

154588

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEMEDE İNCELTME ALGORİTMALARI

Matematik Müh. Armağan ELİBOL

FBE Matematik Mühendisliği Anabilim Dalında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. İbrahim EMİROĞLU

Prof. Dr. Mehmet AHUATÇIOĞLU

Doç. Dr. Ömer GÖK

İSTANBUL, 2004

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTIMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ.....	viii
ÖNSÖZ	ix
ÖZET.....	x
ABSTRACT.....	xi
1. GİRİŞ	12
2. DİJİTAL GÖRÜNTÜ TANIMI	14
2.1 Dijital Görüntü İşleme ve Dijital Görüntü Tanımı.....	14
2.2 Bitmap (BMP) Resim Dosyasının Yapısı	15
2.2.1 Bitmap Dosya Başlığı (Bitmap File Header)	15
2.2.2 Bitmap Bilgi Başlığı(Bitmap Info Header)	16
3. DİJİTAL GÖRÜNTÜ İŞLEMENİN TEMEL ALGORİTMALARI	17
3.1 Dijital Görüntünün Histogramı	17
3.2 Thresholding.....	18
4. FOURIER DÖNÜŞÜMÜ ve YÖNLÜ FOURIER FİLTRESİ (YFF).....	20
4.1 Fourier Dönüşümü.....	20
4.2 Ayırık Fourier Dönüşümü	21
4.3 İki Boyutlu Fourier Dönüşümünün Özellikleri	24
4.3.1 Bölünebilme	24
4.3.2 Dağılım ve Scaling	25
4.3.3 Ortalama Değer	25
4.4 Hızlı Fourier Dönüşümü (The Fast Fourier Transform)	26
4.4.1 FFT Algoritması	26
4.4.2 İşlemlerin sayısı.....	29
4.4.3 Ters FFT	31
4.5 Yönlü Fourier Filtresi (Directional Fourier Transform)	32
4.5.1 Yerel Sağrı Yöneliminin Hesaplanması	32
4.5.2 Önfilterli Resimlerin Oluşturulması.....	34
4.5.3 Önfilterli Resimlerden Uygun Parçaların Seçilmesi	37
5. İNCELTME (THINNING) ALGORİTMALARI	39

5.1	Zhang ve Suen Algoritması (ZS)	40
5.2	Lu ve Wang Algoritması (LW)	43
5.3	Rutovitz Algoritması (RD).....	44
5.4	Zhang ve Wang Algoritması (ZW)	45
6.	TEK PİKSELE İNDİRGEME ALGORİTMALARI	47
6.1	Tek Piksel Genişliğine Düşürmek için İnceltme Sonrası Önerilen Algoritmalar	47
7.	SONUÇLAR	49
KAYNAKLAR		75
ÖZGEÇMİŞ		77



SİMGE LİSTESİ

$f(x, y)$	x ve y uzaysal koordinatlar olmak üzere ışık yoğunluk fonksiyonu
$\mathcal{F}\{f(x, y)\}$	$f(x, y)$ 'nin fourier dönüşümü
$F^*(u, v)$	$F(u, v)$ 'nin kompleks konjugate i
$C(i)$	Resmin histogram değerlerinin birikimli olarak ifadesi
r_k	k . gri seviye
n_k	k . gri seviyeye ait resimdeki toplam piksel sayısı
$p(r_k)$	Histogram fonksiyonu
$F(i, j)$	Orijinal resim
$G(i, j)$	Threshold edilmiş siyah-beyaz resim
$S(p_1)$	p_1 in 3×3 lük komşuluğundaki toplam siyah piksel sayısını
$T(p_1)$	p_1 in 3×3 lük komşuluğundaki siyah pikselden beyaz piksele geçiş sayısı
$K(i, j)$	i, j noktasındaki pikselin yönelimi(yöntü)



KISALTMA LİSTESİ

BMP	Bitmap Resim Dosyası
ZS	Zhang ve Suen Paralel İnceltme Algoritması
LW	Lu ve Wang Paralel İnceltme Algoritması
RD	Rutovitz Paralel İnceltme Algoritması
ZW	Zhang ve Wang Paralel İnceltme Algoritması
YFF	Yönlü Fourier Filtresi
RAT	Relational Average Thresholding



ŞEKİL LİSTESİ

Şekil 2.1 Resim ve dijital gösterimi (Castleman, 1996).....	14
Şekil 2.2 Tezde kullanılan parmakizi resminin 256*256 boyutlarındaki görünümü.	15
Şekil 3.1 Test resmi ve histogram eşitlenmiş halinin 256*256 boyutlarındaki görüntüleri.....	18
Şekil 3.2 Solda 8x8 'lik, sağda 16x16 'lık alt bölgeler ile yöntemin test resmi üzerinde uygulanmış görüntüleri.....	19
Şekil 4.1 $f(x)$ fonksiyonu ve Fourier dönüşümü.....	21
Şekil 4.2 Sürekli fonksiyonun örneklenmesi	22
Şekil 4.3 1-D dönüşümler serisi olarak 2-D Fourier dönüşümünün hesaplanması.....	25
Şekil 4.4 Yönlü Fourier Filtresi	32
Şekil 4.5 8 yönlü resmin hesaplanması (Emiroğlu, 1997)	33
Şekil 4.6 Şekil 2.2'deki resmin piksel ve blok tabanlı yönlü görüntüleri.....	34
Şekil 4.7 Şekil 2.2'deki resmin yerel sağrı yönelimleri.....	34
Şekil 4.8 8 yönde oluşturulan önfiltreli resimlerin 256*256 lık görüntüleri.....	36
Şekil 4.9 Şekil 2.2'deki resmin YFF ile iyileştirilmiş görüntüsü.	37
Şekil 4.10 Şekil 4.9'un RAT yöntemi ile siyah-beyaz görüntüsü.	38
Şekil 5.1 Örnek durumlar	39
Şekil 5.2 İncelenen piksel p_1 ve 3x3 komşuluğu.....	40
Şekil 5.3 (a) $S(p_1) = 2, T(p_1) = 2$ (b) $S(p_1) = 5, T(p_1) = 1$	40
Şekil 5.4 Birinci alt iterasyon 1.Doğu sınır, 2.Güney sınır, 3.Kuzey-Batı Köşe	41
Şekil 5.5 İkinci alt iterasyon 1.Kuzey Sınır,2.Batı sınır,3.Güney-Doğu sınır.....	42
Şekil 5.6 Test resminin ZS Algoritması ile inceltmiş hallerinin 256*256 'lık görüntüleri....	43
Şekil 5.7 Test resminin LW Algoritması ile inceltmiş hallerinin 256*256 'lık görüntüleri.44	44
Şekil 5.8 RD Algoritması ile inceltmiş test resminin 256*256 lık görüntüleri.	45
Şekil 5.9 ZW algoritması için piksellerin etiketlenmesi	45
Şekil 5.10 ZW Algoritması ile inceltmiş test resminin 256*256 'lık görüntüleri	46
Şekil 6.1 Yeni algoritmalar için piksel etiketleme şeması	47
Şekil 7.1 ZS Algoritması ile inceltmiş orijinal parmakizi resmi	51
Şekil 7.2 Şekil 7.1 'deki inceltilmiş resmin I. Algoritma'dan geçirilmiş hali.	52
Şekil 7.3 Şekil 7.1 'deki resmin II. Algoritma'dan geçirilmiş hali.	53
Şekil 7.4 İncelenen on farklı resim için ZS Algoritması ile inceltilen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları	53
Şekil 7.5 Yönlü Fourier Filtresi ile iyileştirilmiş test resminin ZS Alg. ile inceltilmiş hali	54
Şekil 7.6 Şekil 7.5 'in I. Algoritma'dan geçirilmiş hali	55
Şekil 7.7 Şekil 7.5 'in II. Algoritma'dan geçirilmiş hali.....	56
Şekil 7.8 İncelenen on farklı resmin iyileştirilmiş versiyonu için ZS Algoritması ile inceltilen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları.....	56
Şekil 7.9 Test resminin LW Algoritması ile inceltilmiş hali	57
Şekil 7.10 Şekil 7.9 'nin I. Algoritma'dan geçirilmiş hali.	58
Şekil 7.11 Şekil 7.9 'zun II. Algoritma'dan geçirilmiş hali.	59
Şekil 7.12 İncelenen on farklı resim için LW Algoritması ile inceltilen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları	59
Şekil 7.13 YFF'den geçirilmiş test resminin LW Algoritması ile inceltilmiş hali.	60
Şekil 7.14 Şekil 7.13 'ün I.Algoritma'dan geçirilmiş hali	61
Şekil 7.15 Şekil 7.13 'ün II. Algoritma'dan geçirilmiş hali.....	62
Şekil 7.16 İncelenen on farklı resmin iyileştirilmiş versiyonu için LW Algoritması ile inceltilen resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları	62

Şekil 7.17 Test resminin RD Algoritması ile inceltilmiş hali.	63
Şekil 7.18 Şekil 7.17 'nin I.Algoritma'dan geçirilmiş hali	64
Şekil 7.19 Şekil 7.17 'nin II.Algoritma ile tek piksele indirgenmiş hali.	65
Şekil 7.20 İncelenen on farklı resim için RD Algoritması ile inceltiren resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları	65
Şekil 7.21 YFF ile iyileştirilmiş test resminin RD Algoritması ile inceltilmiş hali.	66
Şekil 7.22 Şekil 7.21 'rin I. Algoritma ile tek piksele indirgenmiş hali.	67
Şekil 7.23 Şekil 7.21 'rin II. Algoritma ile tek piksele indirgenmiş hali	68
Şekil 7.24 İncelenen on farklı resmin iyileştirilmiş versiyonları için RD Algoritması ile inceltiren resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları	68
Şekil 7.25 Test resminin ZW Algoritması ile inceltilmiş hali.	69
Şekil 7.26 Şekil 7.25 'in I. Algoritma ile tek piksele indirgenmiş hali.	70
Şekil 7.27 Şekil 7.25 'in II. Algoritma ile tek piksele indirgenmiş hali.	71
Şekil 7.28 İncelenen on farklı resim için ZW Algoritması ile inceltiren resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları	71
Şekil 7.29 YFF ile iyileştirilmiş test resminin ZW Algoritması ile inceltilmiş hali.	72
Şekil 7.30 Şekil 7.29 'zun I. Algoritma ile tek piksele indirgenmiş hali.	73
Şekil 7.31 Şekil 7.29 'zun II. Algoritma ile tek piksele indirgenmiş hali.	74
Şekil 7.32 İncelenen on farklı resmin iyileştirilmiş versiyonları için ZW Algoritması ile inceltiren resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları	74

ÇİZELGE LİSTESİ

Çizelge 2.1 Dijital resimlerin sınıflandırılması	14
Çizelge 2.2 Bitmap dosya başlığı.....	15
Çizelge 2.3 Bitmap bilgi başlığı.....	16
Çizelge 4.1 1 N değişkenleri için $N \log_2^N$ 'e karşı N^2 nin karşılaştırılması	27
Çizelge 5.1 1. Alt iterasyon şartları.....	41
Çizelge 5.2 2. Alt iterasyon şartları.....	41
Çizelge 5.3 LW Algoritması'nın Şartları	43
Çizelge 5.4 RD Algoritması'nın şartları	44
Çizelge 5.5 ZW Algoritması'nın şartları.....	45
Çizelge 6.1 I. Algoritma için piksel silme şartları	47
Çizelge 6.2 II. Algoritma için piksel silme şartları	48
Çizelge 6.3 Resmin taranma yönlerine göre piksel silme şartları.....	48
Çizelge 7.1 İnceltilmiş resimlerdeki siyah piksel sayısı	49
Çizelge 7.2 I. Algoritmanın uygulanması ile resimden silinen piksel sayıları.....	49
Çizelge 7.3 II. Algoritmanın uygulanması ile resimden silinen piksel sayıları	49



ÖNSÖZ

Benimle çalışmayı kabul edip, bu tezin oluşmasında bana önder ve yardımcı olan tez danışmanım Yrd. Doç. Dr. İbrahim Emirođlu'na, beni daima destekleyen aileme ve Türkiye Bilimsel ve Teknik Araştırma Kurumu'na teşekkürlerimi sunarım...

Haziran 2004

Armađan Elibol



ÖZET

Dijital görüntü işleme tıbbi teşhis koyma, adli olaylarda suçlunun tesbit edilmesi v.b. gibi birçok değişik olaylarda yoğun olarak kullanılmaktadır. Hızlı gelişen teknoloji ile birlikte görüntü işleme de gelişmekte ve genişlemektedir. Dijital görüntü işleme birçok kısımdan oluşmaktadır. Örneğin inceltme, iyileştirme, sıkıştırma ve tanımlama gibi.

Bu tezde, inceltme algoritmaları üzerinde yoğunlaştık. İnceltme işleminin dijital görüntü işlemedeki hızlı gelişmelerle birlikte önemi her geçen gün artmaktadır. İnceltme daha çok bir ara işlem olarak geniş bir kullanım alanı içerisinde yer almaktadır. İnceltme işlemi parmakizi analizleri, el yazısı tanımlama, imza doğrulama, şekil tanımlama, örüntü tanımlama, biomedikal sistemlerde ve tıbbi görüntüleme işlemlerinde yoğun olarak kullanılmaktadır. İnceltme işlemi siyah pikselleri silen ve resmi tek piksel genişliğine düşüren bir işlemdir ve buna resmin "iskeleti" denir. İnceltme algoritmalarının, resmin iskeletini bir piksel genişliğine indirmesi beklenmektedir.

