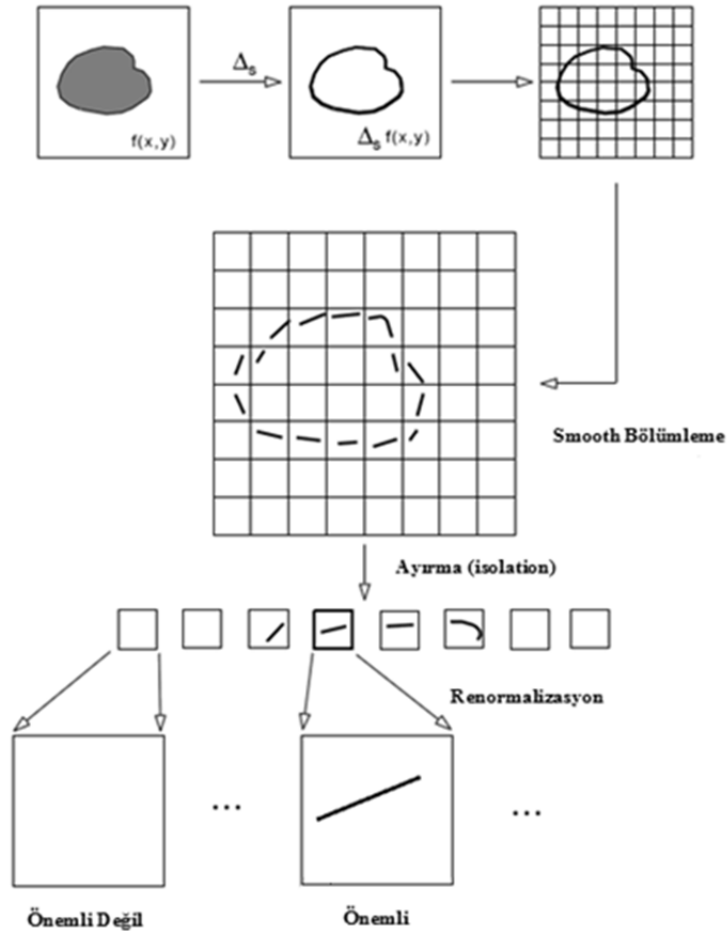
**Şekil 2.20.** Dalgacık ve Curvelet dönüşümleri arasındaki karşılaştırılması (AlZu'bi *et al.* 2011)

Curvelet dönüşümünün uygulanmasında iki temel yaklaşım söz konusudur. Bunlardan birincisi ayrı Ridgelet dönüşümü esas alınarak gerçekleştirilmiştir. Görüntü işlemede, kenar olarak tanımlanan yapılar düz çizgilerden çok eğrilerden oluşmakta ve Ridgelet dönüşümü bu tür görüntülerin temsilinde verimli olamamaktadır. Fakat eğriler içeren kenarlar, özellikle ince (fine) ölçeklerde neredeyse düz çizgiler olarak yerleştirilmiş şekilde indirgenerek ridgelet mekanizması uygulanabilmektedir. Bu birinci nesil Curvelet dönüşümünün (CurveletG1) temel fikridir (Fadili and Starck 2007).

Daha sonra, frekans bölme tekniğine dayalı ikinci nesil Curvelet dönüşümü (CurveletG2) önerilmiştir. CurveletG2, görüntü işleme, sismik veri incelenmesi, akışkanlar mekaniği gibi birçok farklı uygulamalar için çok etkili bir araç olarak gösterilmektedir (Ma and Plonka 2010).

### 2.4.1. Birinci nesil curvelet dönüşümü

Birinci nesil Curvelet dönüşümü, bir görüntüyü farklı blok boyutları kullanarak analiz etme olanağı sağlamaktadır. Dönüşümün temeli, bir dizi dalgacık bantları halinde görüntüyü ayrıştırmak ve her bandın Ridgelet dönüşümüyle analiz edilmesinden oluşmaktadır. Blok boyutları her ölçek seviyesinde değiştirilebilmektedir. Kabaca söylemek gerekirse, çok ölçekli ridgelet piramidinin farklı seviyeleri bir filtre bankası çıktısının farklı alt bantlarını temsil etmek için kullanılmaktadır. Bu alt bant ayrıştırması, aynı zamanda çerçeve elamanlarının uzunluğu ve genişliği arasında bir ilişki geliştirmektedir (Fadili and Starck 2007). Çerçeveler anizotropiktir (yönbağımlı) ve parabolik ölçekleme yasasına uymaktadırlar (genişlik  $\approx$  uzunluk<sup>2</sup>).



**Şekil 2.21.** Bir tek alt bandın uzamsal ayrışması (Donoho and Duncan 1999)



$$f \mapsto (P_0f, \Delta_1f, \Delta_2f, \dots) \quad (2.33.)$$

Bu adım görüntüyü birkaç çözünürlük seviyesine bölmektedir. Her seviye farklı frekans detayları içermektedir ( $P_0 \rightarrow$  Alçak geçiren filtre) ( $\Delta_1, \Delta_2, \dots \rightarrow$  Bant geçiren (yüksek geçiren) filtreler).

Denklem 6.2. ile orijinal görüntü alt bantlardan yeniden elde edilebilmektedir.

$$f = P_0(P_0f) + \sum_s \Delta_s(\Delta_s f) \quad (2.24.)$$

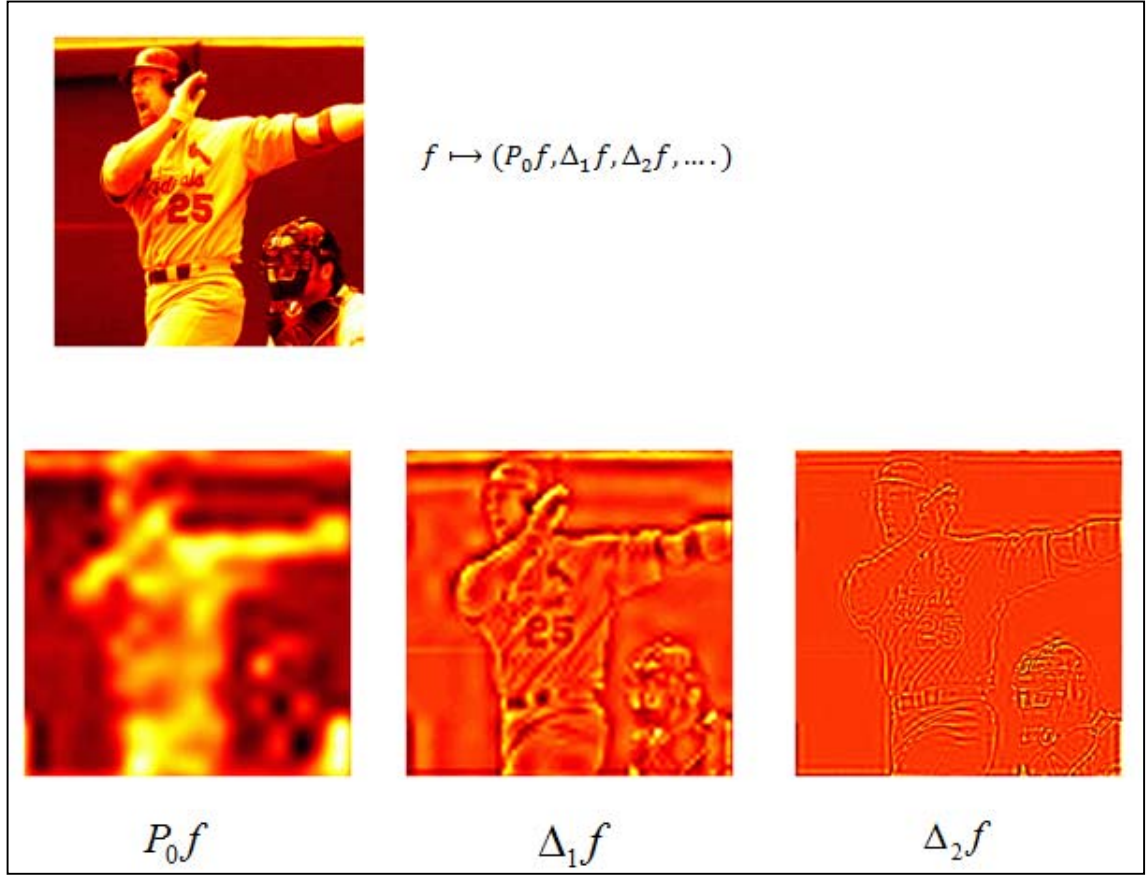
Zaman uzayında öteleme (translation) ve çarpma olarak tanımlanan bu işlem frekans uzayında basit bir konvolüsyon uygulamasıdır. Denklem 2.25 ile frekans uzayında alt bant ayrıştırması sağlanır. Burada,  $\Phi_0(\xi)$  ile  $|\xi| \leq 1$  bölgesinde etkin bir alçak geçiren filtre,  $\Phi_{2^s}(\xi)$  ile  $|\xi| \in [2^{2s}, 2^{2s+2}]$  bölgesinde tanımlı bir bant geçiren filtre ifade edilmektedir.

$$P_0f = \Phi_0 * f \quad , \quad \Delta_s f = \Phi_{2^s} * f \quad s = 0, 1, 2, \dots \quad (2.25.)$$

Curvelet ve Dalgacığın arasında bu bölümde bazı bağlantılar vardır. Alt bant ayrışımına bilinen Dalgacık dönüşümü kullanılarak yaklaşılabilmektedir. Dalgacık dönüşümü kullanılarak  $f$ ,  $S_0, D_1, D_2, D_3, \dots$ 'a ayrışmaktadır.  $P_0f$  kısmen  $S_0$  ve  $D_1$ 'den oluşmaktadır ve  $D_2$  ile  $D_3$ 'de içerebilmektedir.  $\Delta_s f$   $D_{2^s}$  ve  $D_{2^{s+1}}$ 'den oluşmaktadır.

$P_0f$  (low pass) yumuşak (smooth)'tır ve dalgacık tabanı kullanılarak verimli temsil edilebilmektedir. Fakat süreksizlik eğrileri,  $\Delta_s f$  yüksek geçiren seviyeleri etkilemektedir. Bununla eğrinin küçük bir parçasına bakıldığında, nispeten düz çizgiler görünmektedir. Ayrıca hatayı önlemek için küçük parçalar içinde seviyeler incelenmektedir.





Şekil 2.23. Alt bant ayrıştırması

#### 2.4.1.b. Yumuşak bölümlenme

Bir dizi yumuşak (smooth) pencere “ $w_Q(x_1, x_2)$ ” diyadik kareler etrafında yerleştirilmektedir.

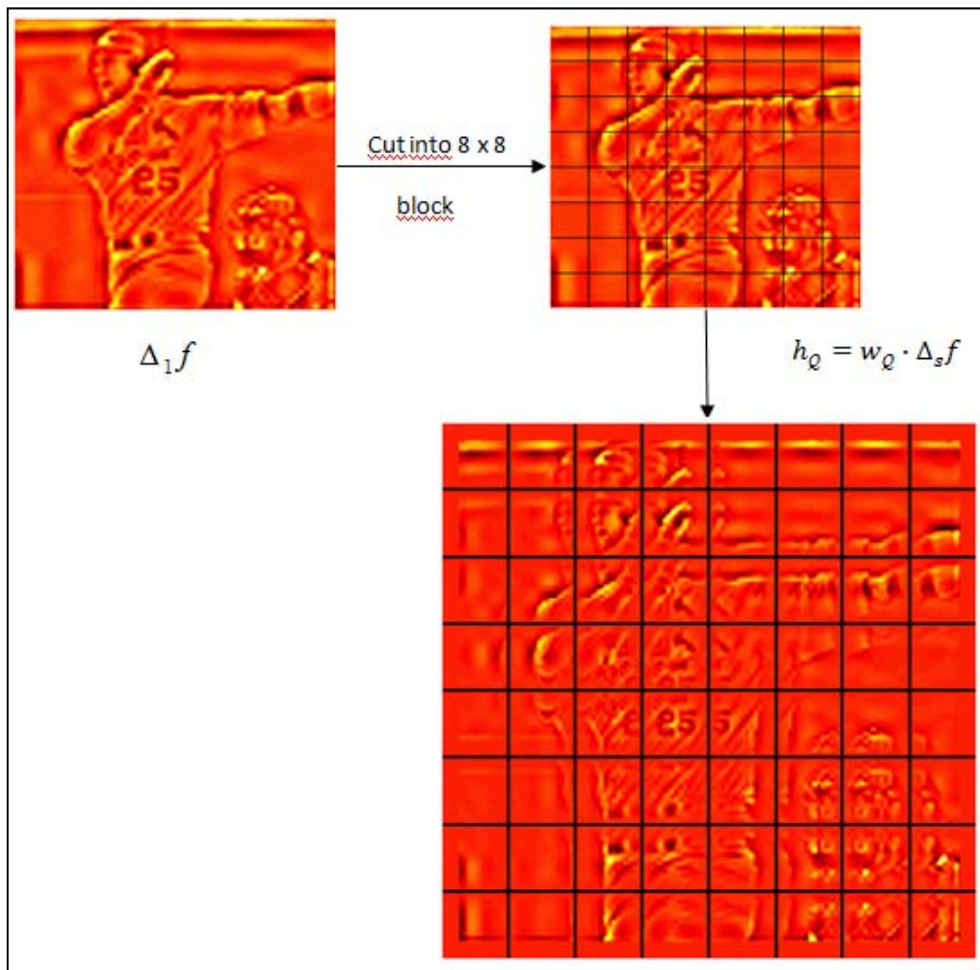
$$Q(s, k_1, k_2) = [k_1/2^s, (k_1 + 1)/2^s] \times [k_2/2^s, (k_2 + 1)/2^s] \in Q_s \quad (2.26.)$$

$W$ , boyutu  $2^s \times 2^s$  desteği ile smooth bir pencere işlevi olsun. İlgili pencere fonksiyonu  $w_Q$  ile bir fonksiyonun çarpımı  $Q$  ( $\forall Q \in Q_s$ ) yakın lokalize bir sonuç üretmektedir. Tüm  $Q$  için belirli bir ölçekte bu işlemi yapmak ( $k_1$  ve  $k_2$  değişken,  $s$  sabit) kareler içinde fonksiyonun smooth bir diseksiyonunu üretmektir. Bu aşamasında, algoritmanın

önceki işlemden izole edilmiş alt bantların her birine bu pencereleme diseksiyonu uygulanmaktadır. Ve bu adım kareler içine fonksiyonun smooth bir diseksiyon üretmektedir.

$$h_Q = w_Q \cdot \Delta_s f \quad (2.27.)$$

Görüntü pencere fonksiyonu ile çarpıldıktan sonra smooth bir hale gelmektedir. Bölümleme yapmak bizim yerel doğru ve eğri tekilliklerini analiz etmemizi kolaylaştırmaktadır.



Şekil 2.24. Smooth bölümleme

### 2.4.1.c. Normalizasyon

Bir diyadik kare  $Q$  için,  $f$  fonksiyonunu taşıyan ve normalize eden :

$$T_Q f(x_1, x_2) = 2^s f(2^s x_1 - k_1, 2^s x_2 - k_2) \quad (2.28.)$$

işlevi belirlenmektedir. Böylece  $Q$  yakınında desteklenen giriş parçası  $[0,1] \times [0,1]$  yakınında desteklenen çıkış parçası olmaktadır. Bu aşamada, önceki işlemde ortaya çıkan her kare birim ölçekle renormalize edilmektedir.

$$g_Q = T_Q^{-1} h_Q \quad (2.29.)$$

### 2.4.1.d. Ridgelet analizi

Her kare ortonormal Ridgelet sistemiyle analiz edilmektedir. Bu,  $L^2(R^2)$ 'de temel elemanı  $\rho_\lambda$  olan bir sistemdir. Ridgelet yapısı frekans bölgesini  $|\xi| \in [2^s, 2^{s+1}]$  diyadik koronalara (taç şeklinde yapılara) bölmektedir. Açısal yönde, en az  $2^s$  kez  $s$ 'inci korona örneklenmektedir. Radyal yönde, yerel dalgacıklar kullanılarak örneklenme yapılmaktadır. Ridgelet elemanı frekans bölgesinde denklem 6.8. ile ifade edilmektedir.

$$\rho_\lambda(\xi) = \frac{1}{2} |\xi|^{-\frac{1}{2}} (\psi_{j,k}(|\xi|) \cdot \omega_{i,l}(\theta) + \psi_{j,k}(-|\xi|) \cdot \omega_{i,l}(\theta + \pi)) \quad (2.30.)$$

Burada;

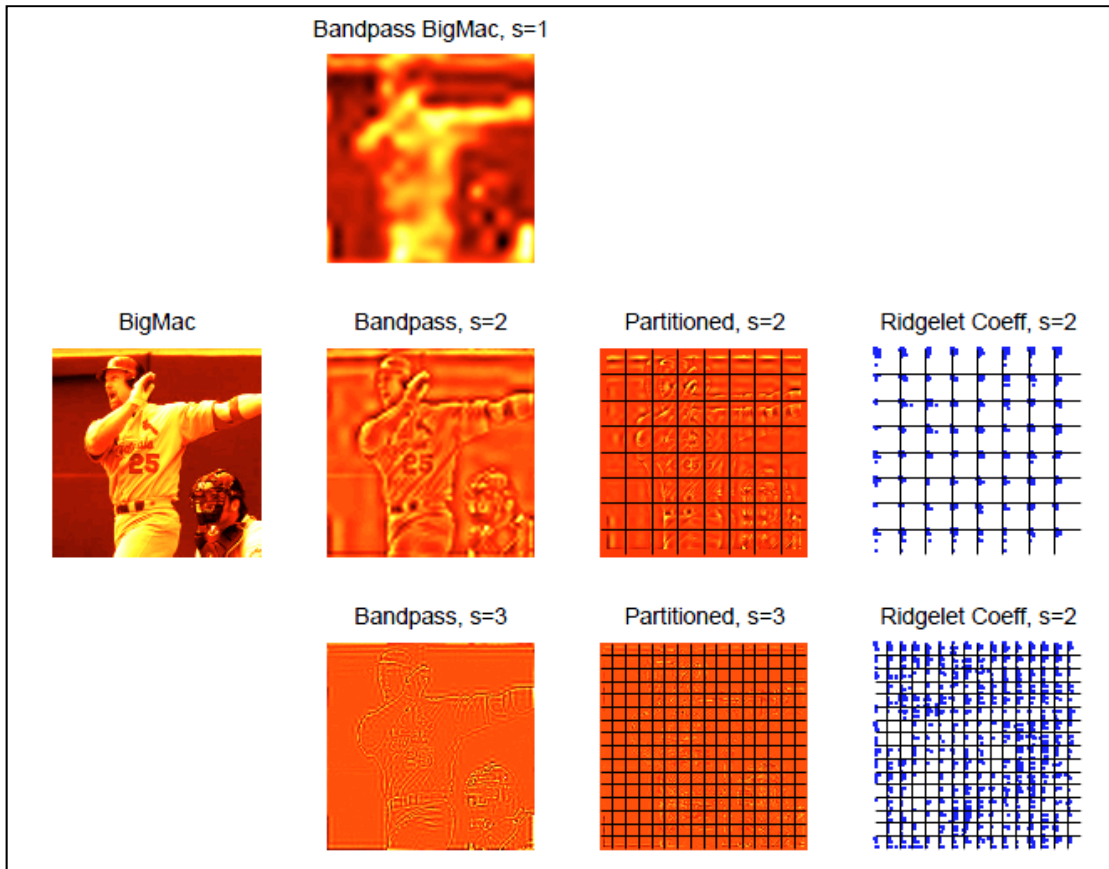
- $\omega_{i,l}$ : periodik dalgacık  $[-\pi, \pi]$ 'da ,

- $i$ : açısız ölçek ,  $l \in [0, 2^{i-1}-1]$ : açısız konum,
- $\psi_{j,k}$ : Meyer Dalgacıđı  $R$ 'de,
- $j$ : ridgelet ölçeđi,  $k$ : ridgelet konumu

olarak tanımlanırlar.