Bu çalışmada en çok kullanılan ve bilinen dört değişik paralel inceltme algoritması incelenmiş ve bu algoritmaların parmakizi resimlerine uygulanması yapılmıştır. Algoritmaların parmakizi resimlerini tek-piksel genişliğinde inceltmediği yapılan çalışmalarla ortaya çıkarılmıştır. Bunu yapmak için, inceltme sonrası uygulanan iki yeni algoritma gerçekleştirilmiştir. Ayrıca tezin içerisinde parmakizi resimlerinin iyileştirilmesinde kullanılan Fourier Dönüşümü ve Yönlü Fourier Filtresi detaylı olarak açıklanmıştır. Bu yöntem ile iyileştirilen resimler ile orijinal resimlerin inceltmeleri gerçekleştirilmiştir ve inceltmelerin sonuçları karşılaştırılmıştır.

Elde edilen sonuçlarından da görülebileceği üzere önerilen algoritmalar, kolay ve hızlı bir şekilde uygulanabilir ve parmakizi resimlerinin tek-piksel genişliğindeki iskeletlerinin oluşturulmasında kullanılabilir. Fakat bu çok iyi bilinen ve varolan algoritmalarla yapılamamaktadır.

Anahtar Kelimeler: Paralel İnceltme Algoritmaları, Tek Piksel Genişliği, Resmin İskeleti, Yönlü Fourier Filtresi, Parmakizi Resimleri

ABSTRACT

Digital image processing is used often in many areas such as diagnosis in medicine and identifies the crimes. As the technology develops faster, the digital image processing also develops and widely changes. The digital image processing can be seen in several parts such as thinning, enhancing, compression and recognition.

In this thesis, we mainly focus on the thinning algorithms. Because of the new changes on the digital image processing, the importance of thinning algorithms increases every day. Thinning is one of the most important pre-processing steps in the digital image processing. It is used in fingerprint identification, pattern and character recognition and biomedical systems. Thinning is a process that deletes the dark points and transforms the pattern into a thin line drawing known as a skeleton. Algorithms on the thinning are supposed to transform input images to one-pixel wide skeleton.

In this work, four different well-known and most used parallel thinning algorithms are surveyed and applied to fingerprint images. During this work, it is found out that these algorithms fail to produce one-pixel wide skeletons of fingerprint images. In order to this, two new post-processing algorithms are introduced in this thesis. These two algorithms are applied after the thinning process. In addition to the algorithms, in the thesis, Fourier Transform and Directional Fourier Filter are explained in detail, which are the enhancement techniques of fingerprint images. Enhanced and original fingerprints are thinned by the use of mentioned algorithms and computational results are compared.

It can be seen from the computational results that the proposed algorithms are easy and fast to apply and they can be used to generate one-pixel wide skeleton from the thinned fingerprint images which is not possible from the existing well-known algorithms.

Keywords: Parallel Thinning Algorithms, One-Pixel Wide, Skeletons, Directional Fourier Filter, Fingerprint Images

1. GİRİŞ

Dijital görüntü işleme günümüzde teknolojinin çok hızlı gelişmesine paralel olarak çok hızlı gelişmektedir. Örneğin parmakizi ile açılan kapılar ve müşterilerini irislerinden tanıyan bankamatik gişeleri gibi uygulamalar bulunmaktadır. Bunların hepsi temelinde görüntü işleme algoritmalarına dayanmaktadır.

Bu çalışmada dijital görüntü işlemede daha çok bir ara işlem olarak kullanılan inceltme algoritmaları incelenmiştir. Algoritmaların parmakizi resimleri üzerine uygulanması ve literatürde var olan sonuçlardan daha iyi sonuçlar elde edilmesi amaçlanmıştır.

Literatürde inceltmenin tanımı üzerinde kesin bir anlaşma yoktur. Farklı tanımlar yapılmıştır, algoritmalar kurulmuş ve değişik bazen de beklenmedik sonuçlar ortaya çıkmıştır. (Zhang ve Wang, 1992) İnceltmede amaç resmin iskeletini olabildiğince düzgün bir şekilde, bozulma olmadan çıkarmak ve resmi bir piksel genişliğine indirmektir. İnceltme işlemi parmakizi analizleri, el yazısı tanımlama, imza doğrulama, şekil tanımlama, örüntü tanımlama, biomedikal sistemlerde ve tıbbi görüntüleme işlemlerinde yoğun olarak kullanılmaktadır. (Kwok, 1988) İnceltme işlemi; işleme girecek resim verilerinin miktarını azaltır. Ayrıca inceltmiş resim orijinal resmin ayırt edici özelliklerini taşımakta olduğundan inceltmiş resimler üzerinde işlem yapmak daha hızlı olmaktadır. İnceltme algoritmaları Rosenfeld'e (Kwok, 1988) göre paralel ve seri olmak üzere ikiye ayrılır. Paralel algoritmalar tüm piksellere simültane olarak uygulanır ve sonuç elde edilir. Sonuç olarak elde edilen resim tekrardan iterasyona sokulur. İterasyona giren resim ile iterasyon sonucu oluşan resim aynı olana kadar işlem tekrarlanır. Seri algoritmalar ise resim taranırken piksel kendisinden önceki pikselin iterasyon sonunda aldığı değer üzerinden işleme sokulur.

Bu çalışmanın kapsamı, siyah beyaz resimler üzerinde çalışan paralel inceltme algoritmalarından en çok kullanılan dört tanesinin (Zhang ve Suen, 1984; Wang ve Zhang, 1989, 1992; Raju ve Xu, 1991; Lam vd., 1992; Suen ve Wang, 1994; Kwon vd., 2001) parmakizi resimleri üzerinde gerçekleştirilmesi, bunların sonuçlarının karşılaştırılması ve bu algoritmalar ile inceltmiş resimlerin tek piksel genişliğine düşürülebilmesi için önerilen ve geliştirilmeye açık iki farklı algoritmadan oluşmaktadır. Bu çalışmada bir piksel bir byte olan gri tonlarında ve 512*512 boyutlarındaki parmakizi resimleri (Watson, 1992) ve onların 128*128 'lik kesitleri kullanılmıştır. Ayrıca tez içinde dijital görüntü işleminin temel işlemlerine ve parmakizi resimlerinin iyileştirilmesinde büyük önem taşıyan Yönlü Fourier Filtresi 'de (Sherlock, 1992, 1994) açıklanmıştır. Bu yöntemde birtakım değişiklikler

yapılarak daha iyi sonuçlar alınmaya çalışılmıştır. Tüm bu çalışmalar C/C++ ile yazılmış programlar aracılığı ile gerçekleştirilmiş olup kaynak kodları mevcuttur.

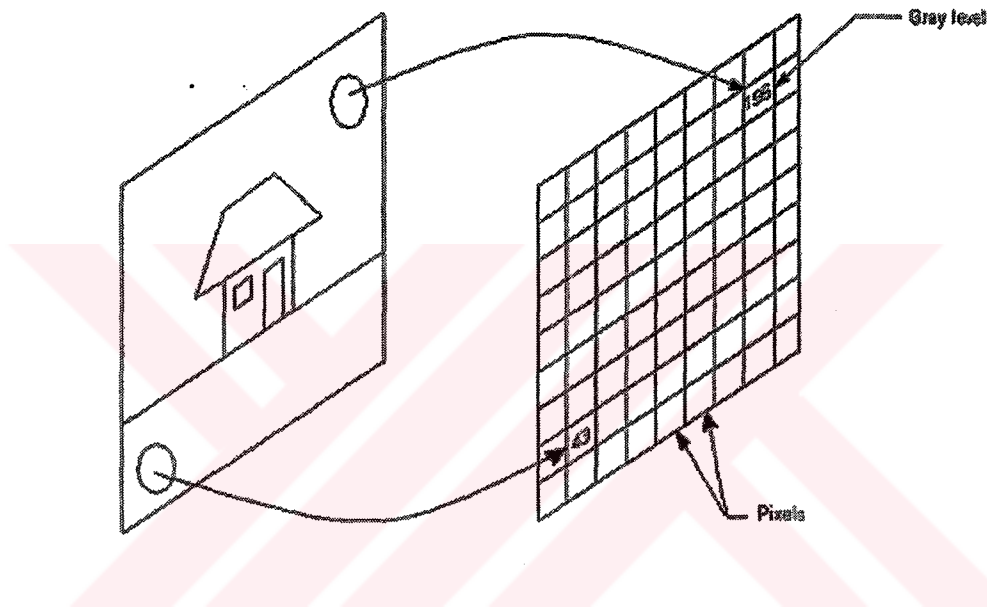
Tez; Sırasıyla giriş, dijital görüntü tanımı, dijital görüntü işlemenin temel algoritmalarına genel bakış, Fourier dönüşümü ve yönlü Fourier filtresi, inceltme algoritmaları, tek piksele indirgeme algoritmaları ve sonuçlar olarak yedi bölümden oluşmaktadır. Bundan sonraki bölümde dijital görüntünün tanımını verip, bu çalışmada kullanılan resimlerin formatı olan Bitmap Picture (BMP) tanıtılacaktır.



2. DİJİTAL GÖRÜNTÜ TANIMI

2.1 Dijital Görüntü İşleme ve Dijital Görüntü Tanımı

Dijital (Sayısal) görüntü işleme, iki boyutlu resimlerin bilgisayar ile üzerinde bazı işlemlerin yapılmasıdır. Bu işlemler filtreleme, renk değiştirme, sıkıştırma, ve inceltme gibi örneklendirilebilir. Dijital görüntü (resim) ise sonlu sayıda bit (byte) tarafından temsil edilen reel yada kompleks sayı dizisidir. (Jain, 1989) Bu iki boyutlu dizinin her bir elemanına bir piksel denir ve buda Dijital görüntünün en küçük elemanıdır.



Şekil 2.1 Resim ve dijital gösterimi (Castleman, 1996)

Bir pikselin alacağı değer pikselin bellekte ne kadar yer tuttuğu ile ilgilidir. Ayrıca pikselin değeri temsil ettiği rengin resmin renk paletinde o renge karşılık gelen sayısal değeridir. Dijital resimler bir pikselin bellekte kaç bit veya byte yer tuttuğuna göre kabaca çizelge 2.1 'deki gibi sınıflandırılabilir.

Çizelge 2.1 Dijital resimlerin sınıflandırılması

Özellikler	1 Piksel	1 Piksel	1 Piksel	1 Piksel
Bit	1	4	8	24
Byte	1/8	4/8	8/8	24/8
Renk Sayısı	2	16	256	256 ³

Bu çalışmada kullandığımız resimler bir piksel bir byte olan, 256 renkli, gri tonlarında BMP formatında parmakizi resimleridir. Şekil 2.2 'de bu tezde kullanılan test resminin küçültülmüş şekli görülmektedir.



Şekil 2.2 Tezde kullanılan parmakizi resminin 256*256 boyutlarındaki görünümü.

2.2 Bitmap (BMP) Resim Dosyasının Yapısı

Bitmap resim dosyalarının yapısı; Header(Başlık) + Renk Paleti + Resim Verisi şeklindedir.

Renk paleti bütün resimlerde bulunmayabilir. Dosyanın başlığı ise iki alt kısma ayrılır;

Bitmap Dosya Başlığı(Bitmap File Header) ve Bitmap Bilgi Başlığı (Bitmap Info Header)

2.2.1 Bitmap Dosya Başlığı (Bitmap File Header)

Resim dosyasının yapısı ile ilgili bilgi verir. Dosyanın ilk ondört bytelık kısmını kapsar.

Dosya başlığının bileşenleri aşağıdaki Çizelge 2.2 'de verilmiştir.

Çizelge 2.2 Bitmap dosya başlığı

Bileşenin Adı	Kapladığı Alan	Açıklaması
Bftype	2 bytes	Dosyanın BMP tipinde olduğunu gösterir.
Bfsize	4 bytes	Dosyanın boyutunu byte cinsinden gösterir.
Bfreserved1	2 bytes	Rezerv edilmiş alandır.Genellikle sıfır değerini gösterir.
Bfreserved2	2 bytes	Rezerv edilmiş alandır.Genellikle sıfır değerini gösterir.

Bfoffbits	4 bytes	Resim verisinin başladığı byte değerini gösterir.
-----------	---------	---

2.2.2 Bitmap Bilgi Başlığı(Bitmap Info Header)

Resim ile ilgili bilgileri verir. Toplam kırk bytelik yer kaplar. Bileşenleri aşağıdaki Çizelge 2.3 'de verilmiştir.

Çizelge 2.3 Bitmap bilgi başlığı

Bileşenin Adı	Kapladığı Alan	Açıklaması
BiSize	4 bytes	Bilgi başlığının boyutunu gösterir.
BiWidth	4 bytes	Resmin genişliğini gösterir.
BiHeight	4 bytes	Resmin yüksekliğini gösterir.
BiPlanes	2 bytes	Resmin boyut sayısını gösterir.
BiBitCount	2 bytes	Bir pikselin kaç bit olduğunu gösterir.
BiCompression	4 bytes	Resimde sıkıştırma olup olmadığını gösterir.
BiSizeImage	4 bytes	Resim datasının büyüklüğünü gösterir.
BiXPelsPerMeter	4 bytes	X yönündeki çözünürlük
BiYPelsPerMeter	4 bytes	Y yönündeki çözünürlük.
BiClrUsed	4 bytes	Resimde fiili olarak kullanılan renk sayısı
BiClrImportant	4 bytes	Resmin gösterimi için önemli kabul edilen renk sayısı

Bu bölümde dijital görüntü ve BMP dosyalarının formatı kısaca özetlenmiştir. Bir sonraki bölümde ise dijital görüntü işlemenin temel algoritmalarından görüntünün histogramı, histogram eşitleme ve thresholding konuları tanıtılacaktır.

3. DİJİTAL GÖRÜNTÜ İŞLEMENİN TEMEL ALGORİTMALARI

3.1 Dijital Görüntünün Histogramı

Resmin histogramı her bir gri seviyesi için, resim içerisinde o gri seviyesine sahip kaç tane piksel olduğunu gösteren bir fonksiyon olarak tanımlanabilir. Histogram her bir gri seviyesinin resim içerisindeki yoğunluğunu gösteren bir fonksiyon olarak da tanımlanabilir (Castleman, 1996 ; Pitas,1993 ; Gonzales ve Woods, 1993).

Dijital resim L farklı gri renk piksel değerine sahip olsun. Bu değer bir piksel bir byte olan resimler için genellikle 0 ile 255 arasındadır. r_k k. gri seviye, n_k k. gri seviyeye sahip resimdeki toplam piksel sayısı, n 'de resimdeki toplam piksel sayısı olmak üzere; histogram fonksiyonu

$$p(r_k) = \frac{n_k}{n}, k = 0,1,\dots,L-1 \quad 3.1$$

şeklinde tanımlanabilir.

Resmin histogramı resim hakkında önemli bilgiler verir. Histogram eğrisine bakılarak resim hakkında bazı çıkarımlar yapılabilir. Eğer resimde küçük gri seviyeler daha yoğun ise “Resim siyah ağırlıklı ve koyu tonlardadır”, resimde büyük gri seviyeler daha yoğun ise “Resim beyaz ağırlıklı ve parlaktır” denir.

Resmin renk kalitesi düşükse ve renk dağılımı düzgün değilse resmin histogramı üzerinde değişiklik yapılarak resmin kalitesi yükseltilebilir, bir başka deyişle, resim iyileştirilebilir. Bu işlem için histogram eşitleme işlemi popüler olarak kullanılmaktadır. Histogram eşitleme temelleri olasılık teorisini gerektirmektedir. Histogram eşitleme yöntemi gri ton değerlerini tekrar dağıtmayı hedefleyen bir yöntemdir. Bu yöntemde piksel sayısının fazla olduğu gri ton değerlerine daha geniş bir aralık, piksel sayısının az olduğu gri ton değerlerine daha dar bir aralık ayrılır. Histogram eşitleme yönteminde sonuç görüntünün histogramı mümkün olduğu kadar düz bir dağılıma sahiptir. Bu yöntemde kullanılan dönüşüm fonksiyonu Formül 3.2 ve formül 3.3 'deki gibi tanımlanmıştır.

$$C(i) = \sum_{k=0}^i n_k \quad i = 0,1,\dots,L-1 \quad k = 0,1,\dots,L-1 \quad 3.2$$

n_k , k. gri seviyesindeki toplam piksel sayısı ve $C(i)$ ise resmin histogram değerlerinin birikimli olarak ifadesidir. Bu aynı zamanda kümülatif frekans olarak bilinmektedir. i gri

seviyelerinin yerine histogram eşitlenmiş yeni gri seviyeleri, aşağıdaki I formülünün yardımıyla hesaplanmaktadır.

$$I = C(i) \times \frac{L-1}{n} \quad 3.3$$

n toplam piksel sayısı, L-1 maksimum gri seviye ve I histogram eşitlenmiş yeni gri seviyeyi temsil etmektedir.

Şekil 3.1 'de test resminin orijinali ve histogram eşitlenmiş halleri görülmektedir.



Şekil 3.1 Test resmi ve histogram eşitlenmiş halinin 256*256 boyutlarındaki görüntüleri

3.2 Thresholding

Thresholding renkli resimleri iki renge (siyah-beyaza) dönüştürme yöntemidir. Çok değişik şekilde uygulamaları vardır. Bu çalışmada RAT(Relational Average Thresholding) (Emiroğlu, 1997, 1997a) yöntemi kullanılmıştır. Bu yöntemde, threshold edilecek resmi alt bölgelere bölüp, her bir bölgenin ortalaması hesaplanarak, bu ortalamanın üstünde olanlar beyaz (255), altında olanlar ise siyah (0) olacak şekilde tüm resim taranır. Bu işlem Formül 3.4 'deki gibi ifade edilebilir.

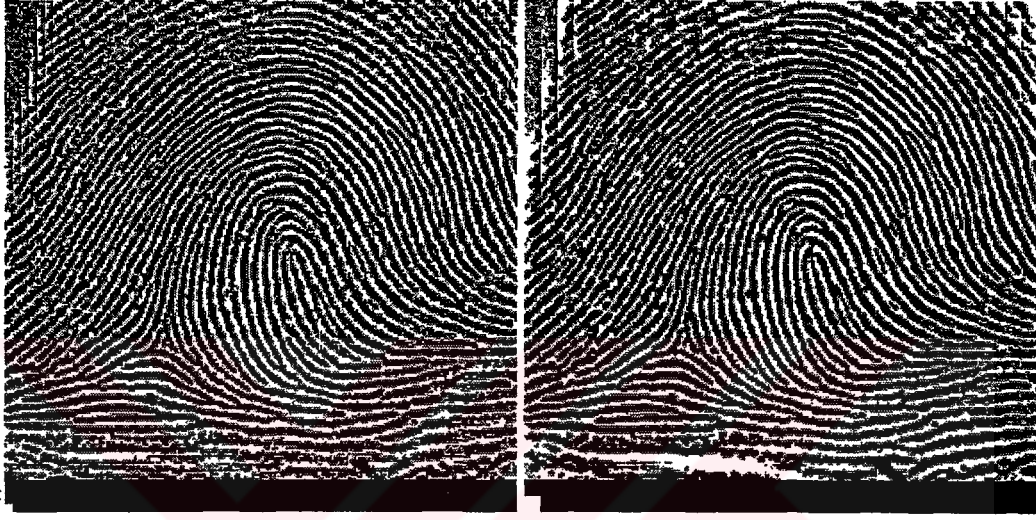
$$G(i, j) = \begin{cases} 255 & F(i, j) > T \\ 0 & F(i, j) \leq T \end{cases} \quad (i = 0, 1, \dots, N, j = 0, 1, \dots, M) \quad 3.4$$

Burada F(i, j) orijinal resmi, G(i, j) threshold edilmiş siyah-beyaz resmi, N ve M resmin yüksekliği ve genişliğini, T ise threshold seviyesini göstermektedir. T bu yöntemde her bir alt

bölgenin aritmetik ortalamasına eşittir.(Formül 3.5)

$$T = \frac{1}{N * M} \sum_{i=0}^N \sum_{j=0}^M F(i, j) \quad 3.5$$

Buradaki N ve M alt bölgenin yükseklik ve genişliğini göstermektedir. Alt bölgeler 8x4,8x8,16x16,32x32 v.b... olacak şekilde seçilebilir. Şekil 3.2’de test resminin 8x8 ve 16x16 alt bölgeler seçilerek oluşturulmuş siyah-beyaz halleri gösterilmiştir.



Şekil 3.2 Solda 8x8 ‘lik, sağda 16x16 ‘lık alt bölgeler ile yöntemin test resmi üzerinde uygulanmış görüntüleri

4. FOURIER DÖNÜŞÜMÜ ve YÖNLÜ FOURIER FİLTRESİ (YFF)

4.1 Fourier Dönüşümü

$f(x)$, reel x 'lerden oluşan sürekli fonksiyon olsun. $f(x)$ 'in Fourier dönüşümü $\mathfrak{F}\{f(x)\}$ şeklinde gösterilir ve $j = \sqrt{-1}$ olmak üzere;

$$\mathfrak{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux] dx \quad (4.1)$$

şeklindedir. $F(u)$ 'nun verilmesiyle $f(x)$ ters Fourier dönüşümünden çıkarılabilir.

$$\begin{aligned} \mathfrak{F}^{-1}\{F(u)\} &= f(x) \\ &= \int_{-\infty}^{\infty} F(u) \exp[j2\pi ux] du \end{aligned} \quad (4.2)$$

(4.1) ve (4.2) eşitlikleri fourier dönüşüm çiftidir. $f(x)$ sürekli ve integrallenebilir ve $F(u)$ nun integrallenebilir olduğu durum için vardır.

Konu boyunca $f(x)$ reel olarak kabul edilmiştir. Reel fonksiyonların fourier dönüşümü genelde kompleksdir.

$$F(u) = R(u) + jI(u) \quad (4.3)$$

$R(u)$ ve $I(u)$, $F(u)$ 'nun sırasıyla reel ve imajiner bileşenleridir. (4.3)'ün üssel formu;

$$F(u) = |F(u)| e^{i\phi(u)} \quad (4.4)$$

şeklindedir. Burada;

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad (4.5)$$

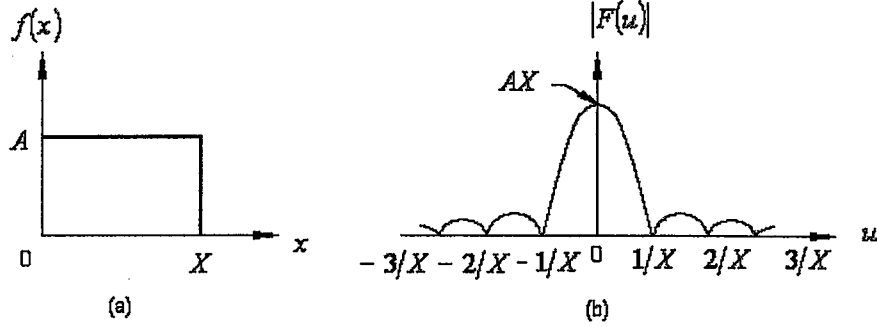
$$\phi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right] \quad (4.6)$$

$|F(u)|$; $f(x)$ 'in Fourier spektrumu, $\phi(u)$ ise faz açısı olarak adlandırılır. Spektrumun karesi $f(x)$ 'in güç spektrumu olarak isimlendirilir. Bu değer spektral yoğunluk olarakta adlandırılmıştır.

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u) \quad (4.7)$$

Fourier dönüşümünde geçen değişken u frekans değişkenidir. Bu isim üsselin tanımından gelir.

$$\exp[-j2\pi ux] = \cos 2\pi ux - j \sin 2\pi ux \quad (4.8)$$



Şekil 4.1 $f(x)$ fonksiyonu ve Fourier dönüşümü

4.2 Ayrık Fourier Dönüşümü

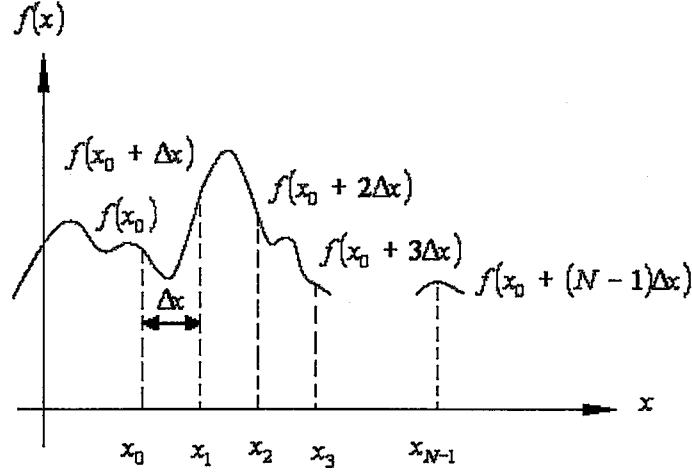
$f(x)$ sürekli fonksiyonunu

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + [N - 1]\Delta x)\}$$

dizisine ayrıştırdığımızı düşünelim. Burada Şekil 4.2'te gösterildiği gibi ΔX birim aralıklı N örnek alınmıştır. Daha sonraki gelişmelerde X 'in ayrık ya da sürekli değerde olması konunun bağlamına uygun olarak seçilecektir. X 'i ayrık $0, 1, 2, \dots, N - 1$ değerlerinde tanımladığımızda aşağıdaki bağıntı geçerli olacaktır.

$$f(x) = f(x_0 + x\Delta x) \quad (4.9)$$

Başka bir deyişle $\{f(0), f(1), f(2), \dots, f(N - 1)\}$ dizisi uygun sürekli fonksiyondan herhangi N boşluklu örneklerle yazılabilir.