Her normalize kare Ridgelet sistemi içinde analiz edilmektedir.

$$\alpha_{(Q,\lambda)} = \langle g_Q, \rho_\lambda \rangle \quad (2.31.)$$



Şekil 2.25. Birinci nesil Curvelet dönüşümü adımları (Donoho and Duncan 1999)

### 2.4.2. Birinci nesil ters curvelet dönüşümü

Birinci nesil Curvelet dönüşümünün yeniden oluşturma (reconstruction) algoritması ayrıştırma algoritması adımlarının ters sırayla işletilmiş hali olarak tanımlanabilmektedir (Donoho and Duncan 1999).

#### 2.4.2.a. Ridgelet sentezi

Her kare ortonormal ridgelet sisteminden geri elde edilmektedir. Tüm ridgelet katsayıları bazla toplanmaktadır.

$$g_Q = \sum_{\lambda} \alpha_{(Q,\lambda)} \rho_{\lambda} \quad (2.32.)$$

#### 2.4.2.b. Normalizasyon

Önceki aşamada ortaya çıkan her kare kendi uygun karesine normalize edilmektedir.

$$h_Q = T_Q g_Q \quad , Q \in Q_s \quad (2.33.)$$

#### 2.4.2.c. Smooth tümleştirme

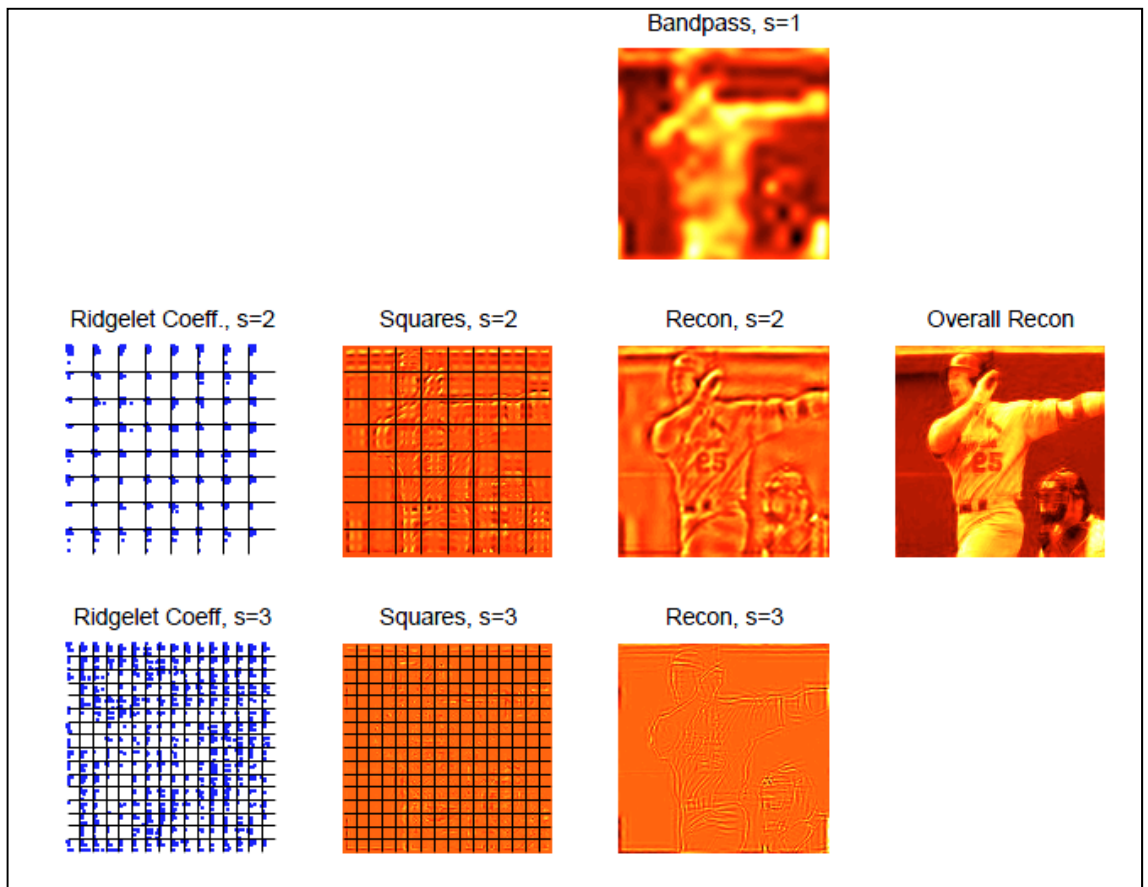
Pencereleme diseksiyonu algoritmanın önceki aşamasında yeniden oluşturulan her pencere için geri alınmaktadır.

$$\Delta_s f = \sum_{Q \in Q_s} w_Q h_Q \quad (2.34.)$$

### 2.4.2.d. Alt Bant birleştirme

Tüm alt bantların toplamını kullanarak  $f$  fonksiyonunu geri elde edilebilmektedir.

$$f = P_0(P_0f) + \sum_s \Delta_s(\Delta_s f) \quad (2.35.)$$



Şekil 2.26. Birinci nesil ters Curvelet dönüşümü adımları (Donoho and Duncan 1999)

### 2.4.3. İkinci nesil curvelet dönüşümü

Son on yıl içinde Curveletler, anlaşılabilirliğini ve kullanımını kolaylaştırmak için yeniden tasarlanmıştır. İkinci nesil Curveletler (CurveletG2) üç parametre ile (ölçek,

yön (açı) ve konum) doğal indeksleme yapısı sergilemekte, dolayısıyla matematiksel analiz basitleştirilmekte ve çok düşük bir fazlalığa (redundancy) sahip olmaktadır. DCTG1 aksine, ayrık CurveletG2 (DCTG2) uygulanmasında Ridgelet dönüşümü kullanılmamaktadır (Fadili and Starck 2007).

#### 2.4.3.a. 2-D curvelet dönüşümü

Bu bölümde, iki boyutta, uzamsal değişken  $x$ , frekans bölgesi değişkeni  $\xi$  ve  $r = \sqrt{\xi_1^2 + \xi_2^2}$  ile  $\omega = \tan^{-1} \xi_2/\xi_1$  frekans bölgesinde kutupsal koordinatları;  $j$  ölçeği,  $k$  traslation (ötelemeyi),  $l$  rotasyonu (döndürmeyi) ifade etmektedir.  $j$ 'inci seviyede temel Curvelet  $\phi_j(x)$  ile frekans bölgesinde  $\hat{\phi}_j(\omega)$  ise ifade edilmek üzere denklemi 2.36.'teki gibi pencere fonksiyonlarıyla tanımlanmaktadır.

$$\hat{\phi}_{j,0,0}(r, \omega) := 2^{-3j/4} W(2^{-j}r) \tilde{V}_{N_j}(\omega) \quad r \geq 0, \omega \in [0, 2\pi), j \in N_0 \quad (2.36.)$$

Burada  $W$  uygun seçilmiş radyal pencere fonksiyonu,  $\tilde{V}_{N_j}$   $2\pi$  ile periyodik açısız pencere fonksiyonlarını ve  $N_j = 4 \cdot 2^{\lfloor j/2 \rfloor}$  ise  $j$  ile temsil edilen çözünürlük seviyesindeki wedge (takoz) sayısını göstermektedir. Bu pencereler çerçeve (frame) oluşturmak için denklem 2.37. ve denklem 2.38.'da ki koşullara her zaman uymalıdır.

$$\sum_{j=-\infty}^{\infty} |W(2^{-j}r)|^2 = 1 \quad r \in [0, \infty) \quad (2.37.)$$

$$\sum_{l=0}^{N-1} \tilde{V}_N^2(\omega - \frac{2\pi l}{N}) = 1 \quad \omega \in [0, 2\pi) \quad (2.38.)$$

Tüm curveletler  $2^j$  ile ölçeklenen  $\phi_j$  ana curveletinin rotasyonu ve ötelenmesiyle elde edilmektedir. Rotasyon açıları,  $\theta_{j,l} := \pi l 2^{-[j/2]}/2 \quad l=0,1,\dots, N_j-1$  aralığında olmak üzere ve konum, öteleme parametreleri  $k_1, k_2 \in \mathbb{Z}$ 'de ve  $\theta$  açısındaki rotasyon matrisi  $R_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$  ile  $x_k^{j,l} := R_{\theta_{j,l}}^{-1}(k_1/2^j, k_2/2^{j/2})$  şeklinde tanımlanmışlardır.

Böylece curvelet fonksiyonu denklem 2.39. ile verilmektedir.

$$\phi_{j,k,l}(x) := \phi_{j,0,0} \left( R_{\theta_{j,l}}(x - x_k^{j,l}) \right) \quad (2.39.)$$

Bu durumda Curvelet katsayıları  $R^2$  tanımlı bir  $f$  fonksiyonu için zaman ve frekans bölgesinde denklem 2.40. ile hesaplanmaktadır.

$$\begin{aligned} c(j, k, l) &:= \langle f, \phi_{j,k,l} \rangle = \int_{R^2} f(x) \overline{\phi_{j,k,l}(x)} dx \\ &= \int_{R^2} \hat{f}(\xi) \overline{\hat{\phi}_{j,k,l}(\xi)} d\xi = \int_{R^2} \hat{f}(\xi) \hat{\phi}_{j,0,0}(R_{\theta_{j,l}}\xi) e^{-i\langle b_k^{j,l}, \xi \rangle} d\xi, \quad j \geq 0 \end{aligned} \quad (2.40.)$$

Frekans bölgesinde, curvelet fonksiyonu  $\hat{\phi}_{j,k,l}$  kutupsal wedge içerisinde  $2^{j-1} \leq r \leq 2^{j+1}$  etki alanıyla desteklenmektedir. Zaman bölgesinde  $2^j$  ile ölçeklemek frekans bölgesinde  $2^j$  ile ölçeklemeye, zaman ve frekans bölgelerinde rotasyonun etkisi aynıken zamanda öteleme yapmak frekans bölgesinde faz kaymasına karşılık gelmektedir.

Dalgacık teorisinde olduğu gibi, curveletlerde kaba ölçek (coarse scale) elemanına sahiptir. Denklem 2.41. sağlayacak şekilde alçak geçiren  $W_0$  penceresi,

$$|W_0(r)|^2 + \sum_{j \geq 0} |W(2^{-j}r)|^2 = 1, \quad (2.41.)$$

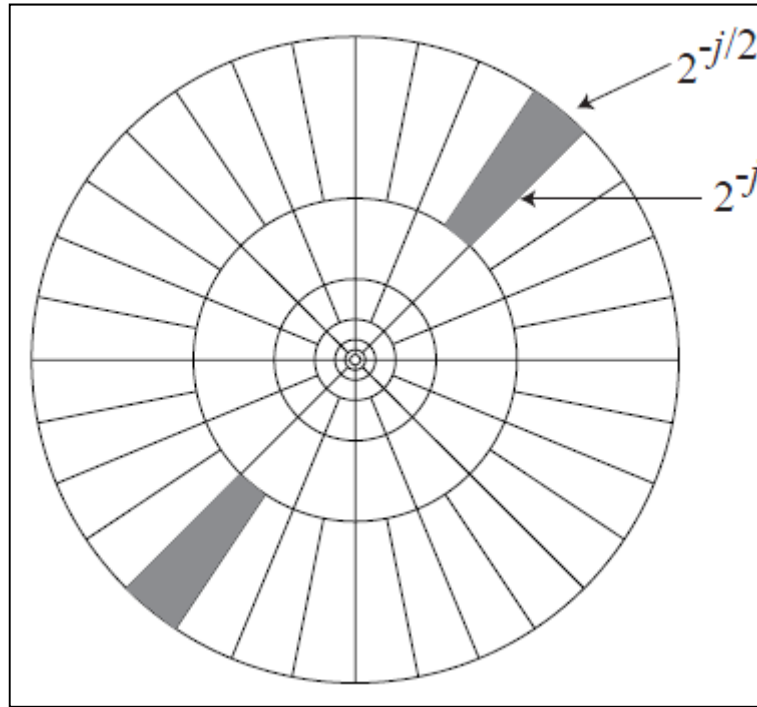


ve  $k = (k_1, k_2) \in \mathbb{Z}^{2\ell}$  de tanımlı olmak üzere coarse scale curvelet,

$$\phi_{j_0, k}(x) = \phi_{j_0}(x - 2^{-j_0}k), \quad \hat{\phi}_{j_0, k}(\omega) = 2^{-j_0}W_0(2^{-j_0}|\omega|), \quad (2.42.)$$

tanımlanmaktadır. Bundan dolayı coarse scale curveletler yönsüzdür. Tüm Curvelet dönüşümü izotropik coarse scale dalgacık (father wavelet) ve yönlü fine scale elemanları içermektedir (Candès *et al.* 2006).

Frekans bölgesinde, curveletler simetrik parabolik wedgelerle desteklenmektedir. Şekil 2.27 frekans düzlemine wedgelerin döşenmesiyle oluşan curvelet yapısını göstermektedir.



**Şekil 2.27.** Frekans bölgesinde curvelet yapısı (Candès *et al.* 2006)

Bu curvelet sistemi bazı özellikleri gerektirmektedir. CurveletG2  $L^2(\mathbb{R}^2)$ 'nin sıkı bir çerçevesini (tight frame) tanımlamaktadır. Bunun anlamı, her fonksiyonun ( $f \in L^2(\mathbb{R}^2)$ )

curvelet serileriyle temsil edilebilmesi yani Curvelet dönüşümünün tersine çevrilebilir bir dönüşüm olmasıdır.

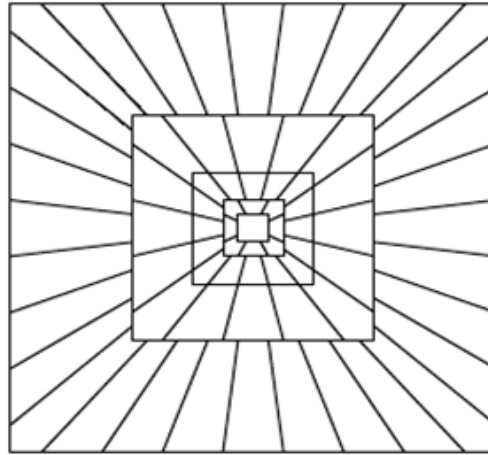
$$f = \sum_{j,k,l} \langle f, \phi_{j,k,l} \rangle \phi_{j,k,l} \quad (2.43.)$$

Bu curveletlerin etkin uzunluğu ve genişliği parabolik ölçekleme ilişkisine uygundur.

$$\text{genişlik} \approx \text{uzunluk}^2 \quad \Rightarrow \quad \text{uzunluk} \approx 2^{-j/2}, \text{ genişlik} \approx 2^{-j} \quad (2.44.)$$

#### 2.4.3.b. 2-D ayrık curvelet dönüşümü

Dijital giriş verileri bir Kartezyen koordinat sisteminde tanımlı olarak örneklenmiştir ve ayrık dönüşüm çıkışta katsayıların toplamı olarak vermektedir. CurveletG2'nin sürekli uzayda tanımı, rotasyonu ve daireleri (coronae) kullanmaktadır ancak bu durum özellikle Kartezyen dizilere uyarlanamamaktadır. Bu kavramları Kartezyen muadilleriyle değiştirmek uygundur. Kartezyen yapı, eşmerkezli daireler yerine eşmerkezli karelere ve rotasyonlar yerine dilimlere (shear) dayanmaktadır. Bu yüzden, pencere fonksiyonlarının yapılarında kutupsal wedgeler yerine yamuklar tercih edilmektedir (Şekil 2.28).



**Şekil 2.28.** Frekans düzleminde ayrık Curvelet dönüşümü (Ma and Plonka 2010)

$f[n_1, n_2]$ ,  $0 \leq n_1, n_2 < n$  formunda verilen giriş Kartezyen dizisinin dijital Curvelet katsayıları  $c^D(j, k, l)$ ;

$$c^D(j, k, l) := \sum_{0 \leq n_1, n_2 < n} f[n_1, n_2] \overline{\phi_{j,k,l}^D[n_1, n_2]} \quad (2.45.)$$

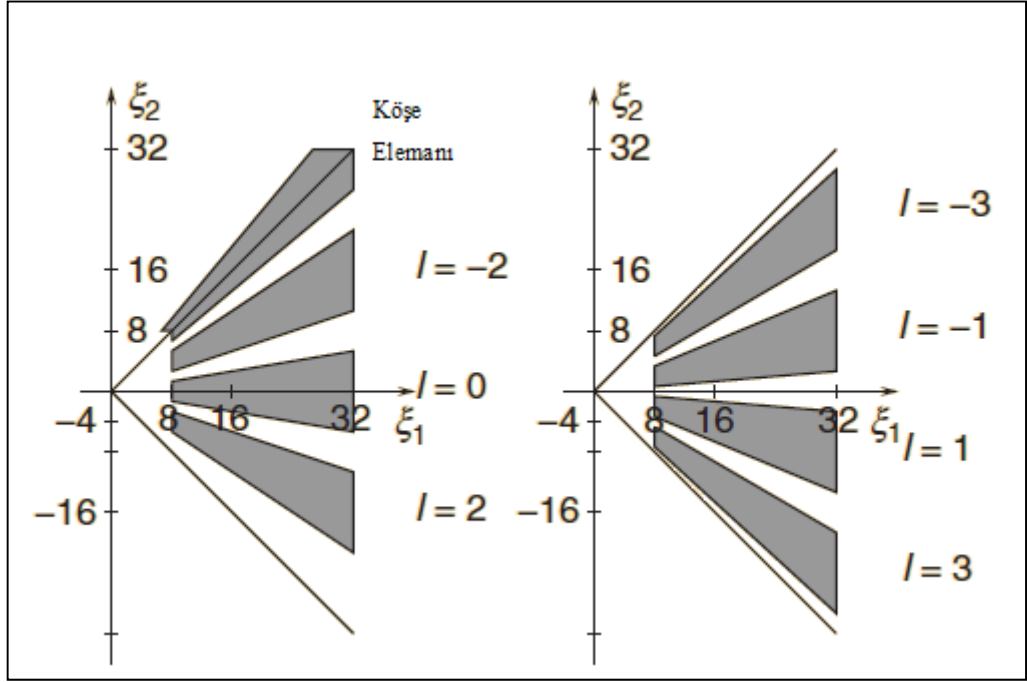
ile tanımlanmaktadır.

Yeni koordinat düzleminde, rotasyonla yer değiştiren dilimler; doğu, batı, kuzey ve güney yönlerinde koni şeklinde yapılar olarak düşünülmektedir. Bu durumda frekans bandı  $\{(\xi_1, \xi_2) : 2^{j-1} \leq \xi_1 \leq 2^{j+1}, -2^{-j/2} \leq \xi_2 / \xi_1 \leq 2^{-j/2}\}$  tanımlı olmaktadır. Doğü konisindeki dilimlemeyi düşünürsek (Şekil 2.29);  $\{(\xi_1, \xi_2) : \xi_1 > 0, -\xi_1 \leq \xi_2 \leq \xi_1\}$  şeklinde frekans düzleminde yerleşmektedir, diğer koniler için uygun curvelet elemanları  $\pm\pi/2$  radyan rotasyonu ve yansımasıyla elde edilmektedir. Eşit uzaklıklı açılar yerine eş aralıklı eğim dizileri doğü konisinde;

$$\tan \theta_{j,l} := l 2^{-|j/2|}, \quad l = -2^{-|j/2|} + 1, \dots, 2^{|j/2|} - 1 \quad (2.46.)$$

şeklinde tanımlıdır. Burada  $\theta$  açısındaki rotasyon matrisi yerine dilimleme matrisi  $S_\theta = \begin{pmatrix} 1 & 0 \\ -\tan \theta & 1 \end{pmatrix}$  oluşturulmakta ve bu durumda  $\tilde{x}_k^{j,l} := S_{\theta_{j,l}}^{-T}(k_1/2^j, k_2/2^{|j/2|})$  şeklinde konum parametresi belirlenmektedir. Böylece curvelet fonksiyonu denklem 2.46. ile verilmektedir.

$$\tilde{\phi}_{j,k,l}(x) := \tilde{\phi}_{j,0,0} \left( S_{\theta_{j,l}}^T (x - \tilde{x}_k^{j,l}) \right) \quad (2.47.)$$



Şekil 2.29.  $\hat{\phi}_{4,k,l}$  fonksiyonunun desteklediği alan (Ma and Plonka 2010)

## 2.5. Görüntü Bölütleme

Görüntü bölütleme; görüntü işleme, görüntü görselleştirme, görüntü tanıma ve görüntü geri kazanım başarısında kritik bir önışleme adıımıdır. Bu alanlardaki başarılı sonuç alma bölütlemenin başarı oranına bağılıdır. Bölütleme görüntüyü birbiriyle örtüşmeyen ve görüntünün tamamını içerecek homojen alt bölgelerine ayırma işleme denilmektedir. Burada amaç; yoğunluk, renk veya doku gibi özellikleri benzer olan ayrık bölgeler belirleyerek hangi veri kısmının bir nesneyi ifade ettiğini bulmaktır. Görüntülerin bu belirli özellikleri yapılacak uygulamaya ve kullanılan görüntüleme bağı olarak deęişiklikler gösterebilmektedir. Örneğin; gri seviyesi görüntüler için parlaklık, renkli görüntüler için ise renk bileşenleri (kırmızı, yeşil ve mavi) önemli özelliklerdir. Ayrıca görüntülerdeki doku (texture) özellięi de bölütleme açısından oldukça yararlı bilgiler içermektedir.

Literatürde, görüntü bölütleme ile ilgili birçok yöntem bulunmasına rağmen tüm görüntülere uygulanabilecek ve mükemmel sonuçlar üretebilecek standart bir yöntem bulunmamaktadır. Uygulamaya özgü temel yöntemler ile bölütleme işlemi gerçekleştirilebilmektedir.

Biyomedikal görüntü bölütlemenin amacı da; ilgili görüntüyü biyolojik yapılara karşılık gelecek şekilde bölgelere ayırmaktır. Görüntülerden elde edilen bu anatomik bölge yapıları; görselleştirme, bilgisayar destekli tanı, teşhis ve morfolojik analiz uygulamalarının en önemli basamağıdır. Anatomik parçaların görselleştirilmesi, anormalliklerin tespitini, dokuların ölçümünü, yüzey tescili için ön işlemeyi, görüntü tescili için ön işlemeyi ve bunların sınıflandırılmasını sağlamaktadır. Burada bölütleme işlemi yapılırken; yoğunluk, doku, gradyent veya benzeri bazı ölçülebilir özelliklerden faydalanılmaktadır.

### **2.5.1. Görüntü bölütleme teknikleri**

#### **2.5.1.a. Eşikleme**

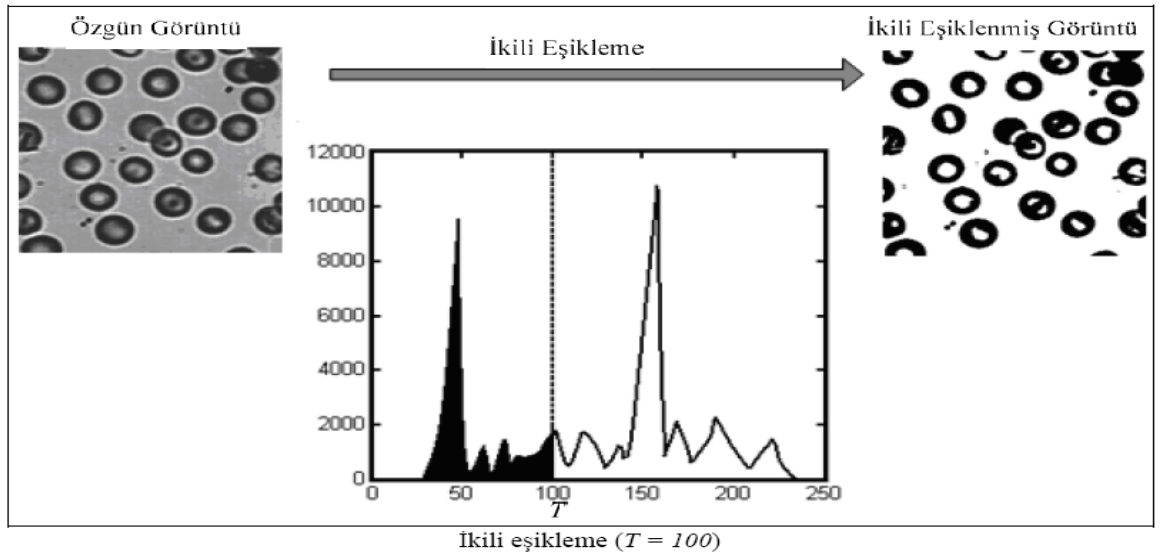
Literatürde en sık rastlanılan yöntemlerinden biridir. Aslında diğer bölütleme teknikleri için de bir ön işleme adımı olarak kullanılabilir. Görüntü eşikleme, hem sayısal görüntü işleme uygulamalarında hem de görüntü tanıma için oldukça önemlidir. Eşikleme; bir görüntünün arka plan ve ön plan bölgelerindeki renk veya yoğunluk farklılıklarına dayanarak görüntü bölütleme gerçekleştiren basit ve kullanışlı bir yöntemdir.

Eşikleme işlemi genelde görüntünün birden fazla özelliğini kullanarak bölütleme gerçekleştiren yöntemlere göre daha hızlı sonuç vermekte ve birçok uygulama alanında kullanılmaktadır. Özellikle biyomedikal görüntülerde parlaklık, bölütleme açısından ayırt edici bir özellik olması nedeniyle eşikleme işleminin kullanılacağı bir

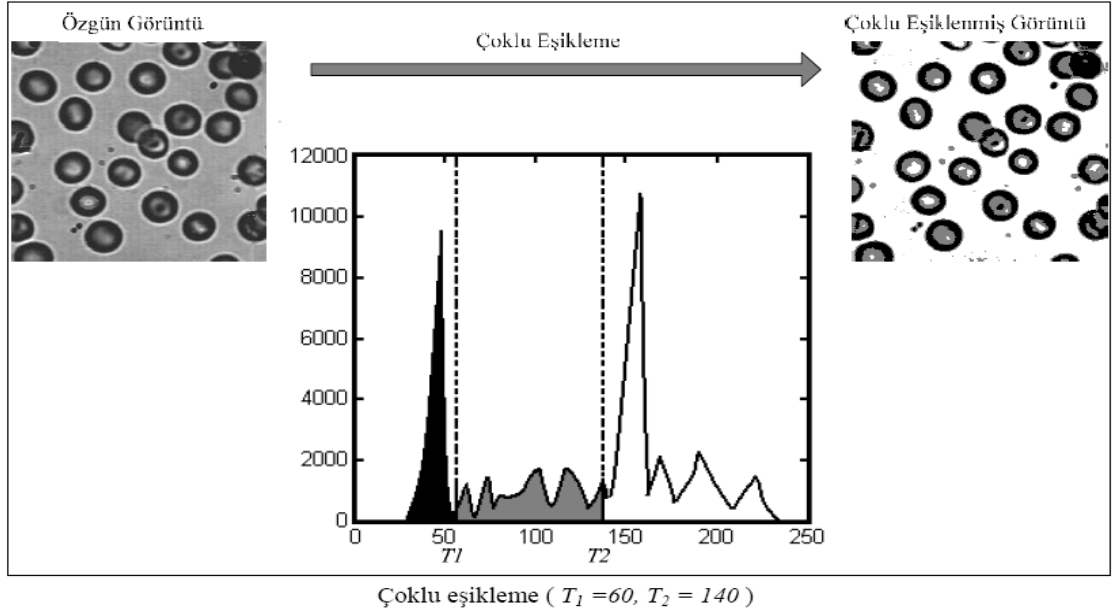
uygulama alanı olmaktadır. Diğer teknikler gibi eşikleme işlemi de beklendiği gibi her görüntü için iyi bölütleme sonucu vermeyip ancak yeterli olduğu uygulamalar da bulunmaktadır.

Piksel tabanlı bir yöntem olan eşikleme için ilk aşamada eşik değerinin belirlenmesi gerekmektedir. Eşikleme için görüntüdeki gri seviye dağılımlarını gösteren histogramlardan faydalanılmaktadır.

İkili eşikleme, görüntünün iki gri seviye grubuna ayrılmasıdır ve bu durumda bir grup nesneyi gösterirken diğeri arka planı göstermektedir. Çoklu eşikleme ise görüntüdeki pikselleri ikiden fazla gri seviye grubuna ayırarak yine ikiden fazla gri seviye ile renklendirmesidir (Şekil 2.30 ve Şekil 2.31).



**Şekil 2.30.** İkili eşikleme (Polat 2007)



**Şekil 2.31.** Çoklu eşikleme (Polat 2007)

### 2.5.1.b. Bileşen etiketleme

Bir görüntüde birden çok bileşen (nesne) bulunması mümkündür. Nesne modeli temelini oluşturulması için tüm bileşenlerin tek tek çıkarılması gerekmektedir. Bir görüntü içerisinde birbiri ile hiçbir piksel komşuluğu olmayan nesnelerin değişik renklere boyanması ile görüntü içerisindeki bu nesnelerin birbirinden ayrılması işlemine etiketleme (labeling) denmektedir.

Arka plan noktaları sıfıra, her bir nesnenin noktaları sıfırdan farklı etikete sahiptir. Taranan görüntüdeki nesneler beyaz, arka plan ise siyah renkte olan görüntüde tarama esnasında beyaz (nesneye ait piksel) bir piksele rastlandığında, bu pikselin tüm komşuluklarına bakılmakta ve komşulukları arasında önceden etiketlenmiş başka bir piksel varsa bu beyaz piksele de aynı etiket atanmaktadır. Eğer bu beyaz pikselin komşuları arasında birbirinden farklı etiketlere sahip birden fazla piksel varsa bu etiketlerden en küçük değerlere sahip olanı bu beyaz piksele etiket değeri olarak atanmaktadır. Aynı zamanda birbirine komşu oldukları halde farklı etiket değerleriyle

etiketlendirilmiş olan piksellere ait etiketler bir eşitlik tablosunda birbirine eşitlenmekte, böylelikle bu piksellerin etiket değerleri farklı olsa da, aslında bu piksellerin aynı bileşenin bir parçası olduğu eşitlik tablosunda belirtilmiş olmaktadır. Eşitlik tablosundaki bu bilgiler bir sonraki tarama işleminde kullanılmak üzere saklanmaktadır. Eğer rastlanan beyaz etiketlenmemiş pikselin komşuları arasında daha önceden etiketlenmiş bir piksel yoksa bu piksele yeni bir etiket değeri atanmakta, bu pikselin yeni bir nesneye ait bir piksel veya daha önceden etiketlendirilmiş bir nesnenin alt çıkıntılarında birine ait bir piksel olduğuna karar verilmektedir. Tarama bu şekilde görüntüyü tamamını kapsayacak biçimde yapılmaktadır. Tarama sonunda eşitlik tablosunda hangi etiketlerin aslında aynı nesneyi temsil ettiği belirlenmekte ve bu yeni veriler göz önünde tutularak görüntü tekrar taranırken eşitlik tablosunda aynı nesneye verilen değerler arasında en küçük değere sahip olan pikselin etiketi, aynı nesnenin tüm piksellerine etiket değeri olarak atanmaktadır. Bu işlem bütün görüntü taranacak şekilde yapılır. Tarama işlemi tamamlandıktan sonra görüntü içerisinde birbirlerine piksel komşulukları olmayan tüm nesnelere farklı bir renge boyanmış, yani etiketlendirilmiş olmaktadır.

#### **2.5.1.c. Nokta süreksizliklerinin belirlenmesi**

Sayısal görüntülerde kullandığımız üç temel gri seviye süreksizliği vardır (Nokta - Çizgi - Kenar, Point - Line - Edge). Süreksizlikleri belirlemenin en bilinen yolu görüntüyü maskelemektir.

Bir görüntüde izole bir noktanın belirlenmesi prensipte basittir. Şekil 2.32'teki gibi bir maske kullanılarak, merkezinde izole bir nokta bulunabilmektedir. Burada T, negatif olmayan bir eşik değeridir. Bu formülasyon merkez noktası ile komşuları arasında ağırlıklı farklılıkları ölçer. İzole nokta çevresinden oldukça farklı olacaktır ve böylece bu tip bir maske ile bu nokta kolayca tespit edilebilmektedir.



$$|R| \geq T$$

-1	-1	-1
-1	8	-1
-1	-1	-1

**Şekil 2.32.** Nokta arama maskesi

Nokta belirlemede diğer bir yaklaşım ise, belirli bir T eşik değerini aşan maksimum ve minimum piksel değerleri arasındaki fark için  $m \times n$  boyutundaki tüm komşuluklarındaki noktaları tespit etmektir.

#### 2.5.1.d. Çizgi süreksizliklerinin belirlenmesi

Şekil 2.33'teki ilk maskeyi görüntüye uyguladığında daha çok yatay doğrultuda bir piksel kalınlığında doğrulara daha kuvvetli tepki vermektedir. Sabit bir arka plan ile çizgi, maskenin orta satırından geçtiği zaman, maksimum tepkiyi verecektir. Benzer şekilde ikinci maske,  $+45^\circ$  doğrultusunda uzanan hatta çok tepkiyi verirken; üçüncü maske düşey çizgilere ve dördüncü maske de  $-45^\circ$  diyagonal uzanan çizgilere tepki vermektedir.

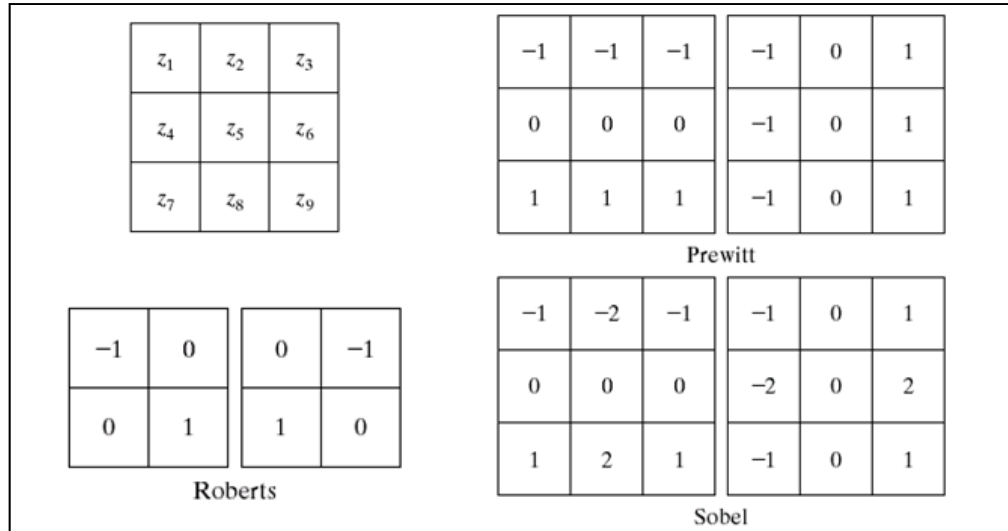
<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>2</td><td>2</td><td>2</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table> <p>Yatay</p>	-1	-1	-1	2	2	2	-1	-1	-1	<table border="1"> <tr><td>-1</td><td>-1</td><td>2</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>2</td><td>-1</td><td>-1</td></tr> </table> <p><math>+45^\circ</math></p>	-1	-1	2	-1	2	-1	2	-1	-1	<table border="1"> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> </table> <p>Düşey</p>	-1	2	-1	-1	2	-1	-1	2	-1	<table border="1"> <tr><td>2</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>2</td></tr> </table> <p><math>-45^\circ</math></p>	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	-1																																					
2	2	2																																					
-1	-1	-1																																					
-1	-1	2																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	2	-1																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	-1	2																																					

**Şekil 2.33.** Çizgi arama maskeleri

### 2.5.1.e. Kenar süreksizliklerinin belirlenmesi

Kenar, bir görüntüdeki bir nesne ile diğer bir nesneye arasındaki geçiş bölgesine denilmektedir. Kenar taramanın temel ilkesi, görüntüdeki her pikselin komşularıyla arasındaki parlaklık değerlerinin karşılaştırılması esasına dayanmaktadır. Bu parlaklık değerindeki süreksizlikler, birinci ve ikinci dereceden türevler kullanılarak tespit edilmektedir. Kenar tarama amacıyla kullanılan filtrelerin çoğu görüntülerdeki gürültü noktalarından olumsuz yönde etkilenir. Bu nedenle, kenar taramadan önce gürültülerin olabildiğince ayıklanması gerekir.