Şekil 4.2 Sürekli fonksiyonun örnekleme

Yukarıdaki tanımlamayı hatırlayarak örnekli fonksiyonlarda uygulanan ayrık Fourier dönüşüm çifti, $u = 0,1,2,\dots, N-1$ ve $x = 0,1,2,\dots, N-1$ için

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \quad (4.10)$$

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \quad (4.11)$$

şekindedir. Ayrık Fourier dönüşümündeki (4.10 eşitliğindeki) $u = 0,1,2,\dots, N-1$ değerleri; $0, \Delta u, 2\Delta u, \dots, (N-1)\Delta u$ değerlerindeki sürekli dönüşümün örneklerine uygundur. Başka bir deyişle $F(u); F(u\Delta u)$ şeklinde gösterilebilir. $F(u)$ 'nun örneklerinin frekans ekseninin orjinenin başladığı durum dışında, bu notasyon ayrık $f(x)$ için kullanılanla benzerdir. Δu ve Δx terimleri arasında bir ilişki vardır ve bu ilişki:

$$\Delta u = \frac{1}{N\Delta x} \quad (4.12)$$

şeklinde tanımlanır.

İki değişkenli durum için ayrık fourier dönüşüm çifti, $u = 0,1,2,\dots, M-1$ ve $v = 0,1,2,\dots, N-1$ için aşağıdaki gibidir;

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux/M + vy/N)] \quad (4.13)$$

$x = 0, 1, 2, \dots, M - 1$ ve $y = 0, 1, 2, \dots, N - 1$ için;

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[j2\pi(ux/M + vy/N)] \quad (4.14)$$

Görüntü kare dizide örneklenirse, $M = N$ ve $u, v = 0, 1, 2, \dots, N - 1$ için;

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux + vy/N)] \quad (4.15)$$

şeklindedir. Burada $f(x, y)$, uzaysal verilerdeki görüntüdür, exp. terim ise fourier uzayındaki her bir $F(u, v)$ noktasına uygun ana fonksiyondur. Yukarıdaki eşitlik; her $F(u, v)$ noktasının değerinin, uzaysal görüntünün, uygun temel fonksiyon ile çarpılıp, çıkan sonuçların toplanması şeklinde elde edileceğini ifade etmektedir.

Ana fonksiyonlar artan frekanslı sinus ve cosinus dalgalarıdır. Örneğin $F(0, 0)$ ortalama parlaklığa uygun görüntünün ayrık bileşenini ifade ederken, $F(N - 1, N - 1)$; en yüksek frekansı gösterir.

Aynı şekilde, Fourier görüntüsü tekrar uzaysal alana dönüştürülebilir.

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp[j2\pi(ux + vy/N)] \quad (4.16)$$

Yukarıdaki işlemin sonucunu elde etmek için Fourier dönüşümünün bölünebilirlik özelliğinden;

$$F(u, v) = \sum_{y=0}^{N-1} P(u, y) \exp(-j2\pi vy/N) \quad (4.17)$$

$$P(u, y) = \frac{1}{N} \sum_{x=0}^{N-1} f(x, y) \exp(-j2\pi ux/N) \quad (4.18)$$

faidalanılır. Bu formüllerin kullanımında, ilk olarak uzaysal verilerdeki görüntüden N bir-boyutlu Fourier dönüşümü ile ara görüntü elde edilir. Bu ara görüntüye, son görüntüyü elde etmek için, tekrar N bir-boyutlu Fourier dönüşümü uygulanır. Bu şekildeki iki-boyutlu Fourier dönüşümü, 2N bir-boyutlu Fourier dönüşümüne dönüştürülerek işlemlerde azalma

sağlanır.

4.3 İki Boyutlu Fourier Dönüşümünün Özellikleri

4.3.1 Bölünebilme

(4.13) ve (4.14) eşitliklerinden oluşan ayrık Fourier dönüşümü bölünmüş formda ifade edilebilir. $u, v = 0, 1, \dots, N - 1$ için;

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \exp[-j2\pi ux/N] \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi vy/N] \quad (4.19)$$

$x, y = 0, 1, \dots, N - 1$ için;

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \exp[j2\pi ux/N] \sum_{v=0}^{N-1} F(u, v) \exp[j2\pi vy/N] \quad (4.20)$$

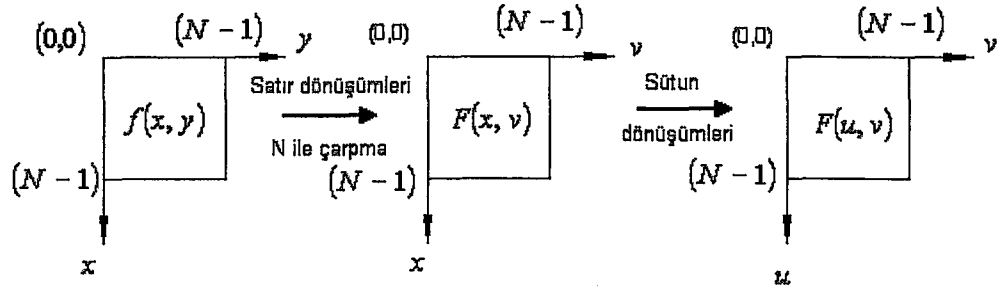
Ayrılabilirlik özelliğinin en temel avantajı $F(u, v)$ ya da $f(x, y)$ 'nin 1-D Fourier dönüşümü ya da ters dönüşümünden iki adımda birbirini izleyen uygulamalarla elde edilebilmesidir.

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} F(x, v) \exp[-j2\pi ux/N] \quad (4.21)$$

Burada;

$$F(x, v) = N \left[\frac{1}{N} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi vy/N] \right] \quad (4.22)$$

Herbir x değeri için, (4.22) 'teki parantez içi ifade; frekans değerleri $v = 0, 1, \dots, N - 1$ olan bir 1-D dönüşümdür. Bu yüzden 2-D fonksiyonu olan $F(x, v)$; $f(x, y)$ 'nin her satırı boyunca dönüşüm alınıp, çıkan sonuç N ile çarpılarak elde edilebilir. İstenen sonuç $F(u, v)$, (4.3) 'te gösterildiği gibi $F(x, v)$ 'nin her sütunu boyunca dönüşüm olarak elde edilir. Bu işlemler Şekil 4.3 de özetlenmiştir.



Şekil 4.3 1-D dönüşümler serisi olarak 2-D Fourier dönüşümünün hesaplanması

Aynı sonuç $f(x, y)$ 'nin sütunları boyunca birinci dönüşüm daha sonra çıkan sonucun satırları boyunca dönüşüm uygulanarak elde edilebilir. (Ballard, 1982)

4.3.2 Dağılım ve Scaling

Sürekli ve ayrık dönüşüm çiftinin tanımından;

$$\mathfrak{F}\{f_1(x, y) + f_2(x, y)\} = \mathfrak{F}\{f_1(x, y)\} + \mathfrak{F}\{f_2(x, y)\} \quad (4.23)$$

ve genel olarak;

$$\mathfrak{F}\{f_1(x, y) \cdot f_2(x, y)\} \neq \mathfrak{F}\{f_1(x, y)\} \cdot \mathfrak{F}\{f_2(x, y)\} \quad (4.24)$$

Diğer bir deyişle Fourier dönüşümü ve ters dönüşümü toplama üzerine dağılıma özelliğine sahiptir ama çarpımda dağılıma özelliği yoktur.

a ve b skalerleri için ;

$$af(x, y) \Leftrightarrow aF(u, v) \quad (4.25)$$

$$f(ax, by) \Leftrightarrow \frac{1}{ab} F(u/a, v/b) \quad (4.26)$$

4.3.3 Ortalama Değer

2-D ayrık fonksiyonların ortalama değerleriyle ilgili en geniş kullanıma sahip tanımlamasıdır ve aşağıdaki ifadeden hesaplanabilir;

$$\bar{f}(x, y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad (4.27)$$

$u = v = 0$ (3.19) eşitliğinde yerine yazılırsa;

$$F(0,0) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \quad (4.28)$$

olur. Bu yüzden $\bar{f}(x, y)$ ile $f(x, y)$ 'nin Fourier dönüşümü arasında aşağıdaki ilişki vardır ve bu ilişki aşağıdaki şekilde ifade edilmiştir;

$$\bar{f}(x, y) = \frac{1}{N} F(0,0) \quad (4.29)$$

4.4 Hızlı Fourier Dönüşümü (The Fast Fourier Transform)

(4.10) eşitliğinin gerektirdiği kompleks çarpma ve toplama sayısı N^2 ile orantılıdır. u 'nun N tane değerinin herbiri için toplamının büyümesi, $f(x)$ ile N adet kompleks çarpımını ve sonuca $N - 1$ adet eklemeyi gerektirir. $\exp[-j2\pi ux/N]$ 'in terimleri daha sonraki uygulamalar için hesaplanarak $\exp[-j2\pi ux/N]$ saklanabilir. Bu sebepten dolayı terimlerin içindeki u nun x ile çarpımı, uygulamanın doğrudan bir parçası olarak kabul edilmez.

(4.10) eşitliğinin uygun ayrıştırması toplama ve çarpımların sayısı $N \log_2^N$ ile orantılı olarak yapılabilir. Ayrıştırma prosedürü “*Hızlı Fourier dönüşümü (FFT)*” olarak adlandırılır. Bu orandaki azalma Tablo 5.1 ‘de görüldüğü gibi hesaplamada harcanan çabada dikkate değer bir azaltma sağlayacaktır. FFT özellikle N ‘in daha büyük olduğu durumlarda Fourier dönüşümünün doğrudan uygulaması üzerinde önemli şekilde hesaplamada avantaj sağlar. Örneğin 8192 noktalı bir dizi için FFT 5 saniyede hesaplama yaparken, aynı teknik şartlar altında (4.10) eşitliği ile Fourier dönüşümü aynı dizi için 600 kez daha fazla (50 dakika) zaman alır.

Bir sonraki konuda FFT algoritması 1-D ‘de geliştirilecektir. (Andrews, 1977)

4.4.1 FFT Algoritması

Bu bölümde FFT Algoritması ardıl ikiye katlama yöntemi kullanılarak yapılacaktır. (4.10) eşitliği uygun olarak aşağıdaki şekilde yazılır.

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux} \quad (4.30)$$

$$W_N = \exp[-j2\pi/N] \quad (4.31)$$

Çizelge 4.1 1 N değişkenleri için $N \log_2^N$ 'e karşı N^2 nin karşılaştırılması

N	N^2 (doğrudan FFT)	$N \log_2^N$ (FFT)	(N/\log_2^N) Hesaplama avantajı
2	4	2	2.00
4	16	8	2.00
8	64	24	2.67
16	256	64	4.00
32	1,024	160	6.40
64	4,096	384	10.67
128	16,384	896	18.29
256	65,536	2,048	32.00
512	262,144	4,608	56.89
1024	1,048,576	10,240	102.40
2048	4,194,304	22,528	186.18
4096	16,777,216	49,152	341.33
8192	67,108,864	106,496	630.15

n pozitif tamsayı olmak üzere, N 'nin aşağıdaki formda kabul edilir.

$$N = 2^n \quad (4.32)$$

Buradan N ifadesi M 'de pozitif tamsayı olmak üzere aşağıdaki şekli alır.

$$N = 2M \quad (4.33)$$

(4.33)'ü (4.30) 'de yerine yazarsak aşağıdaki sonuç bulunur:

$$F(u) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x)W_{2M}^{ux} \quad (4.34)$$

$$= \frac{1}{2} \left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x)W_{2M}^{u(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1)W_{2M}^{u(2x+1)} \right]$$

(4.31) 'den , $W_{2M}^{2ux} = W_M^{ux}$ olduğundan dolayı (4.34) aşağıdaki formda ifade edilebilir:

$$F(u) = \frac{1}{2} \left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x)W_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1)W_M^{ux}W_{2M}^u \right] \quad (4.35)$$

$u = 0,1,2,\dots, M - 1$ için aşağıdaki tanımlama ile;

$$F_{\text{çift}}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x)W_M^{ux} \quad (4.36)$$

ve tekrar $u = 0,1,2,\dots, M - 1$ için ;

$$F_{\text{tek}}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1)W_M^{ux} \quad (4.37)$$

(4.35) 'i indirgersek;

$$F(u) = \frac{1}{2} [F_{\text{çift}}(u) + F_{\text{tek}}(u)W_M^{ux}] \quad (4.38)$$

$W_M^{u+M} = W_M^u$ ve $W_{2M}^{u+M} = -W_{2M}^u$ olduğu için, (4.36) ve (4.38) bize aşağıdaki eşitliği verir:

$$F(u+M) = \frac{1}{2} [F_{\text{çift}}(u) - F_{\text{tek}}(u)W_{2M}^u] \quad (4.39)$$

(4.36)-(4.39) eşitlikleri üzerinde dikkatli bir inceleme yapılacak olursa, bu ifadelerden bazı ilginç özellikler çıkarılabilir. (4.38) ve (4.39) 'da belirtildiği gibi, N -nokta dönüşümü orjinal ifadeyi iki kısma ayırarakta hesaplanabilir. $F(u)$ 'nun ilk yarısının hesabı (4.36) ve (4.37) 'de verilen iki $(N/2)$ -nokta değerlendirmesini gerektirir. $F_{\text{çift}}(u)$ ve $F_{\text{tek}}(u)$ sonucu çıkan değerler $u = 0,1,2,\dots, (N/2 - 1)$ için $F(u)$ 'yu elde etmek üzere (4.38) 'de yerleştirilir. Diğer yarı yalnızca (4.39) 'dan ek dönüşüm değerlendirmeleri olmadan hesaplanır.