Örneğin; eğer bir pikselin gri düzey değeri komşularının gri düzey değerlerine eşit ya da yakınsa, bu bölgede büyük ihtimalle bir kenar bulunmayacaktır. Ancak, pikselin gri düzey değerleriyle komşuları arasında belirgin bir fark varsa bu durumda bu noktada bir kenarın varlığından söz edilebilmektedir. İdeal olarak bir kenar ani ve belirgin gri düzey farkıyla ortaya çıkacaktır. Ancak pratikte böyle ani değişimler yerine, yavaşça değişen renk tonları oluşmaktadır.



**Şekil 2.34.** En çok kullanılan filtreler; Roberts, Prewitt, Sobel

Canny algoritması, en sık kullanılan kenar bulma yöntemidir. Görüntü, önce Gauss maskesinin türeviyle filtrelenir, sonra görüntü içerisindeki gürültü elendikten sonra filtrelenmiş görüntüye Canny operatörü uygulanmaktadır. Genellikle Canny operatörü uygulandıktan sonra elde edilen kenar görüntüsü istenmeyen kenar bilgileri içermektedir. İstenmeyen bu bilgileri elemek için maksimum olmayan noktaların bastırılması işlemi uygulandıktan sonra biri büyük, diğeri küçük iki eşik değeri tanımlanır. Yüksek eşik değerinden yararlanılarak kalın kenar eğrileri belirlenir; düşük eşik değerinden istifade edilerek eğriler devam ettirilmektedir.

### 2.5.1.f. Bölge tabanlı bölütleme

Bir görüntüde bölge, benzer özellikler ile bağlantılı pikseller topluluğu olarak tanımlanmaktadır. Bölgeler bir görüntünün yorumlanmasında, görüntüdeki nesnelere olan ilişkisinden dolayı önemlidir. Bir görüntü birkaç nesne içerebilmekte ve her nesnenin farklı bölümlerinde birkaç bölge bulunabilmektedir. Bir görüntünün doğru bir şekilde yorumlanabilmesi için görüntü nesnelere ya da nesnelere bölümlerine ayrılmalıdır.

Bölgeler belirlenen bir özelliğe göre bölümlenebilmektedir. Örneğin, bu özellik bölgedeki tüm piksellerin aynı parlaklık değerine sahip olması olabilir. Giriş görüntüsü  $R$  ise, alt bölgeler  $R_1, R_2, R_3, \dots, R_n$  olarak gösterilebilir. Her piksel tanımlı bir bölgeye dahil olmalıdır ve bölütleme işlemi tamamlandığında piksel, bir bölgede gösterilmelidir ( $\cup_{i=1}^n R_i$ ) ve bir bölgedeki noktalar bağlanmalıdır ( $R_i, i=1,2,\dots,n$ ). Ayrıca bu bölgeler ayrık olmalıdır ( $R_i \cap R_j = \emptyset, i \neq j$ ).

Bölge temelli yaklaşımlar bölge ayırma, bölge birleştirme ve bölge geliştirme algoritmalarının ya da bunların bir arada kullanılarak sayısal görüntülere uygulanmasını içermektedir.

### 2.5.1.g. Snake modeli

İlk olarak Michael Kass, Adrew Witkin ve Demetri Terzopoulos tarafından ele alınmış ve 1988 yılında literatüre girmiştir. Snake (yılan) modeli yakaladığı bir noktanın izini sürebilme özelliği vardır, ismi de buradan gelmektedir. Görüntüde yer alan nesneyi bulmak için ilk olarak kullanıcı tarafından noktalar belirlenmekte ve daha sonra kullanılan algoritma bu noktaları referans alarak çalışmakta ve gerekli bölütleme gerçekleştirilmektedir.

Parametrik olarak modellenen iki boyutlu kıvrımlar denklem 2.48.'deki gibi ifade edilmektedir.

$$\vec{v}(s, t) = (x(s, t), y(s, t)) \quad , \quad 0 \leq s \leq 1 \quad (2.48.)$$

Burada, s orantılı çizgi uzunluğu, x ve y kıvrımın görüntü koordinatlarıdır. Algoritma iteratif olarak zaman ilerledikçe konumunu şeklini etkileyen kuvvetlere göre kendini deforme etmektedir. Bu kuvvetler iç kuvvetler ve dış kuvvetler olmak üzere ayrılmaktadır. İç kuvvetler parçalar halinde yumuşaklık kısıtlarını zorlayarak algoritmanın şeklini kontrol etmektedir. Belirlenen şeklin yumuşak hatlara sahip olup olmamasına göre belirlenmektedir. Keskin hatlar olduğunda iç enerji yüksektir ve istenilen bir durum değildir. Dış kuvvetler, algoritmanın ilgilenilen bölgeye en yakın kenar noktasına (yerel minimuma) hareketinden sorumludur. Görüntüden türetilmektedirler. Algoritmayı kenarlar gibi görüntü detaylarına yönlendirmektedir. Görüntünün gradyanı alınarak hesaplanmaktadır. Görüntünün gradyanı yüksek çıkarsa, dış enerji düşüktür, çünkü belirlenen nokta için komşu piksellerdeki gri seviyeleri ile olan farkı yüksek demektir ve farkın yüksek olduğu bu noktalarda sınırlar yer almaktadır. Yılan modeli enerji minimizasyonu prensibine göre çalışmaktadır ve kuvvetlerin dengelendiği yerde algoritmanın konumu toplam yerel minimumuna karşılık gelmektedir.

$$E(\vec{v}) = E_{i\check{c}}(\vec{v}) + E_{dış}(\vec{v}) \quad (2.49.)$$

İç enerji algoritmanın şekil üzerindeki geometrik kısıtlamaları yerine getirmesine olanak sağlamaktadır ve denklem 2.50.'teki gibi tanımlanmaktadır.

$$E_{i\check{c}}(\vec{v}) = \frac{1}{2} \int_0^1 \alpha(s) \left| \frac{\partial \vec{v}(s,t)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 \vec{v}(s,t)}{\partial s^2} \right|^2 \quad (2.50.)$$

Burada  $\alpha$  gerilim (tension) katsayısıdır ve integral içindeki ilk terim algoritmanın zar gibi davranmasını;  $\beta$  katılık (rigidity) katsayısıdır ve ikinci terim algoritmanın ince bir tabaka gibi davranmasını sağlamaktadır.

Dış enerji (denklem 2.51.), algoritmayı kenarlara çekerken genellikle verilen gri seviyeli ham görüntünün veya bu görüntünün gauss filtresi uygulanmış halinin gradyentini (denklem 2.52. veya denklem 2.53.) uygulamaktadır.

$$E_{dış}(\vec{v}) = - \int_0^1 P(\vec{v}(s, t)) ds \quad (2.51.)$$

$$P(\vec{v}(s, t)) = |\nabla I(\vec{v}(s, t))|^2 \quad (2.52.)$$

$$P(\vec{v}(s, t)) = |\nabla [G_\sigma(\vec{v}(s, t)) * I(\vec{v}(s, t))]|^2 \quad (2.53.)$$

### 2.5.1.h. Gradyent vektör akışı

Yılan modeli parametrik aktif kontur algoritmasıdır ve iki önemli zorluğu bulunmaktadır. Birincisi, başlangıç konturu genel olarak geçerli sınıra yakın olmalıdır yoksa muhtemelen sonuç yanlış olacaktır. Literatürde bu problemi çözmek için mesafe

potansiyeller ve basınç kuvvetleri gibi birkaç metot önerilmiştir ve temel fikir dış kuvvetin yakalama aralığını artırarak istenilen sınıra konturu yönlendirmektir. İkinci problem olarak içbükey alanlarda tatmin edici sonuçlar verememektedir.

Bu her iki sorunu çözen yeni bir statik dış kuvvet tanımlanmıştır. Buna gradyent vektör akış alanı denilmiş ve bir varyasyonel çerçevede belirli bir enerji fonksiyonunu minimize ederek görüntülerden türetilen yoğunluk vektör alanı olarak tanımlanmıştır. Minimizasyon bir çift doğrusal kısmi diferansiyel denklemin çözümü tarafından sağlanmaktadır.

Burada yılanın başlangıç konturu görüntünün içinde, dışında ya da kenarları üzerinde olabilmektedir. GVF (Gradient Vector Flow) yılanın da yakalama aralığı büyüktür, bunun anlamı, diğer nesnelere girişimi engelleyebilmekte ve kontur sınırdan uzakta başlatılabilmektedir.

$\mathbf{v}(x,y)=[u(x,y),v(x,y)]$  vektör alanı için enerji fonksiyonunu minimize eden Gradyent vektör akış alanı tanımlanmaktadır.

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dx dy \quad (2.54.)$$

Bu varyasyonel formülasyon herhangi bir veri olmadığı zamanda düzgün sonuç verecek standart bir prensip izlemektedir. Burada  $\mu$  düzenlileştirme (regularization) parametresi,  $f(x,y)$  kenar haritasını temsil etmektedir.

### 3. MATERYAL ve YÖNTEM

Curvelet dönüşümü iyi bir potansiyele sahip çok genç bir sinyal analiz yöntemidir. Bu dönüşüm, görüntü işleme ve diğer uygulamalarda bir kilometre taşı olarak kabul edilebilir. Curvelet dönüşümü, neredeyse nesnelere temsiliyi optimum sağlayan bir çok ölçekli yönlü dönüşümdür.

Dalgacık ve Ridgelet dönüşümlerinin yetersiz kaldığı süreksizlikleri görüntülerde belirlemektedir. Çünkü Curvelet dönüşümü yönlü özellikleri belirlemek için frekans bölgesinde açılı polar wedgeleri (takozlar) veya açılı yamuk pencereleri kullanmaktadır. Her ölçek seviyesinde  $N_j$  sayıda bulunan bu wedgeler o ölçekteki belli açılarda incelenen görüntünün katsayılarını tutmaktadır. Görüntü farklı açılar altında incelendiğinde görüntüye ait bilgilere daha doğru ve net bir şekilde sahip olunmaktadır. Ayrıca bu bilgileri aynı ölçüde elde edebilecek şekilde modifiye edilen herhangi bir çoklu çözünürlük analiz dönüşümünden daha az katsayı ile görüntü bilgileri elde edilmektedir. Ve Curvelet dönüşümün daha net olan geometri özelliği anlaşılabilirliğini arttırdığı için kullanılmaktadır.

Bu çalışmada, Curvelet dönüşümünün yeni yapılandırılmış ve geliştirilmiş versiyonu ikinci nesil Curvelet dönüşümü kullanıldı. Bu teknik, ridgeletlere dayalı orijinal Curvelet dönüşümünden (CurveletG1) daha basit ve hızlıdır. Ayrıca daha az gereksiz bilgi içermektedir. Bu sebeple, çalışmada DCTG2 gerçekleştiren bir kaynak MATLAB kodu kullanıldı (<http://www.curvelet.org/software.html>). Bu şekilde görüntünün farklı ölçeklerde ve her ölçekte de farklı açılarla katsayıları elde edildi. Katsayılar ve katsayı grubunun açısı, MATLAB hücreleri içine ilgili ölçek seviyesine göre yerleştirildi. Elde edilen katsayılardan görüntünün kenarlarına ait bilgilerini belirlemek amaçlandı. Bu amaçla, her ölçekte her farklı açıdaki katsayı grubu incelendi ve incelenen grup içerisindeki maksimum katsayıyla gruplar kendi içlerinde önce normalize edildi daha sonra bu normalize değerlere bir eşikleme işlemi uygulandı. Eşikleme sonucu

katsayıların gerçek görüntüde buldukları koordinatlar tespit edildi. Bu koordinatlarda ilgili ölçek seviyelerine göre hücreler kullanılarak saklandı.

Görüntüye ait kenar bilgilerini içeren bu katsayıların kullanımıyla bölütleme tekniklerinden geleneksel Snake algoritması performansının artırılabilceği bir uygulama tasarlandı. Bu amaçla, Snake algoritması için kaynak bir MATLAB kodu kullanıldı (<http://www.mathworks.com/matlabcentral/fileexchange/28149>). Curvelet katsayılarıyla birleştirilen snake algoritmasının performansı GVF algoritmasıyla karşılaştırılarak değerlendirildi.

Geleneksel snake algoritması başlangıç konturuna bağımlıdır; bu kontur görüntü kenarları içinde kalacak şekilde yerleştirildiğinde snake eğrisini görüntü dışına çekecek herhangi bir kuvvet bulunmamaktadır. Bu sorunu çözmek için literatürde kullanılan balon kuvvetleri gerektiğinde aktive olacak şekilde kaynak MATLAB algoritması içerisinde yer almaktadır. Ayrıca geleneksel snake içbükey bölgelerde kenarları belirlemede yetersiz kalmaktadır.

GVF algoritması ise başlangıç koşullarına duyarlı değildir, herhangi bir ön bilgi olmadan kenarları belirlemektedir ve algoritma snake eğrisini içbükey noktalarda içeri çekecek bir dış kuvvet geliştirilerek tanımlanmıştır. Ancak iteratif olarak çalışan GVF algoritması başlangıç koşullarına bağılı olarak çok yüksek sonlanma süreleri ile sonuç verebilmektedir.

Bu çalışmada tasarlanan snake uygulamasıyla amaçlanan verimli bir şekilde otomatik kenar belirleme sağlayan bir model oluşturmaktır. Kenarlara ait bilgileri içerdiği belirlenen Curvelet katsayılarının koordinat verileri; kendisine en yakın mesafedeki bir diğer noktaya götüren sıralama algoritması kullanılarak ve seçilen yarıçap parametresiyle görüntü kenarlarının dışından, balon kuvvetleri kullanıldığında içerisinde, başlatacak şekilde görüntü üzerine yerleştirilirdi.



Kaynak kod kullanılarak görüntüye yerleştirilen snake kontur noktalarının 100 noktaya interpolasyonu sağlandı ve enerji minimizasyonu prensibine bağlı olarak iç ve dış kuvvetlerin enerjileri hesaplanarak kenarlar belirlendi. Bu algoritma kullanılan test imgesine göre çeşitli iterasyonlarda çalıştırıldı.

Kullanılan kaynak kod aynı zamanda GVF algoritmasını da aktif edebilecek şekilde tasarlanmıştır. GVF algoritmasının herhangi bir ön bilgiye ihtiyaç duymadan kenarları algılama özelliğine sahip bir yapısı olduğu için; kaynak kod çalıştırırken bu durum göz önünde bulundurularak başlangıç konturu görüntünün boyutlarına uygun olarak dışarısına en küçük yarıçaplı olacak şekilde yerleştirildi. Bu algoritma da kullanılan test imgesine göre çeşitli iterasyonlarda çalıştırıldı.

Geleneksel snake ve GVF snake uygulanırken her iki algoritma içinde programa aynı gerilim (tension) ve katılık (rigidity) parametreleri verildi. Her test imgesi için iki algoritmanın uygulanan iterasyonda görüntü kenarlarına olan yaklaşıklıklarıyla hata oranları tespit edilerek performansları karşılaştırıldı. Başlangıç konturu de dahil olmak üzere her 100 iterasyonda bir gelinen noktalarla referans olarak belirlediğimiz noktalar karşılaştırıldı.

Aynı zamanda geleneksel Snake algoritması Curvelet katsayıları kullanılmadan otomatik kenar belirleme sağlamak adına GVF algoritmasında belirlenen başlangıç konturuyla çalıştırılarak uygulanan yöntemin geleneksel algoritmayla karşılaştırıldığında geldiği nokta gösterildi.

#### 4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Çalışmada farklı boyutlardaki dört ayrı test imgesi üzerine sırasıyla geleneksel Snake algoritması, GVF algoritması ve Curvelet dönüşümü ile geliştirilen Snake algoritması uygulandı. Her imgeden her algoritmanın Snake hareketlerine, nesnenin referans kenarlarına, Snake başlangıç ve bitiş konturuna ait görüntülerle birlikte imgenin Curvelet katsayılarını içeren görüntüler alındı.

Bu çalışmada Snake modeli ile iç enerji hesaplanırken gerilim ve katılık parametreleri, sırasıyla alfa ve beta, her imgede tüm algoritmalar için 0,02 olarak kullanıldı.

Üzerinde çalışılan bu dört imge için her algoritmanın 1 000 iterasyonla çalıştırılmasıyla elde edilen sonuçlar değerlendirildi. Ayrıca her imgede uyulanan her bir algoritmanın, görüntü kenarı belirlemede oluşan hataların iterasyon adım sayısına göre performanslarının değerlendirilmesi yapıldı.

Buna göre; Şekil 4.1 ile birinci test görüntüsü; Şekil 4.2, Şekil 4.3 ile geleneksel Snake algoritması sonuçları; Şekil 4.4, Şekil 4.5 ile GVF algoritması sonuçları; Şekil 4.6 ile ilgili test görüntüsünün Curvelet katsayıları; Şekil 4.7, Şekil 4.8 ile bu çalışmada uygulanan Curvelet-Snake algoritması sonuçları ve Şekil 4.9 ile bu üç farklı yaklaşımın performansının karşılaştırıldığı hata-iterasyon adımı grafiği verilmiştir.

Benzer şekilde; Şekil 4.10 ile ikinci test görüntüsü; Şekil 4.11, Şekil 4.12 ile geleneksel Snake algoritması sonuçları; Şekil 4.13, Şekil 4.14 ile GVF algoritması sonuçları; Şekil 4.15 ile ilgili test görüntüsünün Curvelet katsayıları; Şekil 4.16, Şekil 4.17 ile bu çalışmada uygulanan Curvelet-Snake algoritması sonuçları ve Şekil 4.18 ile bu üç farklı yaklaşımın performansının karşılaştırıldığı hata-iterasyon adımı grafiği verilmiştir.

Ayrıca; Şekil 4.19 ile üçüncü test görüntüsü; Şekil 4.20, Şekil 4.21 ile geleneksel Snake algoritması sonuçları; Şekil 4.22, Şekil 4.23 ile GVF algoritması sonuçları; Şekil 4.24 ile ilgili test görüntüsünün Curvelet katsayıları; Şekil 4.25, Şekil 4.26 ile bu çalışmada uygulanan Curvelet-Snake algoritması sonuçları ve Şekil 4.27 ile bu üç farklı yaklaşımın performansının karşılaştırıldığı hata-iterasyon adımı grafiği verilmiştir.

Son olarak; Şekil 4.28 ile dördüncü test görüntüsü; Şekil 4.29, Şekil 4.30 ile geleneksel Snake algoritması sonuçları; Şekil 4.31, Şekil 4.32 ile GVF algoritması sonuçları; Şekil 4.33 ile ilgili test görüntüsünün Curvelet katsayıları; Şekil 4.34, Şekil 4.35 ile bu çalışmada uygulanan Curvelet-Snake algoritması sonuçları ve Şekil 4.36 ile bu üç farklı yaklaşımın performansının karşılaştırıldığı hata-iterasyon adımı grafiği verilmiştir.

Sonuçta geleneksel Snake algoritmasının içbükey noktalardaki zayıflığı giderildi ve GVF algoritmasından daha hızlı ve doğru kenarları belirleyen bir algoritma geliştirildi.

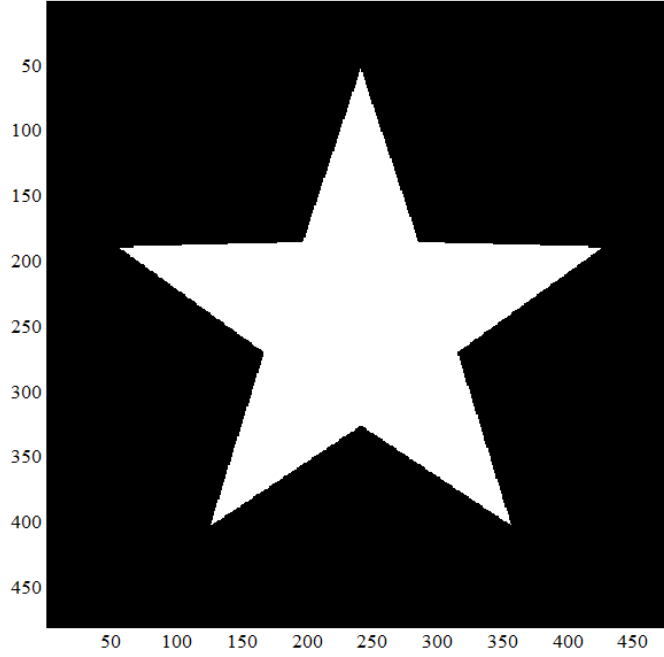
GVF algoritmasıyla karşılaştırdığımızda, geliştirilen yöntemde geleneksel Snake algoritması kullanıldığı için; başlangıç konturunun kenarların içinden başlatıldığı durumlarda yetersiz kalacağını ön görerek geleneksel Snake modelinin bu zayıflığını yenmek için geliştirilen ve literatürde balon kuvveti adıyla yer alan kuvvetten de yararlandı. Başlangıç konturlarının dışarı doğru hareketi sağlanarak bu eksik giderilmiş oldu.

Buna göre; Şekil 4.37, Şekil 4.38 ile geleneksel Snake algoritması sonuçları; Şekil 4.39, Şekil 4.40 ile GVF algoritması sonuçları Şekil 4.41, Şekil 4.42 ile bu çalışmada uygulanan Curvelet-Snake algoritması sonuçları ve Şekil 4.43 ile bu üç farklı yaklaşımın performansının karşılaştırıldığı hata-iterasyon adımı grafiği verilmiştir.

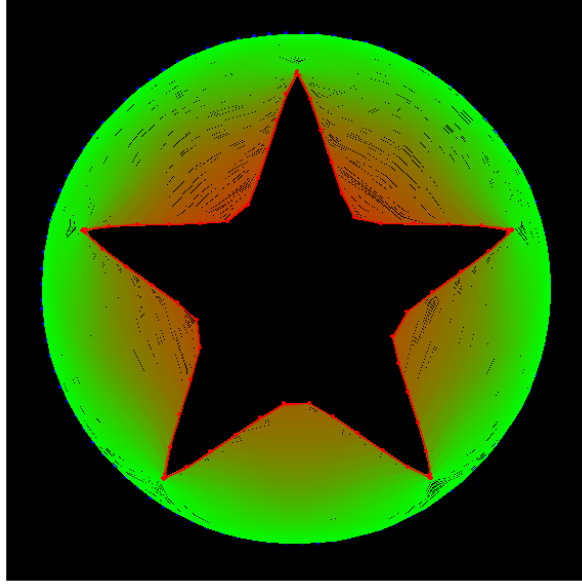
Her imge için öncelikle geleneksel Snake algoritması uygulandı. Testimge\_2 imgesindeki içbükey bölgedeki noktaların belirlenmesi haricinde 500 ile 900 iterasyon

arasında kenarlara başarılı sayılabilecek bir derecede ulaşıldı. GVF algoritmasıyla 300 ile 600 iterasyon arasında sonuca varılırken uyguladığımız yöntemle daha az hata oranıyla 100 iterasyonda başarıml sağlandı.

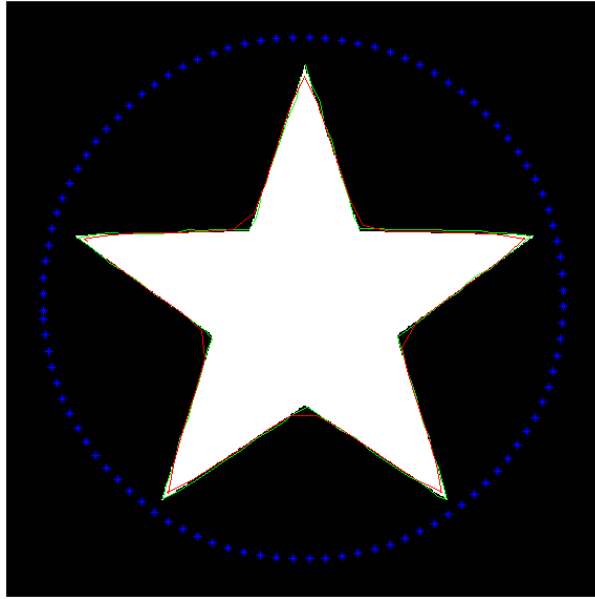
MATLAB “tic toc” hazır komutuyla algoritmaların süreleri 4GB RAM, Intel Core Duo P8700 2,53 GHz ve 32 bit işletim sistemli dizüstü bilgisayarla hesaplatıldı. Geleneksel Snake algoritmasının yaklaşık olarak 90 ila 96 sn arasında iterasyonu tamamladığı, Curvelet katsayılarının yaklaşık olarak 0,86 ila 1,9 sn arasında hesaplandığı ve aslında geliştirilen yöntemle 100 iterasyonla sonuca ulaşıldığı halde 1 000 iterasyonu tamamlamak için geçen sürenin 87 ila 96 sn arasında olduğu, GVF algoritmasının ise yaklaşık olarak 150 ila 170 sn arasında iterasyonu tamamladığı gözlenmiştir. Çünkü geliştirdiği statik kuvveti hesaplamak için algoritmanın uygulanan imgeler için yeterli görülen 600 iterasyonu daha kendi içinde hesaplariken gerçekleştirilmesi gerekmektedir.



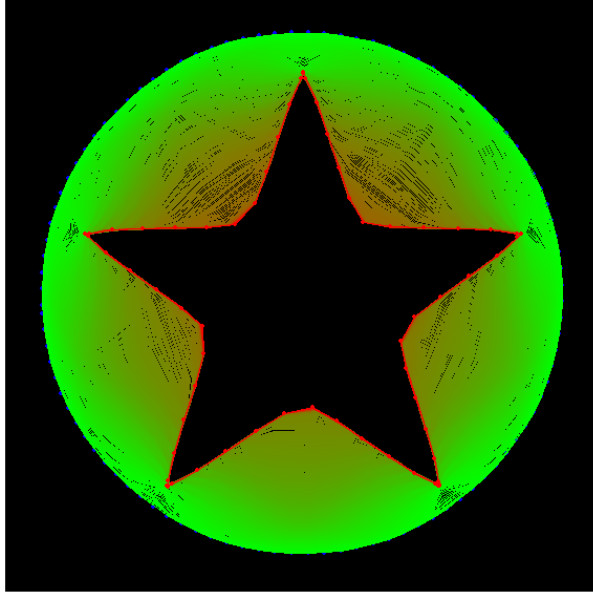
**Şekil 4.1.** Testimge\_1 orijinal görüntü (480×480)



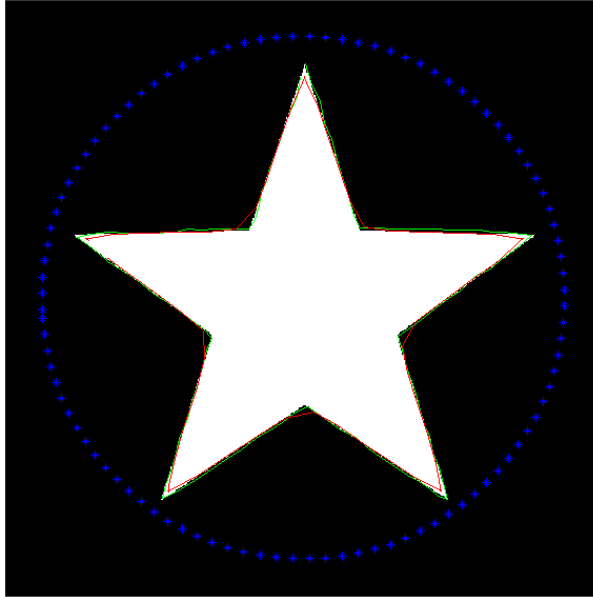
**Şekil 4.2.** Testimge\_1 geleneksel Snake algoritmasında Snake hareketi



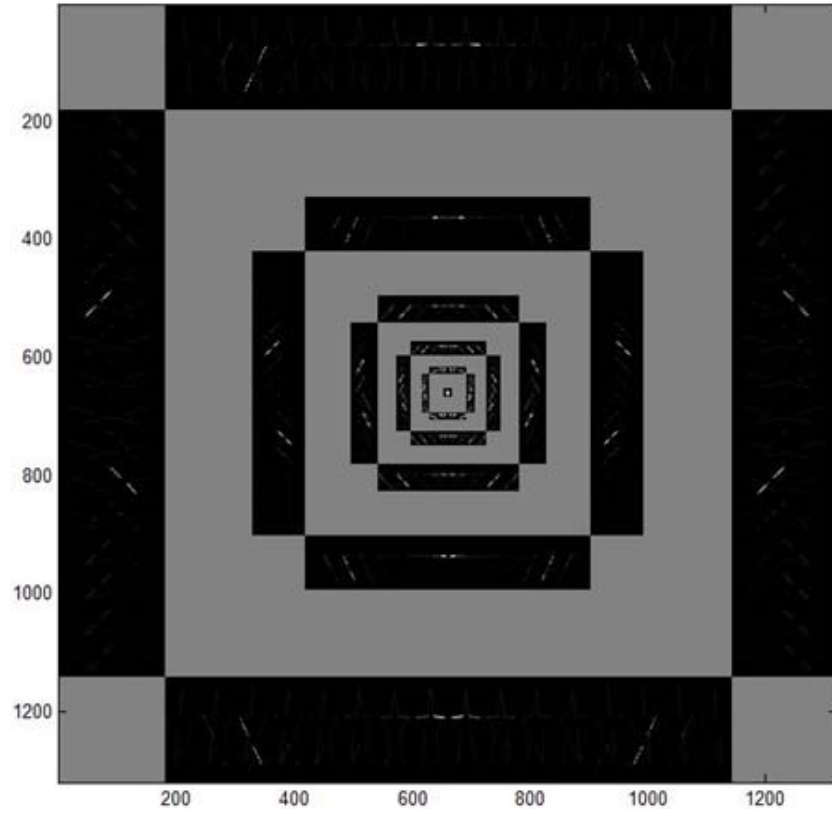
**Şekil 4.3.** Testimge\_1 geleneksel Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



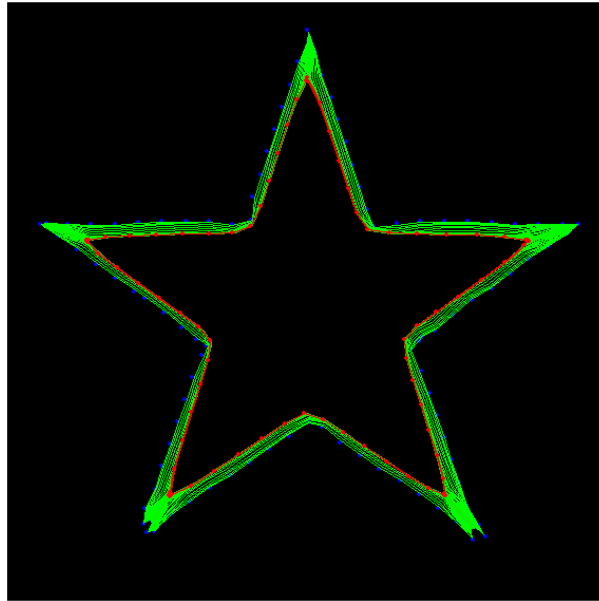
Şekil 4.4. Testimge\_1 GVF algoritmasında Snake hareketi



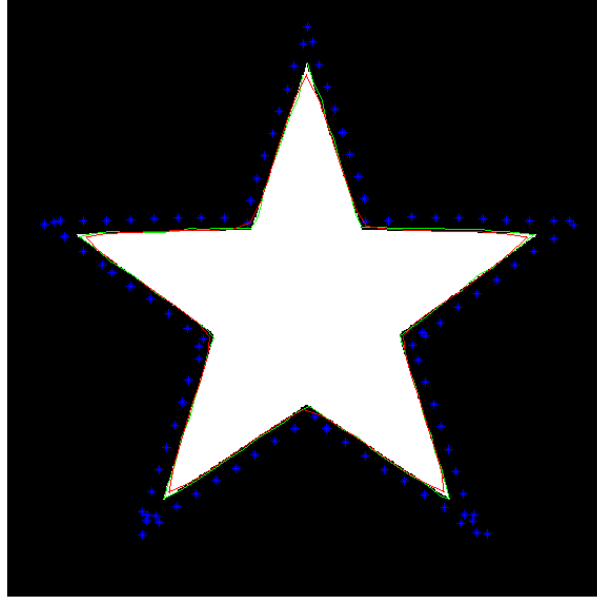
Şekil 4.5. Testimge\_1 GVF algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



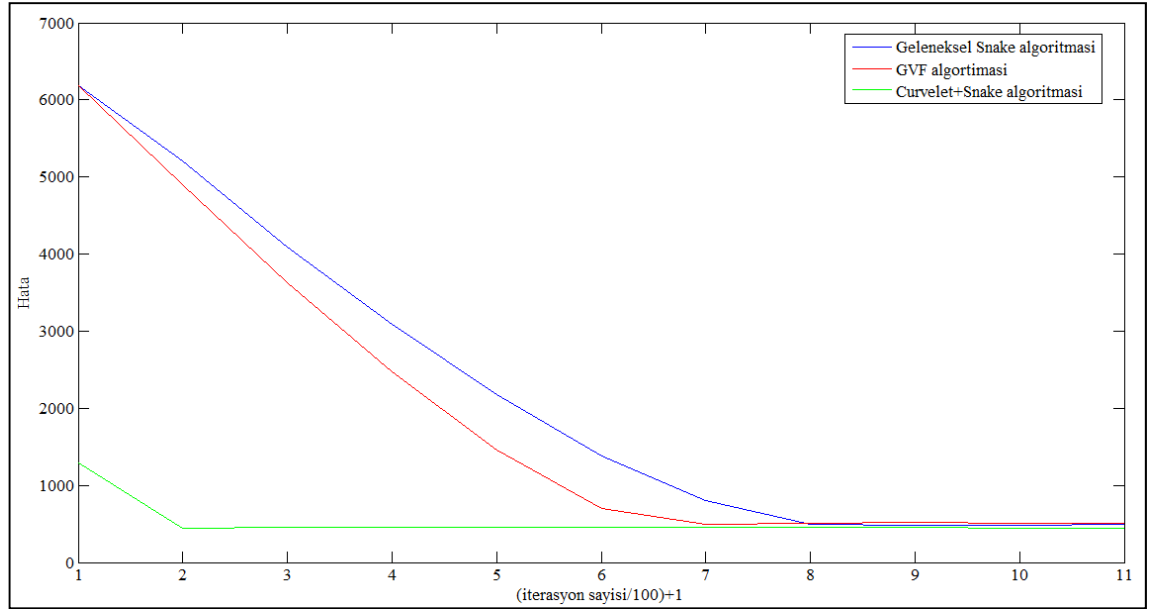
Şekil 4.6. Testimge\_1 Curvelet katsayıları