Bu prosedürün işlemsel anlamını incelemek için, $m(n)$ ve $a(n)$ ifadeleri sırasıyla kompleks çarpma ve toplama sayılarını temsil etsin. Bundan önce n pozitif tamsayı olmak üzere, örneklerin sayısı 2^n dir. Öncelikle $n = 1$ alalım. İki nokta dönüşümü $F(0)$ 'ı daha sonrada (4.39)'u takiben $F(1)$ 'i gerektirir. $F(0)$ 'ı elde etmek için $F_{\text{tek}}(0)$ ve $F_{\text{çift}}(0)$ 'in hesaplanması gerekir. Bu durumda $M = 1$ ve (4.36), (4.37) eşitlikleri bir-nokta dönüşümleridir. Çünkü bir tek noktanın Fourier dönüşümü kendisinin örneğidir. Böylece $F_{\text{çift}}(0)$ ve $F_{\text{tek}}(0)$ 'ı elde etmek için çarpma yada toplama gerekli değildir. $F_{\text{tek}}(0)$ 'ın W_2^0 ile bir çarpımı ve bir toplama (4.38)'dan $F(0)$ 'ı verir. Böylece $F(1)$, fazladan bir toplama (çıkarma toplama ile aynı kabul edilir) ile (4.39)'un sonucu olarak elde edilir. $F_{\text{tek}}(0)W_2^0$ 'ın

önceden hesaplanmış olması durumunda, iki-nokta dönüşümünün toplam işlem sayısı $m(1) = 1$ adet çarpma, $a(1) = 2$ adet toplama'dır.

n için bir sonraki değer 2 olsun. Yukarıdaki gelişmelere uygun olarak dört-nokta dönüşümü iki parçaya bölünebilir. $F(u)$ 'nin ilk yarısı iki için hesaplama (iki-nokta dönüşümü), gerektirir ($M = 2$ için (4.36) ve (4.37) 'de olduğu gibi). İki nokta dönüşümü $m(1)$ adet çarpma ve $a(1)$ adet toplama gerektirir. Bu yüzden iki denklemin hesaplanması toplamda $2m(1)$ çarpma ve $2a(1)$ toplama gerektirir. $F(0)$ ve $F(1)$ 'i (4.38) 'den elde etmek için iki fazla toplama ve çarpma gerekmektedir. Çünkü $u = \{0,1\}$ için $F_{tek}(u)W_{2M}^u$ değeri önceden hesaplandığı için, fazladan iki toplama $F(2)$ ve $F(3)$ 'ü verir. Toplam alınırsa; $m(2) = 2m(1) + 2$ ve $a(2) = 2a(1) + 4$ olur.

$n = 3$ olduğunda, iki dört-nokta dönüşümü $F_{tek}(u)$ ve $F_{çift}(u)$ 'nin hesaplanmasında işleme katılır. Bu işlemler $2m(2)$ çarpma ve $2a(2)$ toplama gerektirir. Dört fazla çarpma ve sekiz fazla toplama ile dönüşüm tamamlanır. Toplam alınırsa; $m(3) = 2m(2) + 4$ ve $a(3) = 2a(2) + 8$ olur.

n 'nin herhangi pozitif tamsayı değerleri için hesaplamalara devam edilirse FFT'yi tamamlamak için gereken toplama ve çarpmaların sayısı için tekrarlanan ifadelerle karşılaşılır:

$$m(n) = 2m(n - 1) + 2^{n-1} \quad n \geq 1 \quad (4.40)$$

$$a(n) = 2a(n - 1) + 2^n \quad n \geq 1 \quad (4.41)$$

burada $m(0) = 0$ ve $a(0) = 0$ 'dır, çünkü tek noktanın dönüşümü çarpma ya da toplama gerektirmez.

(4.36)-(4.39) denklemlerinin uygulaması bir ardıl ikiye katlama FFT algoritması oluşturur. Bu ise, iki adet bir-nokta dönüşümünden bir adet iki-nokta dönüşümü, iki adet iki-nokta dönüşümünden bir adet dört-nokta dönüşümü hesaplanmasından gelmektedir. Böylece devam edilirse, herhangi N için 2 'nin kuvvetine eşit olacak sayıda dönüşüm gerektirir.

4.4.2 İşlemlerin sayısı

Kompleks çarpma ve toplama işlem sayısı tümevarım ile FFT algoritmasının sırasıyla aşağıda

ki uygulamalarını gerektirir.

$$\begin{aligned}
 m(n) &= \frac{1}{2} 2^n \log_2 2^n \\
 &= \frac{1}{2} N \log_2 N \\
 &= \frac{1}{2} Nn \quad n \geq 1
 \end{aligned} \tag{4.42}$$

$$\begin{aligned}
 a(n) &= 2^n \log_2 2^n \\
 &= N \log_2 N \\
 &= Nn \quad n \geq 1
 \end{aligned} \tag{4.43}$$

Tümevarım ile ispat etmek için (4.42) ve (4.43) eşitliklerinin $n = 1$ için doğru olduğu gösterilir;

$$m(1) = \frac{1}{2} (2)(1) = 1 \text{ ve } a(1) = (2)(1) = 2$$

Sonra ifadenin n için doğru olduğu kabul edilerek, $n + 1$ için doğruluğu ispatlanır. (4.40)den;

$$m(n + 1) = 2m(n) + 2^n$$

n için geçerli olduğu kabul edilerek $m(n)$, (4.42) 'te yerine konulursa;

$$\begin{aligned}
 m(n + 1) &= 2 \left(\frac{1}{2} Nn \right) + 2^n \\
 &= 2 \left(\frac{1}{2} 2^n n \right) + 2^n \\
 &= 2^n (n + 1) \\
 &= \frac{1}{2} (2^{n+1}) (n + 1)
 \end{aligned}$$

böylece (4.42) eşitliği n 'nin bütün pozitif tamsayı değerleri için geçerli olur. (4.41)'den;

$$a(n + 1) = 2a(n) + 2^{n+1}$$

$a(n)$, (4.43) 'te yerine yazılırsa ispat tamamlanmış olur.

$$\begin{aligned}
a(n+1) &= 2Nn + 2^{n+1} \\
&= 2(2^n n) + 2^{n+1} \\
&= 2^{n+1}(n+1)
\end{aligned}$$

4.4.3 Ters FFT

Ayrık ileri dönüşümü sağlayan herhangi bir algoritma aynı zamanda girdilerde çok küçük değişikliklerle tersini hesaplamak için uygulanabilir, bunu göstermek için (4.2) ve (4.3) eşitliklerine dönelim. Tekrar edersek;

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \quad (4.44)$$

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \quad (4.45)$$

(4.45)'nin kompleks konjugate'ini alarak her iki tarafı N ile bölersek;

$$\frac{1}{N} f^*(x) = \frac{1}{N} \sum_{u=0}^{N-1} F^*(u) \exp[-j2\pi ux/N] \quad (4.46)$$

Bu sonucun (4.44) ile karşılaştırılması (4.46) 'de eşitliğin sağ tarafının ileri Fourier dönüşümü formunda olduğunu gösterir. İleri Fourier dönüşümünü hesaplamak üzere tasarlanan algoritmada $F^*(u)$ 'yu yerleştirmek, $f^*(x)/N$ miktarını verir. Kompleks konjugate'tinin alınması ve N ile bölünmesi $f(x)$ 'in tersini verir.

2-D kare dizi için (4.6)'un kompleks konjugate'ti alınır,

$$f^*(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F^*(u, v) \exp[-j2\pi(ux + vy/N)] \quad (4.47)$$

elde edilir. Bu ifade (4.5) 'un 2-D ileri dönüşümünün formundadır. Bu yüzden ileri dönüşümü hesaplamak için hazırlanan bir algoritmada $F^*(u, v)$ 'nin yerleştirilmesi, $f^*(x, y)$ 'yi verir. Bu ifadenin Kompleks konjugate 'tinin alınması $f(x, y)$ 'yi verecektir. $f(x)$ ya da $f(x, y)$ reel olduğunda Kompleks konjugate alınması gereksizdir çünkü reel fonksiyonlarda $f(x) = f^*(x)$ ve $f(x, y) = f^*(x, y)$ 'dir.

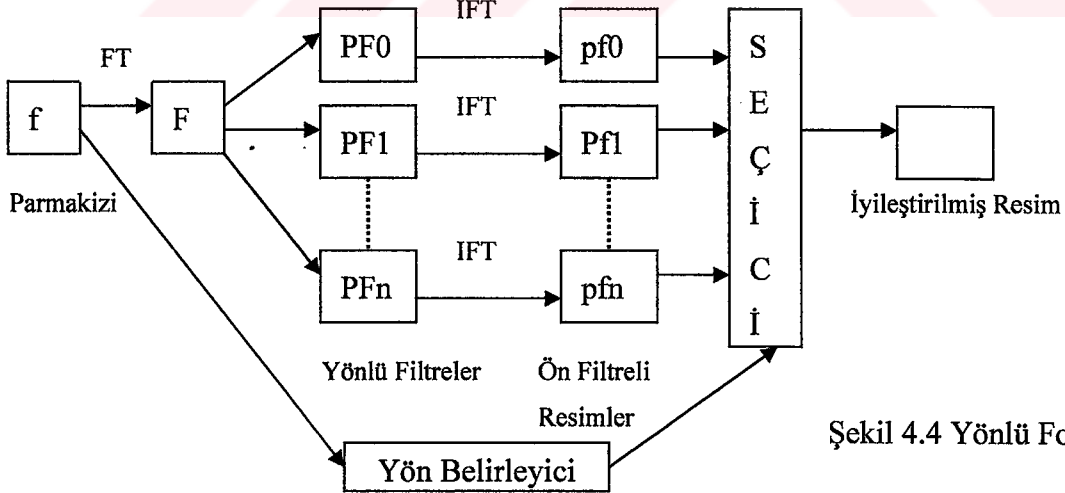
1-D dönüşümden ardıl geçişle 2-D dönüşümün hesaplanması, ters alınırken önceki tekniğin kullanılması durumunda sık görülen hataların kaynağıdır. Başka bir deyişle 2-D ters için, 1-D

algoritması kullanıldığında, herbir satır ve sütunda ilerledikten sonra Kompleks konjugate hesaplanmaz.

4.5 Yönlü Fourier Filtresi (Directional Fourier Transform)

Parmakizi resimlerinin iyileştirilmesinde yön hesabının çok önemli olduğu daha önceki çalışmalar ile kanıtlanmıştır. (Mehtre vd., 1987, 1989) Bu çalışmalardan yola çıkarak Sherlock ve arkadaşlarının çalışmaları ile parmakizi resimleri için Yönlü Fourier Filtresi (YFF) geliştirilmiştir. (Sherlock, 1994).

Parmakizi resimleri, yerel sağrı aralığı (local ridge spacing) ve yerel sağrı yönü veya yönelimi (local ridge orientation) ile tanımlanan bir yapıdadır. Yerel sağrı yönü resmin o noktasındaki doğrultusunu göstermektedir, aralığı ise sağrılar (ridge) arası mesafeyi göstermektedir. Resmin Fourier dönüşümü alınınca resimdeki her noktanın civarındaki komşular tek bir frekans bileşeni içerir. İşte bu bileşenin Fourier merkezine olan uzaklığı yerel sağrı aralığını, açısıl durumu da yerel sağrı yönelimini verir. Bu yöntemde yerel sağrı aralığına ve yönelimine bağlı olarak n adet filtre tanımlanır ve her bir filtre birbirinden bağımsız olarak Fourier dönüşümünden geçmiş resime uygulanır. Bunun sonucunda ters Fourier dönüşümü uygulanarak n adet önfiltreli resim elde edilmiş olur. Daha sonra bu önfiltreli resimlerden yerel sağrı yönelimine göre ilgili parçalar seçilerek iyileştirilmiş resim oluşturulur. Şekil 4.4 'de Yönlü Fourier Filtresinin işleyişini göstermektedir.



Şekil 4.4 Yönlü Fourier Filtresi

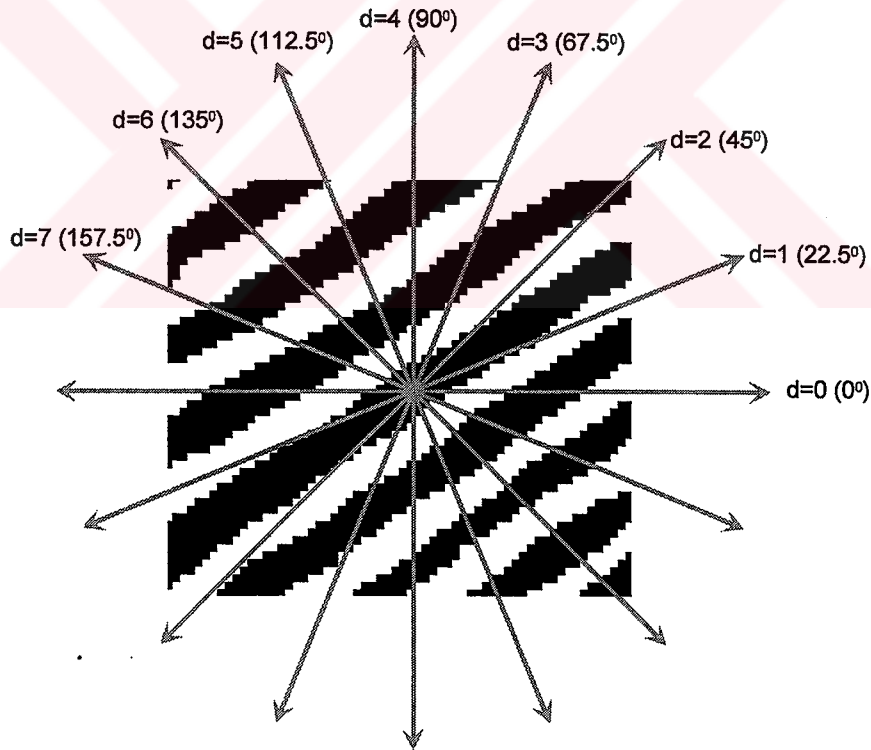
4.5.1 Yerel Sağrı Yöneliminin Hesaplanması

Yerel sağrı yöneliminin hesaplanması parmakizi resimlerinde filtrelemede, sınıflandırmada ve parçalara ayırma gibi işlemlerde kullanılır. Değişik yollarla hesaplamalar yapılmaktadır. Bu çalışmada Mehtre vd. (1987) tarafından geliştirilen yöntem kullanılmıştır. Yönlü resim (Directional Image) aşağıdaki formülün hesaplanması ile bulunur.