Şekil 4.7. Testimge\_1 Curvelet katsayıları uygulandığında Snake hareketi

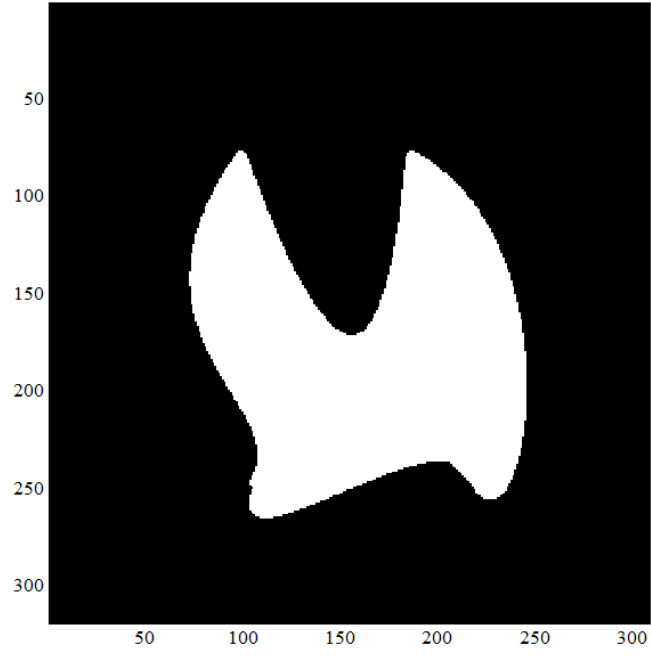


**Şekil 4.8.** Testimge\_1 Curvelet katsayıları ile Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)

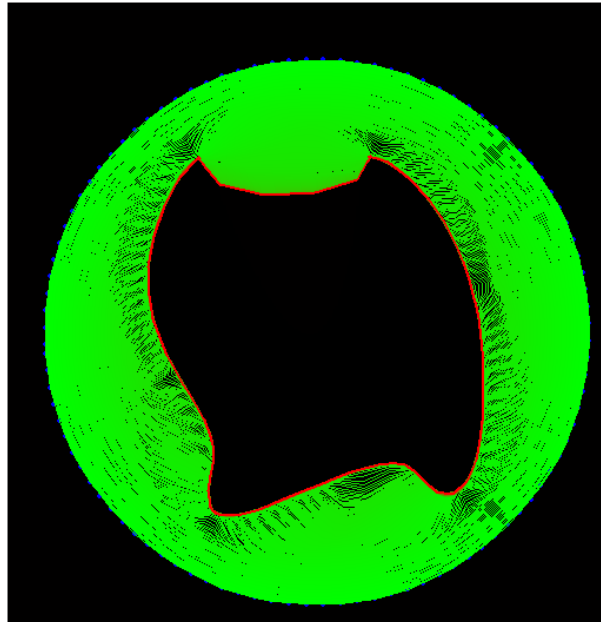


**Şekil 4.9.** Testimge\_1 iterasyon adımları için kenar algılama hata grafiği

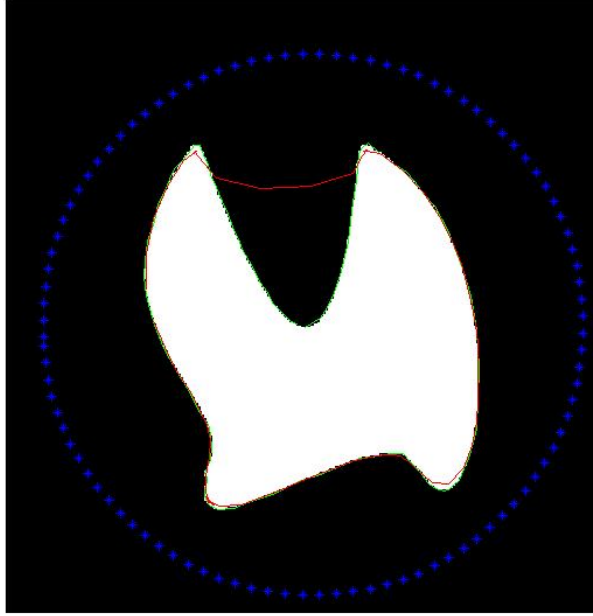




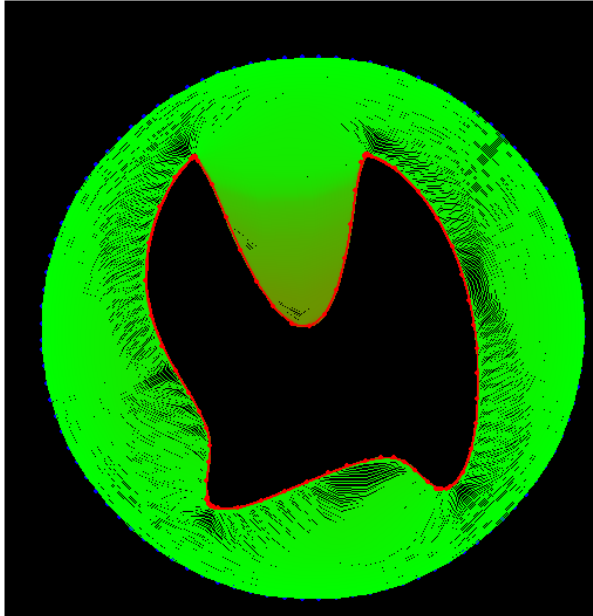
Şekil 4.10. Testimge\_2 orijinal görüntü (319×309)



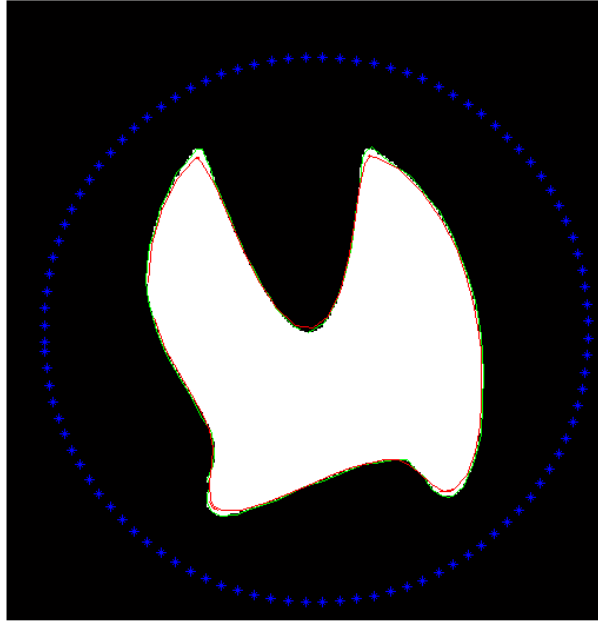
Şekil 4.11. Testimge\_2 Geleneksel Snake algoritmasında Snake hareketi



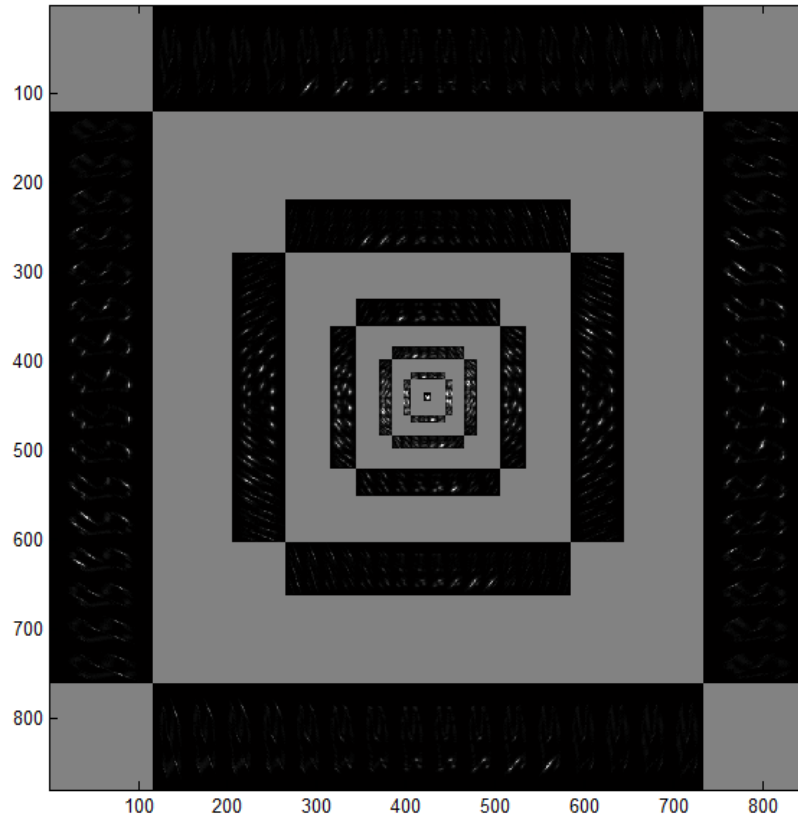
**Şekil 4.12.** Testimge\_2 Geleneksel Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



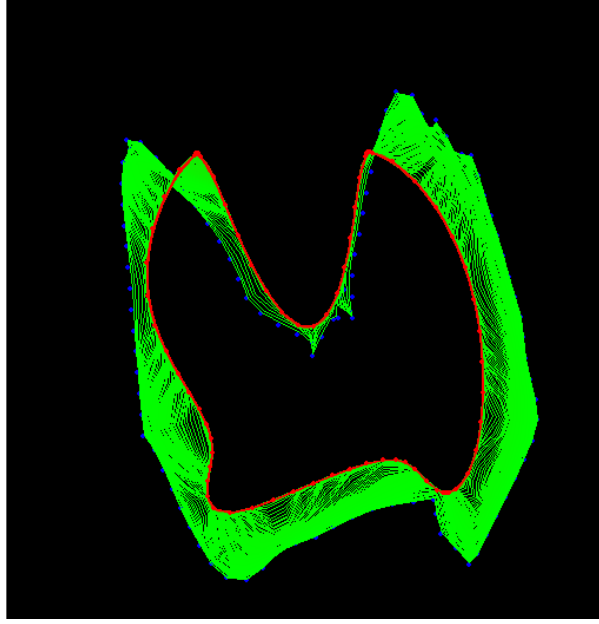
**Şekil 4.13.** Testimge\_2 GVF algoritmasında Snake hareketi



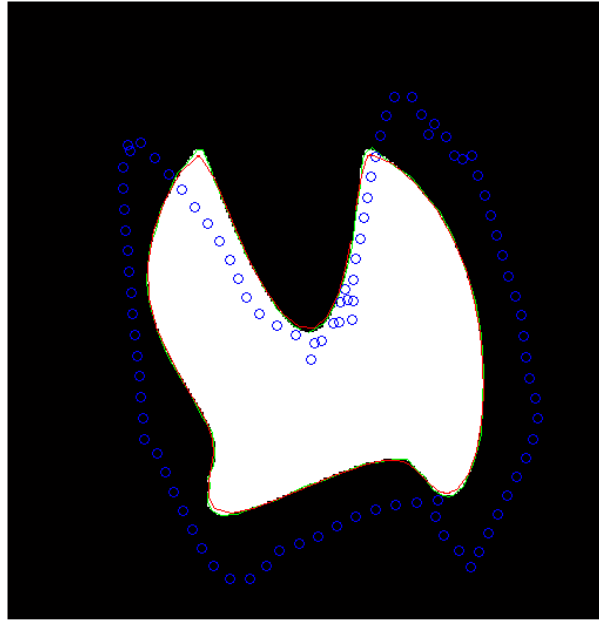
**Şekil 4.14.** Testimge\_2 GVF algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



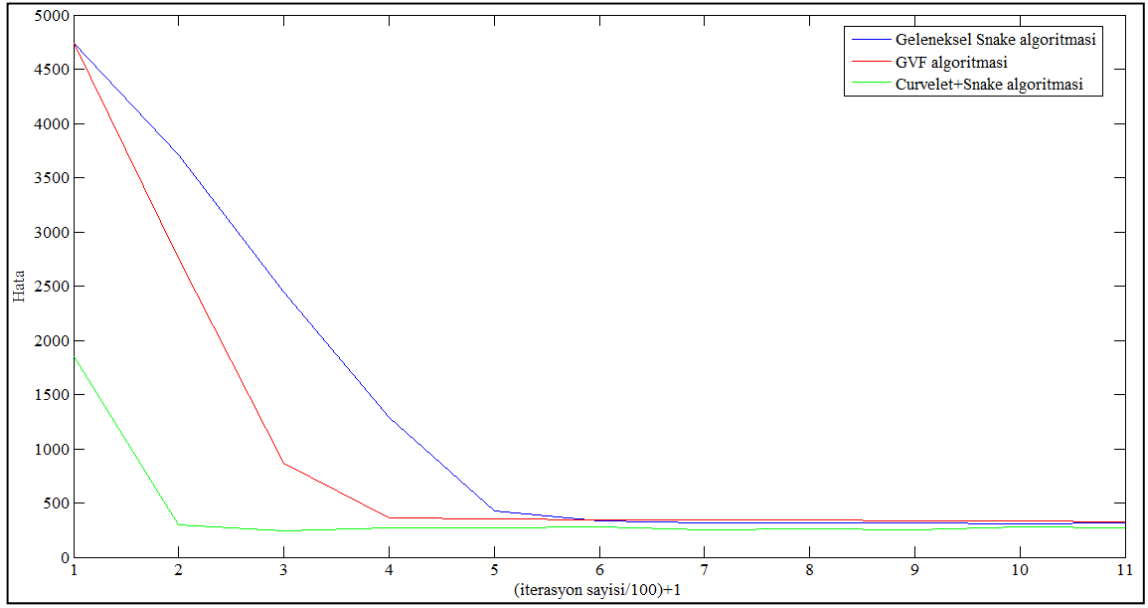
**Şekil 4.15.** Testimge\_2 Curvelet katsayıları



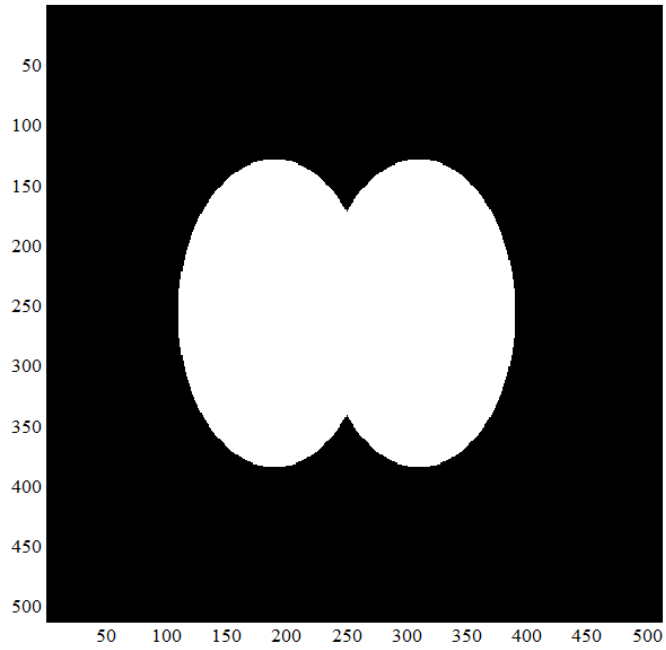
**Şekil 4.16.** Testimge\_2 Curvelet katsayıları uygulandığında Snake hareketi



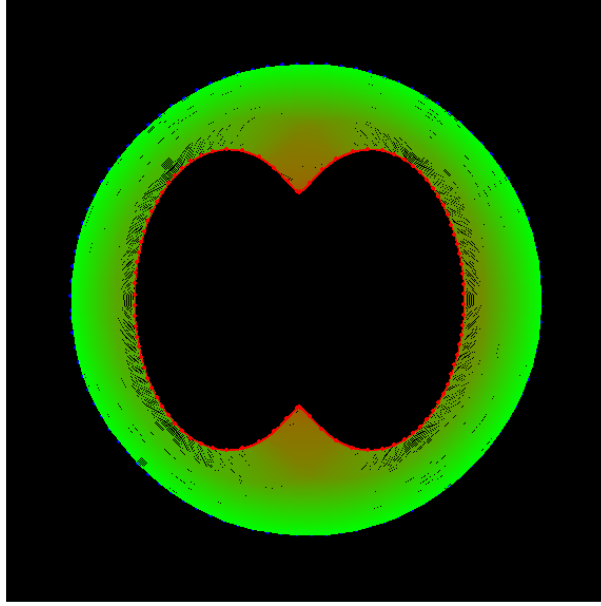
**Şekil 4.17.** Testimge\_2 Curvelet katsayıları ile Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



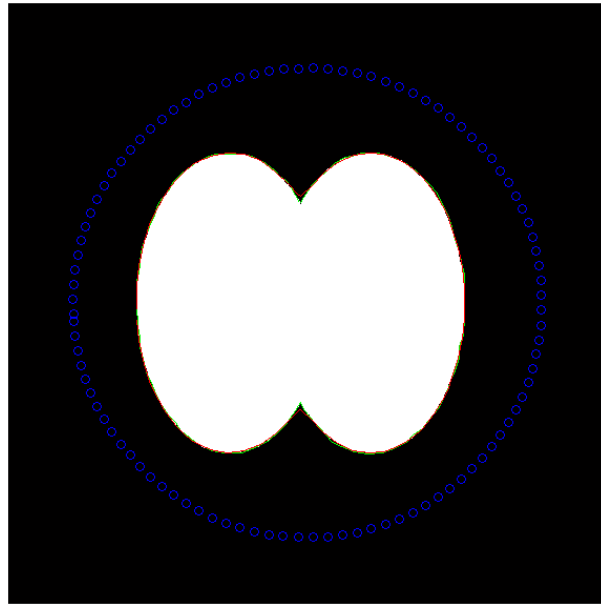
Şekil 4.18. Testimge\_2 iterasyon adımları için kenar algılama hata grafiği



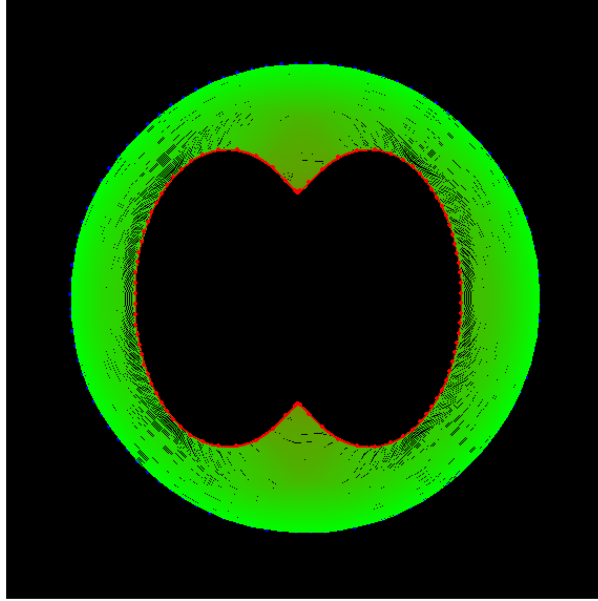
Şekil 4.19. Testimge\_3 orijinal görüntü (512×512)



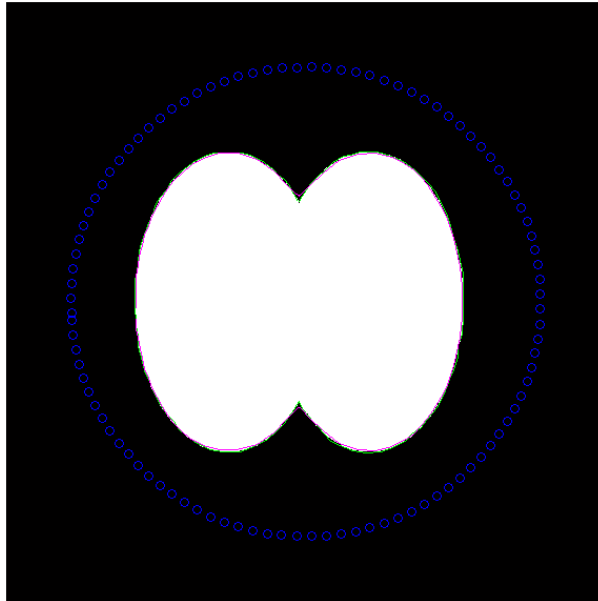
**Şekil 4.20.** Testimge\_3 geleneksel Snake algoritmasında Snake hareketi