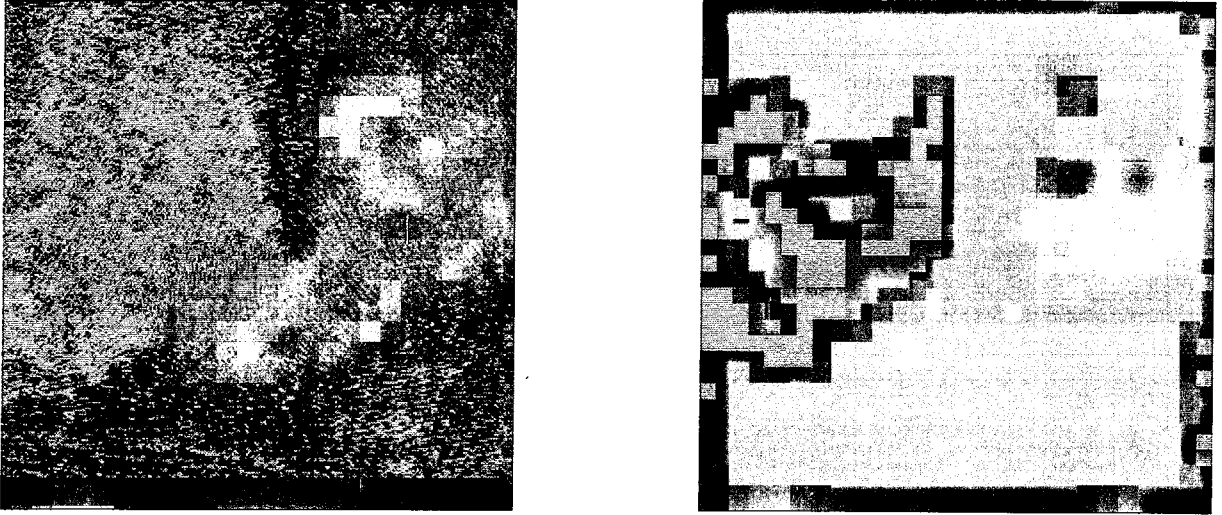
$$K(i, j) = \text{Min} \sum_{k=1}^n |C(i, j) - C_d(i_k, j_k)| \quad d = 0, 1, \dots, N-1 \quad (4.48)$$

Burada $K(i, j)$ i, j noktasının yerel sağrı yönelimini, $C(i, j)$ orijinal resimdeki yerel sağrı yönelimi hesaplanacak pikselin değeri, $C_d(i_k, j_k)$ d yönündeki k . pikselin değeri, n her bir d yönündeki incelenecek piksel sayısı ve N de toplam yön sayısını göstermektedir. $K(i, j)$ 'lerin oluşturduğu resime "Piksel tabanlı yönlü resim (pixel-wise directional image)" denir. İşlem sayısını azaltmak ve daha hızlı işlem yapabilmek için piksel tabanlı yönlü resimden blok tabanlı yönlü resme geçilir. Burada piksel tabanlı resim $16*16$, $32*32$ vb. gibi alt bloklara bölünür ve her bir alt bloğun baskın yönü hesaplanır. Bu hesaplamada baskın yön bloğun içinde en çok tekrar eden yöndür. Bir başka deyişle o bloğun histogramının maksimum değeridir.

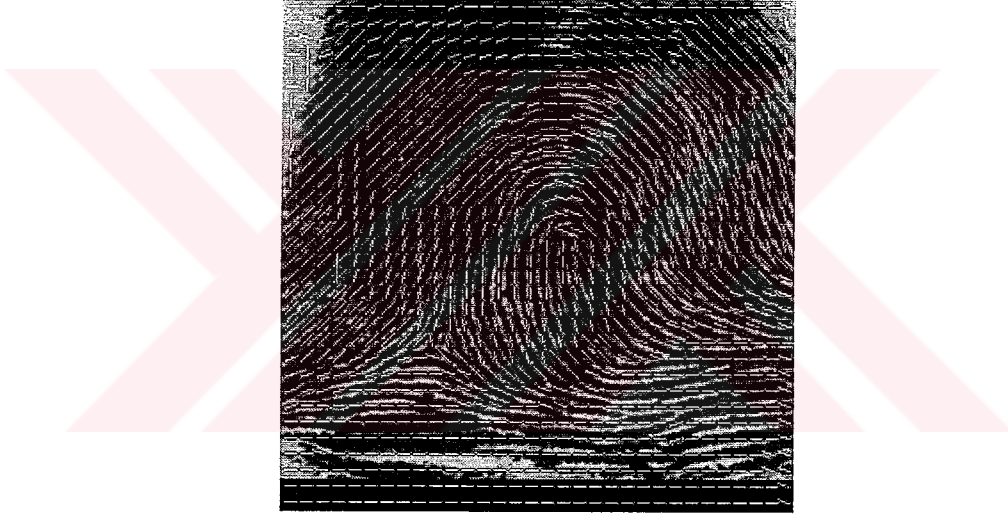
Bu çalışmada $N=8$ yönde ve herbir yönde $n=13$ nokta alınarak piksel ve blok tabanlı yönlü görüntüler oluşturulmuştur.



Şekil 4.5 8 yönlü resmin hesaplanması (Emiroğlu, 1997)



Şekil 4.6 Şekil 2.2 deki resmin piksel ve blok tabanlı yönlü görüntüleri



Şekil 4.7 Şekil 2.2 deki resmin yerel sağrı yönelimleri

4.5.2 Önfilterli Resimlerin Oluşturulması

Önfilterli resimlerin oluşturulması, Fourier dönüşümü uygulanmış resimlere, ButterWorth ve Knutsson filtrelerinin uygulanması ve Ters Fourier dönüşümü uygulanmasıdır. Burada ilk filtre belli bir frekansın altındaki ve üstündeki frekansları yok eden böylece bozulmaları temizleyen bir band-stop filtre, ikincisi ise belli bir yönde olan frekanslara ağırlık veren filtredir. Sherlock (1994) çalışmalarında uzaysal ve yönlü filtreleri bir arada kullanmıştır. Filtre uzaysal ve yönsel filtrelerin çarpımından oluşmaktadır.

$$H(\rho, \phi) = H_{spatial}(\rho) * H_{angle}(\phi) \quad (4.49)$$

$H_{spatial}(\rho)$ Butterworth band pass (band stop) filtre.

$$H_{spatial}(\rho) = \sqrt{\frac{(\rho\rho_{BW})^{2n}}{(\rho\rho_{BW})^{2n} + (\rho^2 - \rho_0^2)^{2n}}} \quad (4.50)$$

Burada $\rho = \sqrt{u^2 + v^2}$ u ve v uzaysal alanın koordinatları. ρ_{BW} band genişliği, ρ_0 merkez frekansı. $H_{angle}(\phi)$ ise

$$H_{angle}(\phi) = \begin{cases} \cos^2 \frac{\pi(\phi - \phi_c)}{2\phi_{BW}} & |\phi - \phi_c| < \phi_{BW} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.51)$$

$\phi = \arctan(u/v)$, ϕ_c $H_{angle}(\phi)$ maksimum olduğu açı yönelimi, ϕ_{BW} açısal band genişliğidir. $\phi_{BW} = \pi/n$ ise eşit aralıklı $\phi_c = i\pi/n$ yönelimine sahip n adet yönlü filtre tanımlanabilir. Bu çalışmada 8 değişik yönde önfiltreli resimler oluşturulmuştur. Şekil 2.2 deki parmakizi resmi için oluşturulan önfiltreli resimler şekil 4.8 'deki gibidir. Bu resimlerin oluşturulması için kullanılan algoritmanın pseudo kodu:

```

ResimFFT := FFT(RESIM)                /* Resmin Fourier Dönüşümü Hesaplanır*/

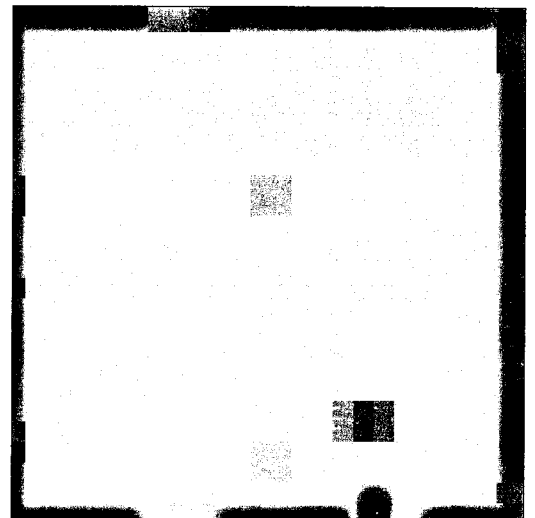
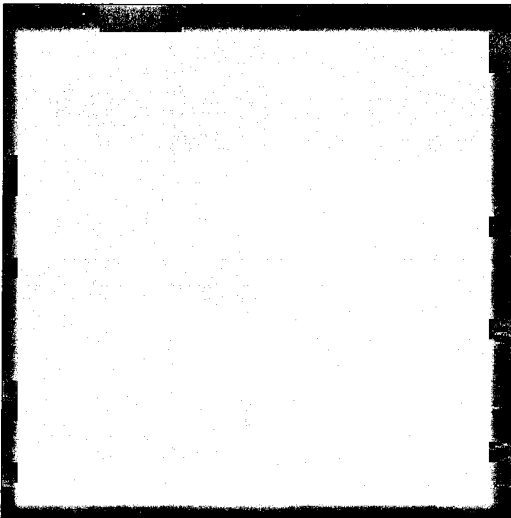
For i := 0 to 7                          /* Her bir yön için*/

TEMP := RESIMFFT * F[i]                 /* F[i] i. Yönlü filtreyi uygula*/

PF[i] := IFFT(TEMP)                     /*Ters Fourier Dönüşümü ile Önfiltreli resmi oluştur*/

Next i

```

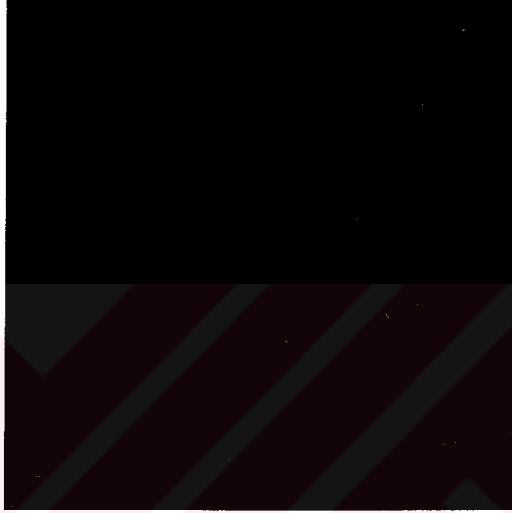




Şekil 4.8 8 yönde oluşturulan önfiltreli resimlerin 256*256 lık görüntüleri

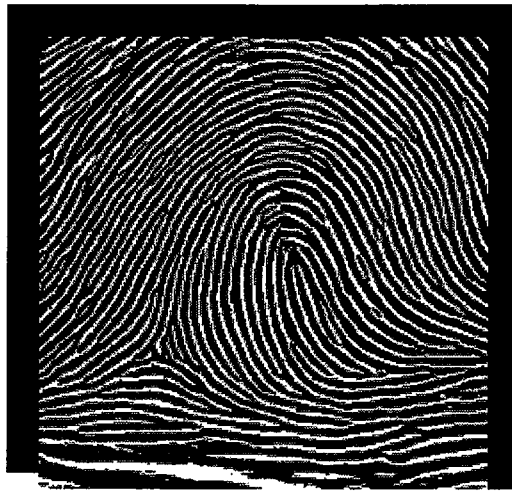
4.5.3 Önfiltreli Resimlerden Uygun Parçaların Seçilmesi

Önfiltreli resimler oluşturulduktan sonra bu resimlerden ilgili parçaların seçilerek iyileştirilmiş resim oluşturulur. Bu parçaların seçilmesi için orijinal resim üzerinde noktaların yönlerinin bulunması gerekmektedir. Bu yönlerin bulunmasında 16*16 lık bölge içerisindeki tüm noktaların aynı yönde olduğu varsayılarak hesaplanmıştır. Bu hesaplamalar 4.5.1 'de anlatılmıştır. Orijinal resimden elde edilen yöne göre önfiltreli resimlerden ilgili blok seçilerek yeni oluşturulacak resme yerleştirilir. Böylelikle iyileştirilmiş resimler bloklar halinde önfiltreli resmin değişik kısımlardan oluşturulmuş olur.



Şekil 4.9 Şekil 2.2 deki resmin YFF ile iyileştirilmiş görüntüsü.

Şekil 4.8 deki resmin RAT yöntemine göre 16x16 lık alt bloklar seçilerek siyah-beyaza çevrilmiş hali ise Şekil 4.9 daki gibidir.



Şekil 4.10 Şekil 4.9 in RAT yöntemi ile siyah-beyaz görüntüsü.

Şekil 4.9, şekil 3.2 ile karşılaştırıldığında YFF nin parmakizi resimleri üzerinde çok iyi sonuç verdiği görülür. Çizgilerin düzgünlüğü (Smooth) ve sürekliliği (Continuous) orijinal resme göre çok daha iyidir.



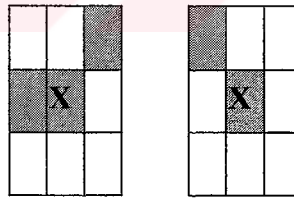
5. İNCELTME (THINNING) ALGORİTMALARI

İnceltme, dijital görüntü işlemede daha çok bir ara işlem olarak kullanılır. Parmakizi analizleri, el yazısı tanımlama, imza doğrulama ve biomedikal sistemlerde inceltme işlemi yoğun olarak kullanılmaktadır (Kwok,1988).

Literatürde inceltmenin tanımı üzerinde kesin bir anlaşma yoktur. Farklı tanımlar yapılmış, algoritmalar kurulmuş ve değişik bazen de beklenmedik sonuçlar elde edilmiştir (Zhang ve Wang, 1992). İnceltme de amaç resmin iskeletini çıkarmak ve resimde herhangi bir kayıp olmadan tek piksel genişliğine düşürmektir. İnceltme algoritmaları Rosenfeld'e (Kwok, 1988) göre paralel ve seri olmak üzere ikiye ayrılır. Paralel algoritmlar tüm piksellere simültane olarak uygulanır ve sonuç elde edilir. Bu sonuç olarak elde edilen resim tekrardan iterasyona sokulur. Seri algoritmalar ise resim taranırken piksel kendisinden önceki pikselin iterasyon sonunda aldığı değer üzerinden işleme sokulur. Bu çalışmada siyah-beyaz resimler için paralel inceltme algoritmaları esas alınmıştır.

İnceltme algoritmaları:

- 1) Resimleri inceltirken resmin sürekliliğini korumalı ve kaldırılması süreksizliğe, kesilmelere sebep olacak sınır noktaları kaldırılmamalıdır.
- 2) Resmin inceltirken gerekli yerleri kaldırmamalı ve resmi kısaltmamalıdır (Pitas,1993).



Şekil 5.1 Örnek durumlar

Şekil 5.1 'de soldaki resimde X işaretli pikselin kaldırılması süreksizliğe sebep olur. Sağdaki X işaretli pikselin kaldırılması ise resmi kısaltır. Yukarıda bahsettiğimiz bu iki şartın sağlanmasını algoritma anlamında inceleyecek olursak, resmin pikselleri şekil 5.2 deki gibi etiketlenir ve 3x3 lük bir alt komşuluğu ile birlikte incelenmeye alınır.

p ₉	p ₂	p ₃
p ₈	p ₁	p ₄
p ₇	p ₆	p ₅

Şekil 5.2 İncelenen piksel p_1 ve 3x3 komşuluğu

Resmin sürekliliğinin bozulmaması için p_1 etrafındaki siyah piksellerin kaç tane olduğu önemlidir. Toplam siyah piksel sayısını $S(p_1)$ ile gösterelim. Bu Formül 5.1 ile hesaplanabilir. Eğer $S(p_1) \leq 2$ ise p_1 merkezli alt blok resmin herhangi bir sonunu, dalını veya uzantısını gösteriyor demektir. Bu piksel kaldırılması da resmi kısaltır.

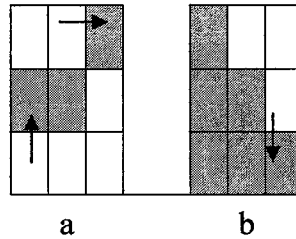
$$S(p_1) = \sum_{i=2}^9 p_i = p_2 + p_3 + \dots + p_9 \quad (5.1)$$

Eğer $S(p_1) > 7$ ise p_1 merkezli alt blok resmin bir sınır noktası değilde iç nokta olduğunu gösterir. Çünkü $S(p_1) > 7$ olması p_1 in çevresinde hiç beyaz pikselin olmamasına karşı gelir, bu da onun resmin sınırında olmadığını belirtir.

$T(p_1)$ sayısı da $p_2, p_3, \dots, p_9, p_2$ sırası ile p_1 'in etrafındaki piksellerdeki beyazdan (piksel değeri=0) siyaha (piksel değeri=1) geçiş sayısını versin. Bu Formül 5.2 ile hesaplanabilir. Eğer $T(p_1) = 1$ ise p_1 in 3x3 komşuluğunda sadece bir tane bağlı bileşen olduğunu ve p_1 'rin kaldırılması, bağlılığı yani sürekliliği bozmaz.

$$T(p_1) = \sum_{i=2}^9 \bar{p}_i p_{i+1}, \quad p_{10} = p_2 \text{ ve } \bar{p}_i = 1 - p_i \quad (5.2)$$

$S(p_1)$ ve $T(p_1)$ sayısı ile ilgili örnekler şekil 5.3 'deki gibidir.



Şekil 5.3 (a) $S(p_1) = 2, T(p_1) = 2$ (b) $S(p_1) = 5, T(p_1) = 1$

Bu iki şartın sağlanması ile oluşan inceltme algoritmasına “One Pass Thinning” algoritması denir (Pitas,1993).

5.1 Zhang ve Suen Algoritması (ZS)

Bu algoritma resimden iskeletine ait olmayan sınır noktalarını kaldırarak inceltmeye çalışır. (Zhang ve Suen,1984) Her iterasyon iki alt iterasyona bölünmüştür. Birinci alt iterasyon

Çizelge 5.1 'deki gibidir. Bu şartlar sağlandığında p_1 silinebilir.

Çizelge 5.1 1. Alt iterasyon şartları

a-)	$2 \leq S(p_1) \leq 6$
b-)	$T(p_1) = 1$
c1-)	$p_2 \times p_4 \times p_6 = 0$
d1-)	$p_4 \times p_6 \times p_8 = 0$

İkinci alt iterasyonda Çizelge 5.2'deki gibidir. Bu şartlar sağlandığında p_1 silinebilir.

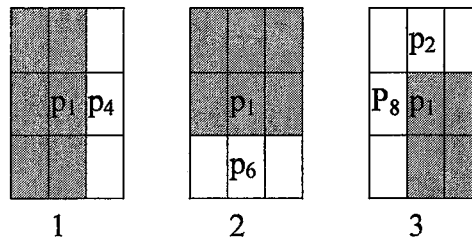
Çizelge 5.2 2. Alt iterasyon şartları

a-)	$2 \leq S(p_1) \leq 6$
b-)	$T(p_1) = 1$
c2-)	$p_2 \times p_4 \times p_8 = 0$
d2-)	$p_2 \times p_6 \times p_8 = 0$

Bu iterasyonlar hiç bir pikselin değeri değişmeyene kadar devam eder.

$$\left. \begin{array}{l} p_2 \times p_4 \times p_6 = 0 \\ p_4 \times p_6 \times p_8 = 0 \end{array} \right\} \text{ Bu iki şart } p_4=0 \text{ veya } p_6=0 \text{ veya } (p_2=0 \text{ ve } p_8=0) \text{ olduğunda sağlanır.}$$

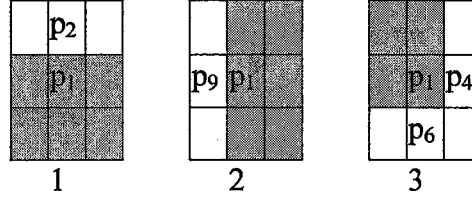
Bu durumlara örnekler Şekil 5.4 'de verilmiştir.



Şekil 5.4 Birinci alt iterasyon 1.Doğu sınır, 2.Güney sınır, 3.Kuzey-Batı Köşe

$$\left. \begin{array}{l} p_2 \times p_4 \times p_8 = 0 \\ p_2 \times p_6 \times p_8 = 0 \end{array} \right\} \text{Bu iki şart } p_2=0 \text{ veya } p_8=0 \text{ veya } (p_4=0 \text{ ve } p_6=0) \text{ olduğunda sağlanır.}$$

Bu durumlara örnekler Şekil 5.5 'de verilmiştir.

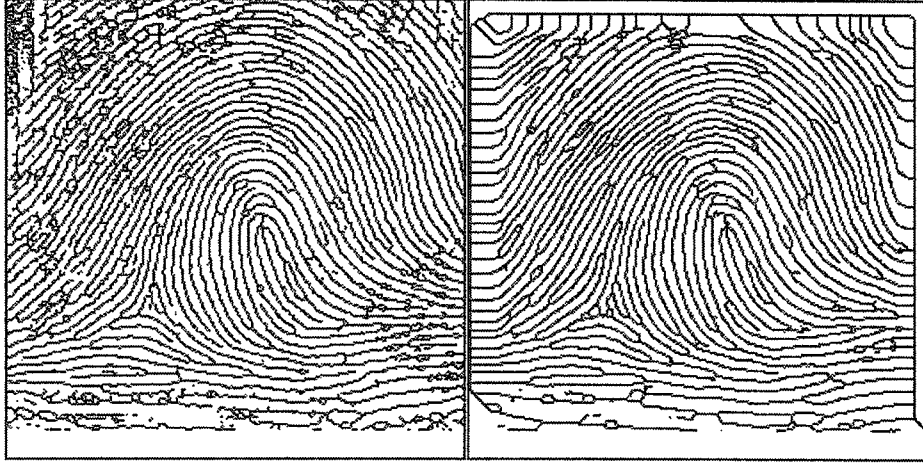


Şekil 5.5 İkinci alt iterasyon 1.Kuzey Sınır,2.Batı sınır,3.Güney-Doğu sınır.

Yukarıdaki şekillerden anlaşılacağı gibi birinci alt iterasyonda silinen p_1 pikseli doğu veya güney sınır noktası yada kuzey-batı köşe noktası olabilir. Benzer şekilde ikinci iterasyonda silinen p_1 pikseli kuzey veya batı sınır noktası ya da güney-doğu köşe noktası olabilir.

One Pass Thinning algoritmasında, eğer resim soldan sağa olarak işleme sokuluyorsa, önce kuzey-ve batı noktaları kaldırılıyor ve incelmış resim orijinal resmin güney ve doğu sınırında oluşuyor. Bu da algoritmanın simetrik çalışmadığını yani resmi simetrik olarak inceltmediğini gösteriyor. İşte ZS algoritmasındaki c_1, d_1, c_2 ve d_2 şartları ile bu sorun ortadan kaldırılmaya çalışılmıştır. Bu algoritmaya bu yüzden “İki geçişli inceltme” algoritması da denir (Pitas, 1993).

Bu algoritma, özellikle parmak izi görüntülerine uygulandığı zaman, bazı durumlarda eğik çizgileri sildiği ve düzgün inceltmediği görülmüştür. İki piksel genişliğindeki diyagonal çizgiler bir iki nokta haline dönüşmektedir. Bu inceltme algoritması, parmak izleri üzerinde bazı kaldırmaması gereken noktaları kaldırdığını ve bu yüzden resmin sürekliliğini bazı bölgelerde bozduğunu ve resmi tek piksel genişliğine indiremediği ortaya çıkarılmıştır. ZS Algoritması'nın test resminin üzerinde uygulanmış sonuçları aşağıdaki şekilde görülmektedir.



Şekil 5.6 Test resminin ZS Algoritması ile inceltmiş hallerinin 256*256 'lık görüntüleri

Soldaki parmakizi resmi orijinal resmin threshold edildikten sonra inceltmeye gönderilmiş halini, sağdaki ise orijinal resmin Yönlü Fourier Filtresi'nden geçirilip daha sonra sırasıyla threshold ve inceltmiş halini göstermektedir.