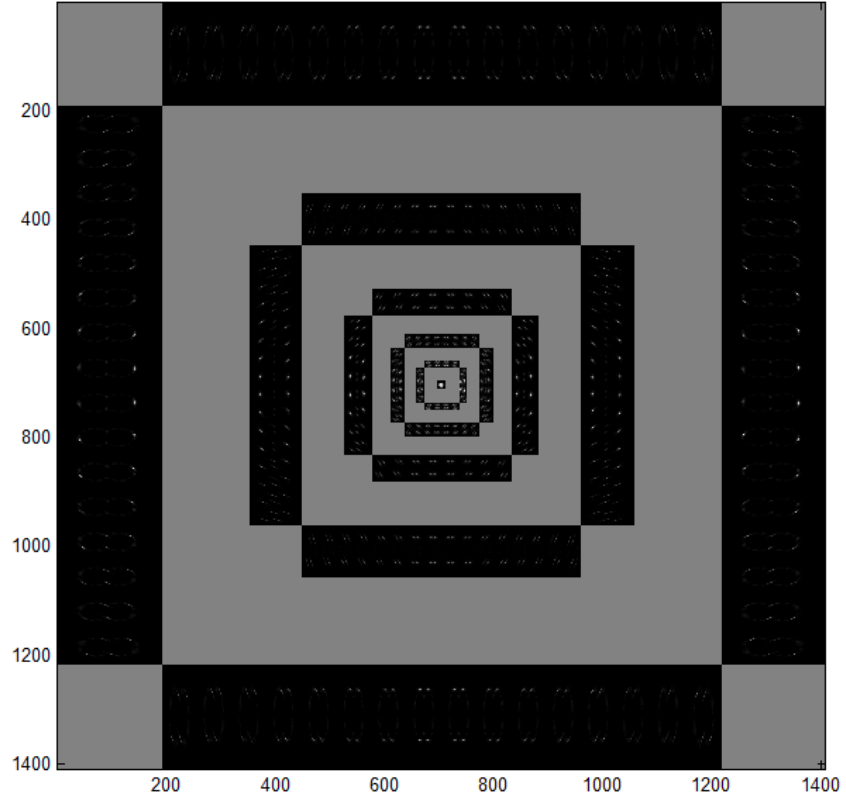
**Şekil 4.21.** Testimge\_3 geleneksel Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



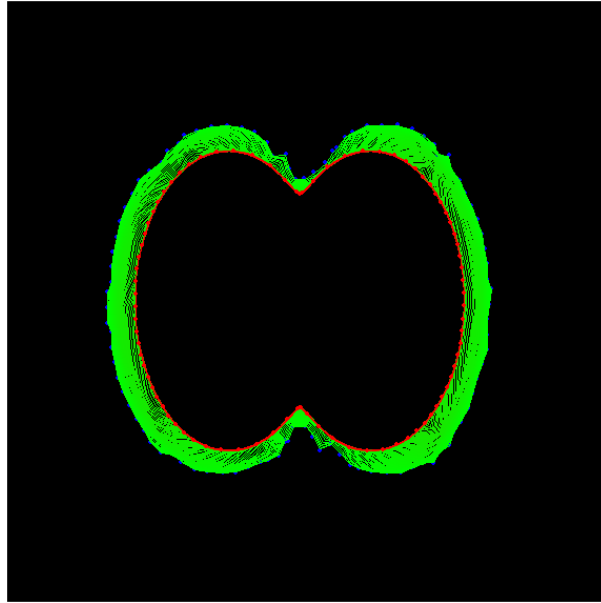
**Şekil 4.22.** Testimge\_3 GVF algoritmasında Snake hareketi



**Şekil 4.23.** Testimge\_3 GVF algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)

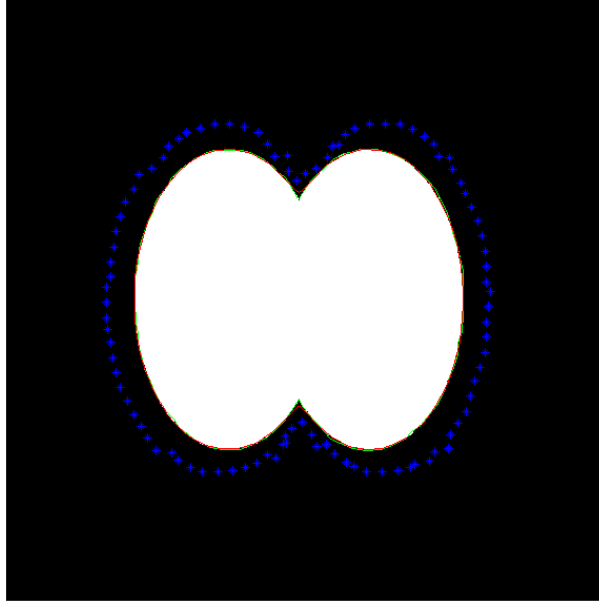


Şekil 4.24. Testimge\_3 Curvet katsayıları

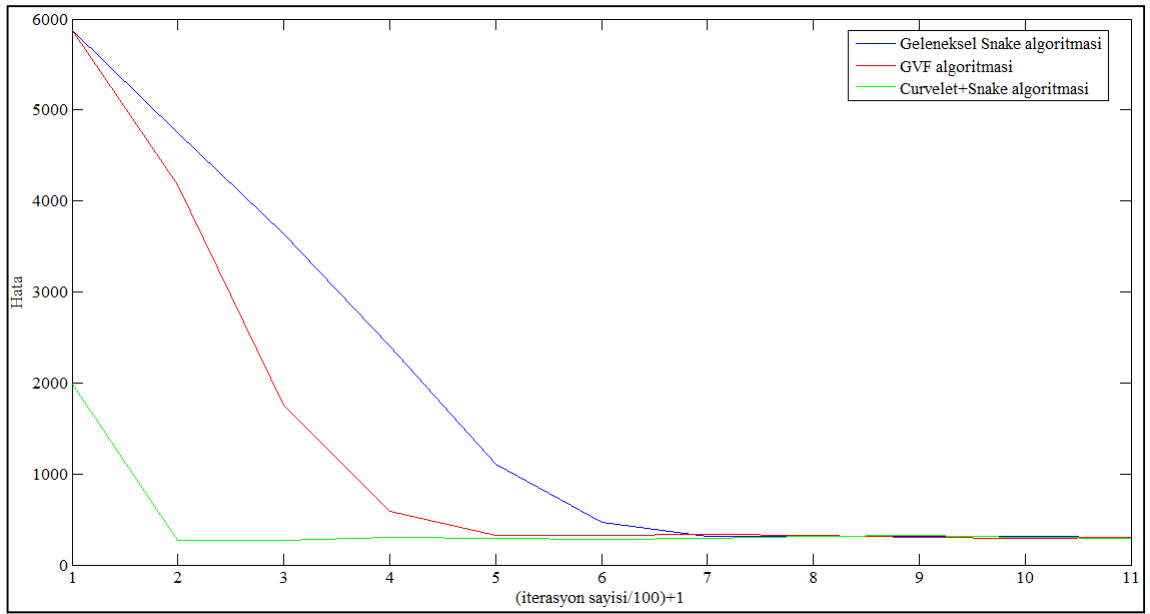


Şekil 4.25. Testimge\_3 Curvet katsayıları uygulandığında Snake hareketi

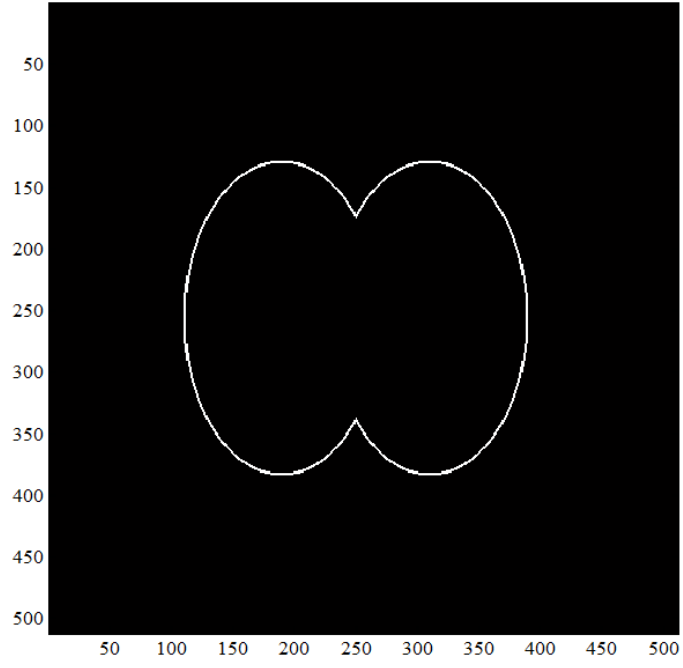




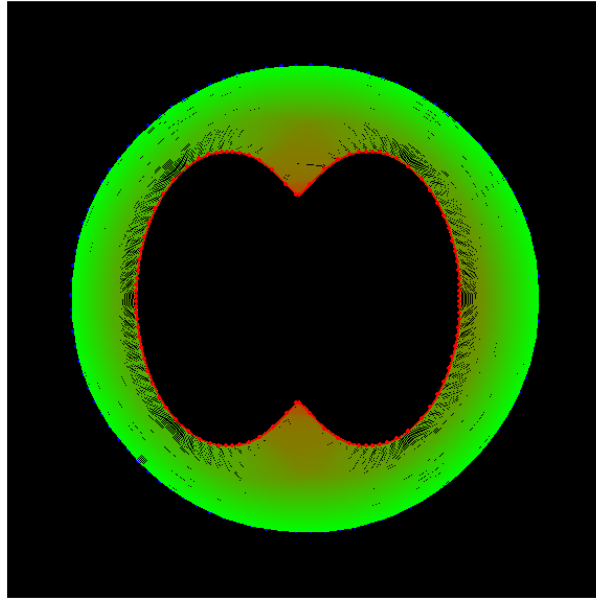
**Şekil 4.26.** Testimge\_3 Curvelet katsayıları ile Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



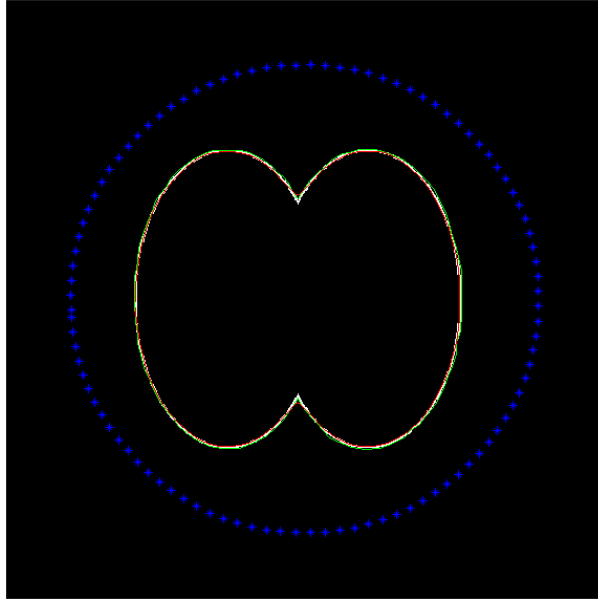
**Şekil 4.27.** Testimge\_3 iterasyon adımları için kenar algılama hata grafiği



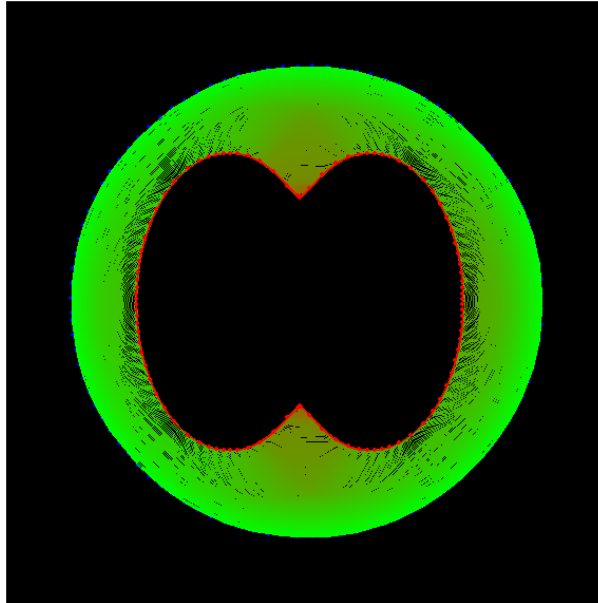
Şekil 4.28. Testimge\_4 orijinal görüntü (512×512)



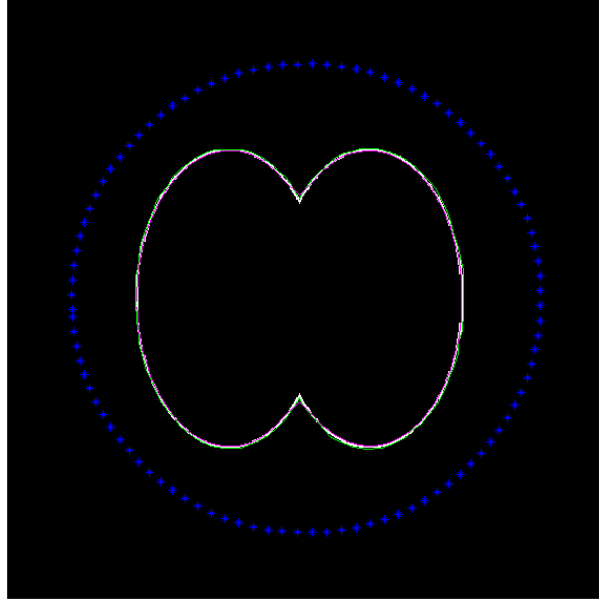
Şekil 4.29. Testimge\_4 geleneksel Snake algoritmasında Snake hareketi



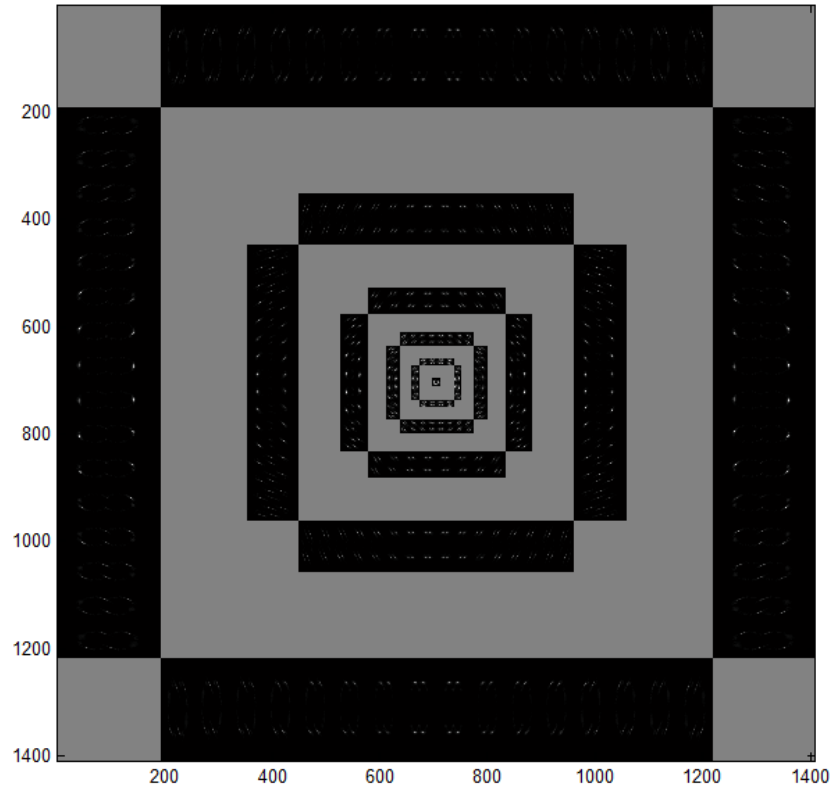
**Şekil 4.30.** Testimge\_4 geleneksel Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



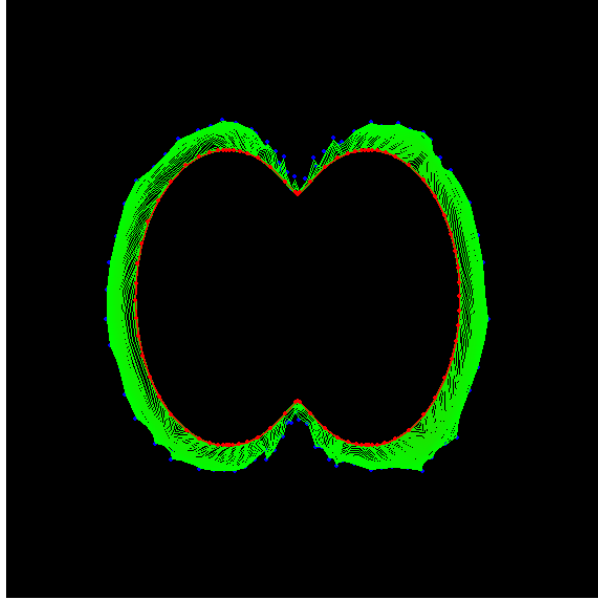
**Şekil 4.31.** Testimge\_4 GVF algoritmasında Snake hareketi



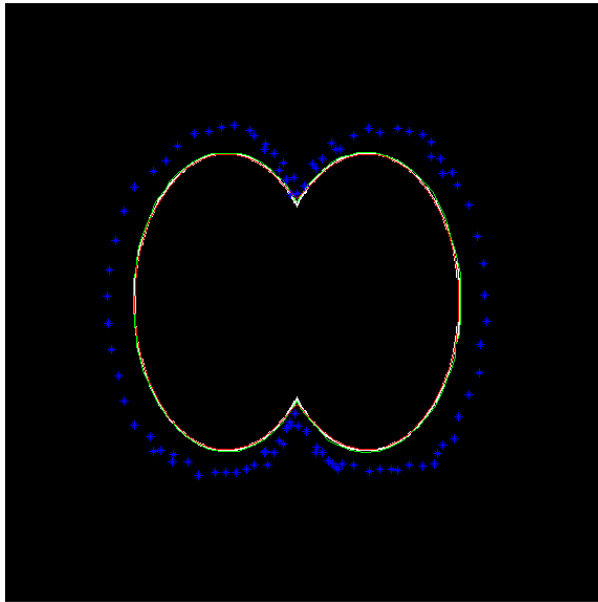
**Şekil 4.32.** Testimge\_4 GVF algoritması gösterimi  
 \*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



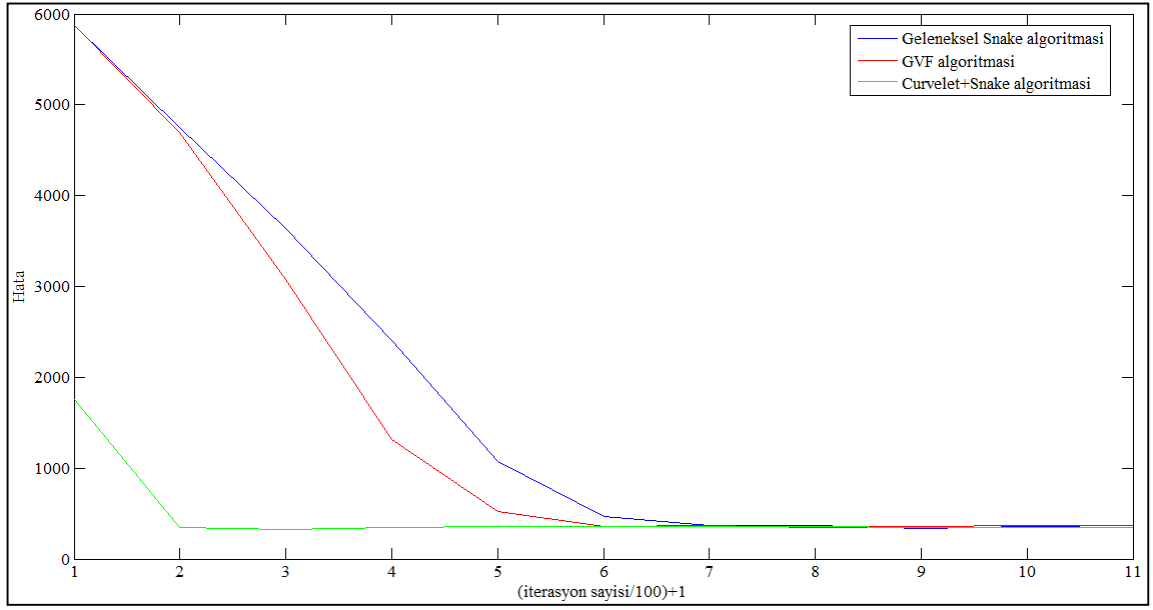
**Şekil 4.33.** Testimge\_4 Curvelet katsayıları



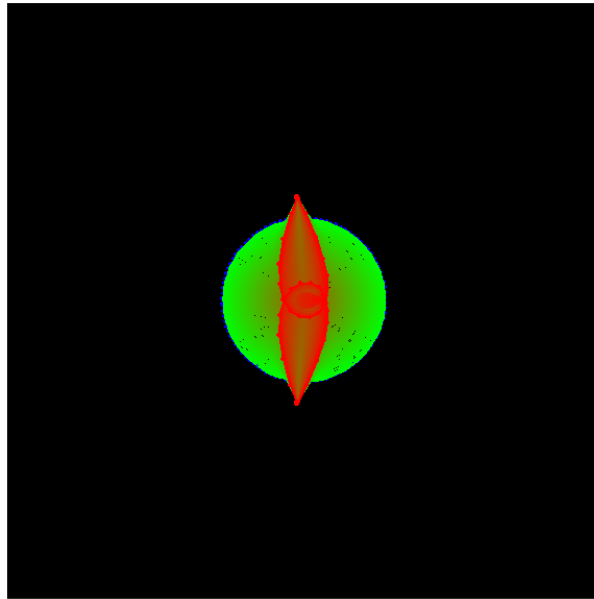
**Şekil 4.34.** Testimge\_4 Curvelet katsayıları uygulandığında Snake hareketi



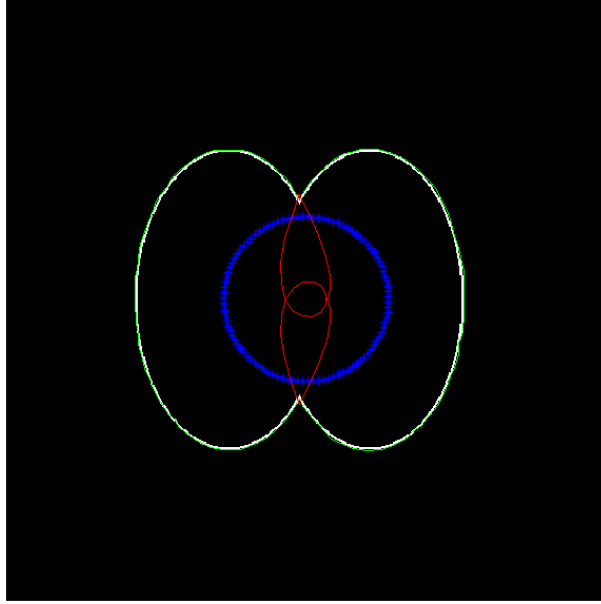
**Şekil 4.35.** Testimge\_4 Curvelet katsayıları ile Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



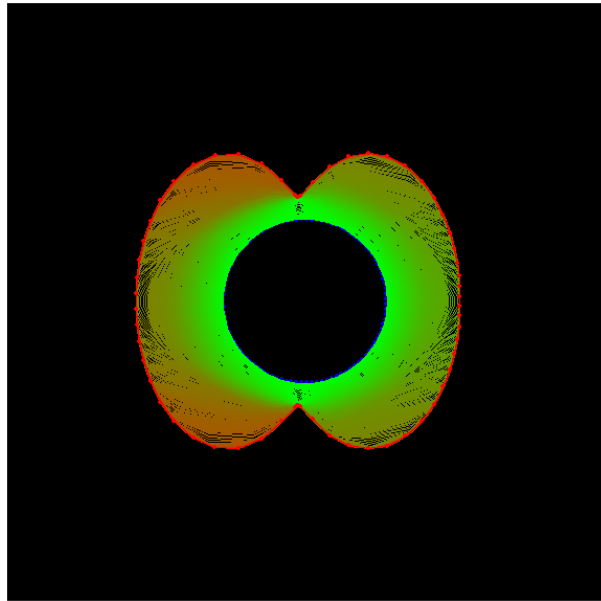
Şekil 4.36. Testimge\_4 iterasyon adımları için kenar algılama hata grafiği



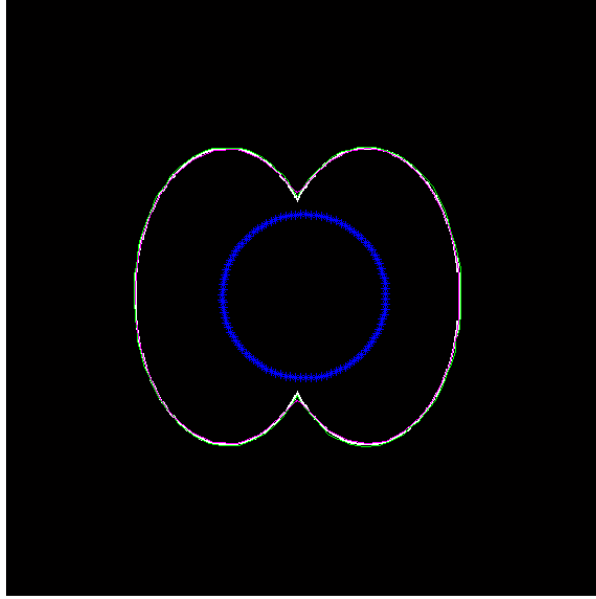
Şekil 4.37. Testimge\_4 geleneksel Snake algoritmasında Snake hareketi



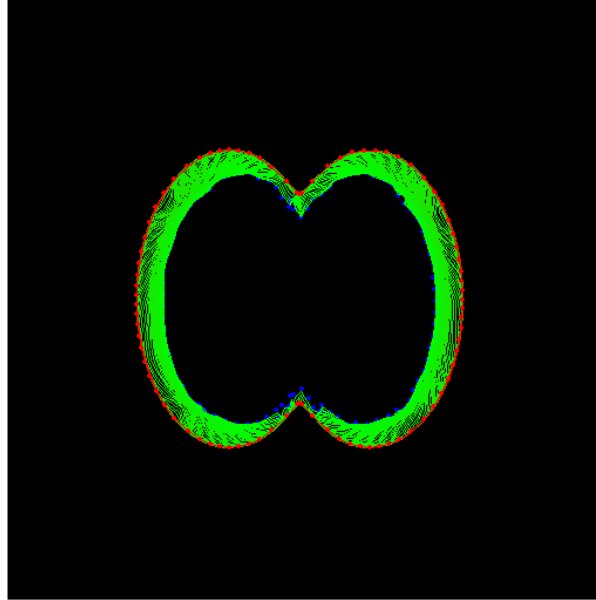
**Şekil 4.38.** Testimge\_4 geleneksel Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



**Şekil 4.39.** Testimge\_4 GVF algoritmasında Snake hareketi

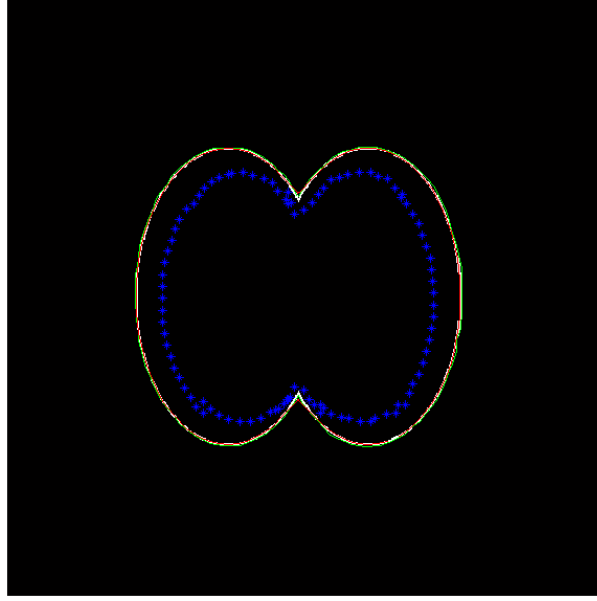


**Şekil 4.40.** Testimge\_4 GVF algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)

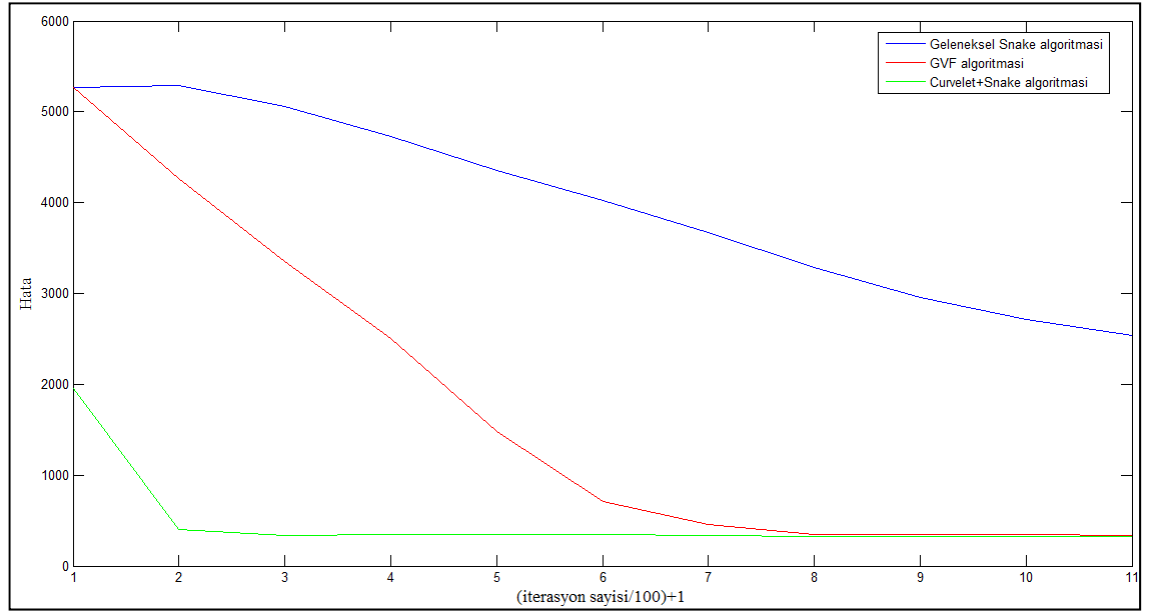


**Şekil 4.41.** Testimge\_4 Curvelet katsayıları uygulandığında Snake hareketi





**Şekil 4.42.** Testimge\_4 Curvelet katsayıları ile Snake algoritması gösterimi  
\*Referans noktalar (yeşil), başlangıç konturu (mavi), sonuç konturu (kırmızı)



**Şekil 4.43.** Testimge\_4 iterasyon adımları için kenar algılama hata grafiği

## 5. SONUÇ

Bu çalışmada, kenar algılamada etkili otomatik bir yöntem geliştirilmeye çalışıldı. Kenar algılama; çoklu çözünürlük analizi ve bir bölütleme tekniği kullanılarak iki aşamada gerçekleştirildi.

Bu amaçla, çoklu çözünürlük analizlerinden ikinci nesil Curvelet dönüşümü kullanıldı. Curvelet dönüşümü görüntüyü farklı açılarda inceleme fırsatı sağladığı için görüntü hakkında daha fazla bilgi sahibi olundu ve böylelikle görüntünün sınırları daha net bir şekilde belirlendi. Curvelet dönüşümü sonucu elde edilen katsayılarından alınan kenar bilgileri görüntü bölütleme tekniklerinden Snake algoritmasında ön bilgi olarak kullanarak bu algoritmanın performansı artırıldı ve kenarlar küçük hata oranlarıyla başarıyla belirlendi.

Geliştirdiğimiz bu yöntem ile ilk olarak, görüntü bölütlemeye yaygın olarak kullanılan herhangi bir Snake algoritması için başlangıç noktalarının kullanıcı desteği olmaksızın, otomatik olarak, belirlenebileceği bir yaklaşım elde edilmiş oldu. İkinci olarak, bu bilginin kullanımıyla geleneksel Snake algoritmasının performansının normal şartlar altında çok daha iyi olan Gradyent Vektör Akışı (GVF) algoritması performansının üstüne çıkarılabileceği sonucu elde edilmiş oldu.

Buna göre; GVF algoritması daha çok iterasyonla daha az başarıyla görüntü kenarlarını belirlerken, geleneksel Snake algoritması üzerine otomatik olarak verilen ön bilgiyle kenarların belirlenmesinde daha hızlı ve başarılı olduğu sonucuna varıldı.

GVF algoritmasının daha çok iterasyonla daha az başarıyla görüntü kenarlarını belirlediği, geleneksel Snake algoritmasının yetersizliği üzerine otomatik olarak verilen ön bilgiyle geliştirilen yöntemin ise kenarların belirlenmesinde daha hızlı ve başarılı olduğu sonucuna varıldı.

**KAYNAKLAR**

- Aygün, O., 2006. Konuşmacı Tanıma Sistemlerinde Dalgacık Dönüşümü. Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Candès, E., Demanet, L., Donoho, D., Ying, L., 2006. Fast Discrete Curvelet Transforms. *SIAM Multiscale Model. Simul.*, 5/3, 861-899.
- Cvetkovic, D., Ubeyli, E. D., Cosic, I., 2008. Wavelet Transform Feature Extraction From Human PPG, ECG, and EEG Signal Responses To ELF PEMF Exposures: A Pilot Study. *Digital Signal Processing*, Vol. 18, 861-874.
- Çamaşırcıoğlu, E., 2007. Araç Plakası Algılama ve Tanıma. Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Daubechies, I., 1990. The Wavelet Transform, Time-frequency Localization and Signal Analysis. *IEEE Trans. on Information Theory*, Vol.36, No.5, 961-1005.
- Demir, Ö., 2008. EEG Dalgalarının Wavelet (Dalgacık) Dönüşümü İle Değerlendirilmesi. Yüksek Lisans Tezi, Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü, Kütahya.
- Dettori, L., Semler, L., 2007. A comparison of wavelet, ridgelet, and curvelet-based texture classification algorithms in computed tomography. *Computers in Biology and Medicine* 37 486 – 498.
- Do, M. N. , Vetterli M., 2001. Contourlets. *Beyond Wavelets*, J. Stoeckler and G. V. Welland, Academic Press, 1–27.
- Do, M. N. , Vetterli M., 2003. The Finite Ridgelet Transform for Image Representation. *IEEE Transactions on Image Processing*, vol. 12, num. 1, 16-28
- Donoho, D.L., Duncan, M.R., 1999. Digital Curvelet Transform: Strategy, Implementation and Experiments, <http://www-stat.stanford.edu/~donoho/Reports/1999/DCvT.pdf> (25.05.2012).
- Fadili, M.J., Starck, J.-L., 2007. Curvelets and Ridgelets, <http://jstarck.free.fr/curvencyclop09.pdf> (21.05.2012).
- Günel, S., 2008. Örüntü Tanıma Uygulamalarında Altuzay Analiziyle Öznitelik Seçimi ve Sınıflandırma. Doktora Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir.
- Jung Jun, L., Sang Min, L., In Young, K., Hong Ki, M., Seung Hong, H., 1999. Comparison Between Short Time Fourier and Wavelet Transform For Feature Extraction Of Heart Sound, In *Tencon 99. Proceedings Of The Ieee Region 10 Conference*, Vol.2, 1547-1550.
- Labate, D., Lim, W-Q., Kutyniok, G., Weiss, G., 2005. Sparse multidimensional representation using shearlets. In *Wavelets XI*, volume 5914, 254–262. SPIE.
- Le Pennec, E., Mallat, S., 2000. Image processing with geometrical wavelets. In *International Conference on Image Processing*.
- Ma, J., Plonka, G., 2010. The Curvelet Transform. *Signal Processing Magazine, IEEE* Volume 2, pages 118-133.
- Mallat, S., 1999. *A Wavelet Tour Of Signal Processing*, Academic Press, 637.
- Mallat, S., 2007. Geometrical grouplets. *Applied and Computational Harmonic Analysis*.

- Misiti, M., Misiti, Y., Oppenheim, G., And Poggi, J. M., 1996-1997. Wavelet Toolbox Users Guide, The Mathworks, Inc.
- Polat, R., 2007. Biyomedikal Görüntü Bölütleme. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Polikar, R., 1999. The Story Of Wavelets IMACS /IEEE CSCC'99 Proceedings, Pp. 5481-5486.
- AlZu'bi, S., Islam, N., Abbod, M., 2011. Multiresolution Analysis Using Wavelet, Ridgelet, and Curvelet Transforms for Medical Image Segmentation. Hindawi Publishing Corporation International Journal of Biomedical Imaging Volume 2011, Article ID 136034, 18 pages doi:10.1155/2011/136034.
- Selesnick, I. W., 2007. Wavelet Transforms — A Quick Study.
- Tufekci, Z., Gowdy, J. N. ,2000. Feature Extraction Using Discrete Wavelet Transform For Speech Recognition, Proceedings Of The IEEE Southeastcon, 116–123.
- Velisavljevic, V., Beferull-Lozano, B., Vetterli, M., Dragotti, P.L., 2006. Directionlets: Anisotropic multidirectional representation with separable filtering. Image Processing, IEEE Transactions on 15(7): 1916-1933.
- Yakar, E., 2006. Sar Görüntülerinde Benek Gürültüsünün Giderilmesi İçin Çoklu-Dalgacık Dönüşümüne Dayalı Yeni Bir Yaklaşım. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

## ÖZGEÇMİŞ

F.Saba AHISHALI 1987 yılında Erzurum’da doğdu. İlk, orta ve lise öğrenimini Erzurum’da tamamladı. 2005-2009 yılları arasında Atatürk Üniversitesi Elektrik&Elektronik Mühendisliği ana bilim dalında lisans derecesini aldı. Mezuniyetinin ardından 2009 yılında Yüksek lisans eğitimine başladı. 2011 yılında Atatürk Üniversitesi Elektrik&Elektronik Mühendisliği Elektronik Bilim dalında araştırma görevlisi olarak göreve başladı ve halen bu görevi sürdürmektedir.