5.2 Lu ve Wang Algoritması (LW)

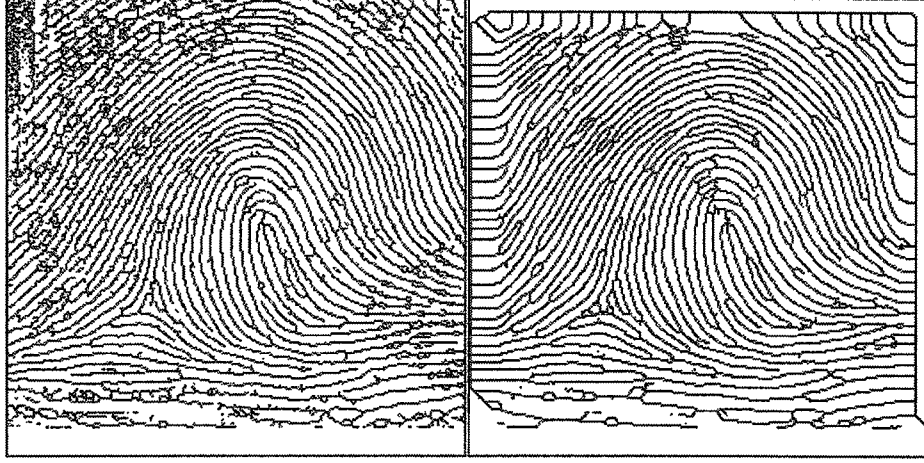
ZS algoritmasının eğik çizgilere olan problemi gidermek için Lu ve Wang, ZS Algoritması'nın birinci şartını $2 \leq S(p_1) \leq 6$ dan $3 \leq S(p_1) \leq 6$ olarak değiştirmişlerdir.

Çizelge 5.3 LW Algoritması'nın Şartları

a-)	$3 \leq S(p_1) \leq 6$
b-)	$T(p_1) = 1$
c1-)	$p_2 \times p_4 \times p_6 = 0$
d1-)	$p_4 \times p_6 \times p_8 = 0$
c2-)	$p_2 \times p_4 \times p_8 = 0$
d2-)	$p_2 \times p_6 \times p_8 = 0$

Bu algoritma ile elde edilen sonuçlar ZS Algoritması ile karşılaştırıldığında eğrilerin devamlılığı ve düzgünlüğü ZS Algoritması'na göre daha iyi olduğu gözlenmektedir. Şartın değişmesi ile LW Algoritması ZS'ye göre daha az noktayı kaldırdığı, çıkan sonuçlardan da gözlenmektedir. Fakat bu eğik çizgilerin iki piksel genişliğinde olması yine istenen bir sonuç

değildir. Çünkü inceltmenin amacı resmin iskeletini bir piksel genişliğinde elde etmektir. Bu algoritmanın test resmimiz üzerindeki sonuçları Şekil 5.7 'deki gibidir.



Şekil 5.7 Test resminin LW Algoritması ile inceltilmiş hallerinin 256*256 'lık görüntüleri.

5.3 Rutovitz Algoritması (RD)

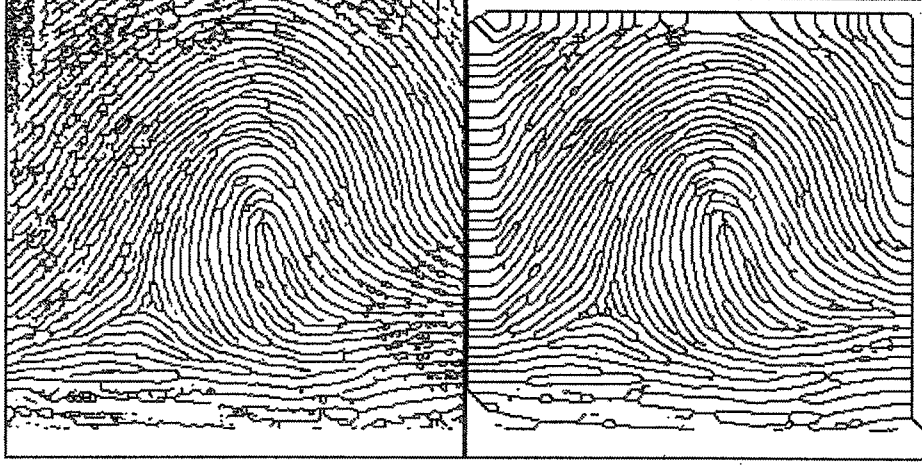
Bu algoritmadaki piksel kaldırma (silme) şartları Çizelge 5.4 'deki gibidir. Bu şartları sağlayan p_1 pikseli silinir.(Jain, 1989; Zhang ve Wang, 1992)

Çizelge 5.4 RD Algoritması'nın şartları

a-)	$3 \leq S(p_1) \leq 6$
b-)	$T(p_1) = 1$
c-)	$p_2 \times p_4 \times p_8 = 0$ veya $T(p_2) \neq 1$
d-)	$p_2 \times p_4 \times p_6 = 0$ veya $T(p_4) \neq 1$

Yukarıda verilen şartları sağlayan p_1 pikseli silinir.

Bu algoritma ile elde edilen sonuçlar da çizgiler arasında gereksiz uzantıların kaldığı gözlenmiştir. Resimdeki çizgilerin devamlılığı ve sürekliliği daha önce incelediğimiz iki algoritmaya göre daha iyidir. Sonuçlar Şekil 5.8 'de gösterilmiştir.



Şekil 5.8 RD Algoritması ile inceltilmiş test resminin 256*256 lık görüntümleri.

5.4 Zhang ve Wang Algoritması (ZW)

Bu algoritma için pikseller aşağıdaki şekilde etiketlenir.

	p ₁₁		
p ₉	p ₂	p ₃	
p ₈	p ₁	p ₄	p ₁₅
p ₇	p ₆	p ₅	

Şekil 5.9 ZW algoritması için piksellerin etiketlenmesi

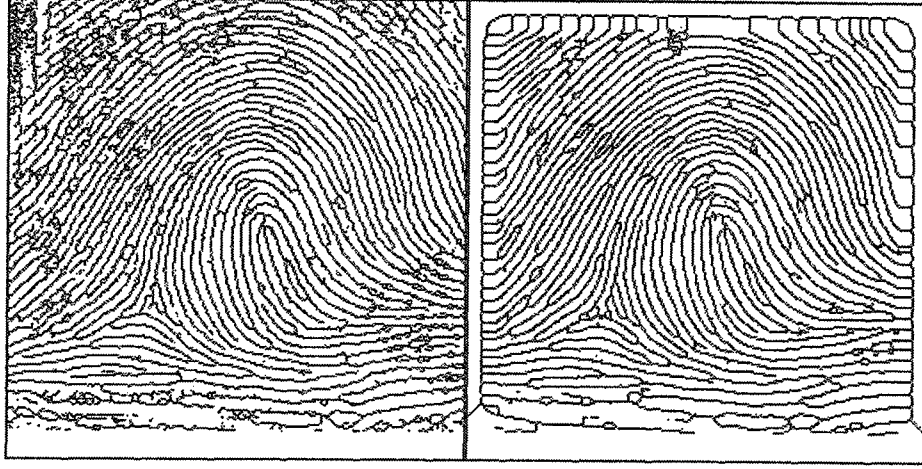
p₁'in silinebilmesi için Çizelge 5.5'deki şartları sağlaması gereklidir. (Zhang ve Wang, 1992)

Çizelge 5.5 ZW Algoritması'nın şartları

a-)	$2 \leq S(p_1) \leq 6$
b-)	$T(p_1) = 1$
c-)	$p_2 \times p_4 \times p_8 = 0 \text{ or } p_{11} = 1$
d-)	$p_2 \times p_4 \times p_6 = 0 \text{ or } p_{15} = 1$

ZW Algoritması ile elde edilen sonuçlar incelenen diğer üç algoritmanın sonuçlarına göre

daha iyidir. Çizgilerin sürekliliği nispeten daha iyidir. Tek piksel genişliğine en çok yaklaşan sonuçlar bu algoritma ile elde edilmiştir ama yine de tüm resmi tek piksel genişliğine düşürememiştir. Sonuçlar Şekil 5.10 'da görülebilir.



Şekil 5.10 ZW Algoritması ile inceltilmiş test resminin 256*256 'lık görünümü

İncelenen dört algoritmada diğer uygulamalarda iyi sonuçlar vermesine rağmen parmakizi resimlerinin inceltmesinde istenilen sonuçları verememiştir. Bundan sonraki bölümde parmakizi resimlerinin tek piksel genişliğine düşürülmesi için iki orijinal algoritma verilecektir.

6. TEK PİKSELE İNDİRGEME ALGORİTMALARI

6.1 Tek Piksel Genişliğine Düşürmek için İnceltme Sonrası Önerilen Algoritmalar

Bu bölümde anlatılacak olan algoritmalar yukarıda anlatılan inceltme algoritmaları ile inceltmiş resimlere uygulanmaktadır. Bu algoritmalar inceltme algoritmalarının aksine seri işlem yapmaktadır. Seri algoritmalar, resim taranırken incelenen piksel kendisinden önceki pikselin iterasyon sonunda aldığı değer üzerinden işleme sokulur. Bu algoritmalar işlem yaparken, resimin piksellerini sol alt köşeden sağ üst köşeye doğru taramaktadır. Bu algoritmalar için piksel etiketleme şeması Şekil 6.1 'deki gibidir. Algoritmalarındaki şartlar esnek bir yapıya sahiptir; yani resmin taranma yönüne göre şartları değiştirmek mümkündür. Algoritmalar incelenen inceltme algoritmalarından geçmiş ama tek piksel genişliğine düşürülememiş resimler üzerinde uygulanmaktadır. İncelenen dört inceltme algoritması da parmakizi resimlerini tek piksel genişliğine düşürememiştir. Burada anlatılacak olan algoritmalarından geçen resimler hangi inceltme algoritmasından sonra uygulandı ise tek piksele indirgenmiş resimler de o resim ile aynı yapıya sahiptir. Çünkü algoritmalar incelenen inceltme algoritmaları için hep aynı şekilde piksel kaldırmaktadır. Bu yüzden bu algoritmalar inceltme algoritmalarından kalan bazı gereksiz pikselleri kaldırmadığı görülür.

p ₂	p ₃	p ₄
	p ₁	p ₆
		p ₅

Şekil 6.1 Yeni algoritmalar için piksel etiketleme şeması

I. Algoritma için piksel silme şartları Çizelge 6.1 'deki gibidir.

Çizelge 6.1 I. Algoritma için piksel silme şartları

a-)	$p_1 = 1$
b-)	$p_2 + p_4 \geq 1$

Çizelge 4.6 'daki şartlar sağlandığında p₃ ile etiketlenen piksel silinebilir. Resmin tamamında bu şartları sağlayan tüm siyah p₃ pikselleri silinir. Bu algoritmanın çalışması sonucunda tek piksel genişliğine indirilmiş resim elde edilir.

II. Algoritma için piksel silme şartları Çizelge 6.2 'deki gibidir.

Çizelge 6.2 II. Algoritma için piksel silme şartları

a-)	$p_1 = 1$
b-)	$p_4 + p_5 \geq 1$

Yukarıda verilen şartlar sağlandığında p_6 ile etiketlenen piksel silinebilir. Bu algoritmanın da inceltme algoritmalarından sonra resme uygulanması halinde tek piksel genişliğine indirilmiş resim elde edilmektedir. Sonuçlar detaylı olarak bir sonraki bölümde görülebilir. Resmin taranma yönüne göre piksel kaldırma koşulları Çizelge 6.3 'de verilmiştir. Bu çizelgedeki şartlar için piksel etiketleme şeması ise Şekil 5.2 'deki gibidir.

Çizelge 6.3 Resmin taranma yönlerine göre piksel silme şartları

	Sol Alt-Sağ Üst	Sağ Alt-Sol Üst	Sol Üst-Sağ Alt	Sağ Üst-Sol Alt
I. Algoritma	$p_1 = 1, p_3 + p_9 \geq 1$ p_2 silinir.	$p_1 = 1, p_3 + p_9 \geq 1$ p_2 silinir.	$p_1 = 1, p_5 + p_7 \geq 1$ p_6 silinir.	$p_1 = 1, p_5 + p_7 \geq 1$ p_6 silinir.
II. Algoritma	$p_1 = 1, p_3 + p_5 \geq 1$ p_4 silinir.	$p_1 = 1, p_7 + p_9 \geq 1$ p_8 silinir.	$p_1 = 1, p_3 + p_5 \geq 1$ p_4 silinir.	$p_1 = 1, p_7 + p_9 \geq 1$ p_8 silinir.

7. SONUÇLAR

- ◆ Bu tezde önerilen iki orjinal algoritma, literatürde var olan dört inceltme [ZS, LW, RD, ZW] algoritmalarından sonra ortaya çıkarılmıştır. Her iki algoritmada resimleri tek piksel genişliğine indirmeyi başarmıştır.
- ◆ Çalışmalar sırasında on farklı parmakizinin hem orijinal hemde iyileştirilmiş versiyonları kullanılmıştır ve algoritmalarımız herbirinin inceltmesinde tek piksel genişliğini yakalamayı başarmıştır.
- ◆ Bu algoritmalar ile incelediğimiz inceltme algoritmalarının parmakizi resimlerini düzgün bir şekilde inceltmesini sağlamış olmaktadır.

Çizelge 7.1 'de test resminin orijinal ve iyileştirilmiş hallerinin inceltilmiş resimlerindeki toplam siyah piksel sayıları görülmektedir. Çizelge 7.2 ve 7.3 'de I. ve II. Algoritma'nın uygulanması ile resimlerden silinen siyah piksel sayılarını göstermektedir.

Çizelge 7.1 İnceltmiş resimlerdeki siyah piksel sayısı

	ZS Algoritması	LW Algoritması	RD Algoritması	ZW Algoritması
Orijinal Resim	31044	31279	30171	32398
İyileştirilmiş resim	27073	26544	25405	27260

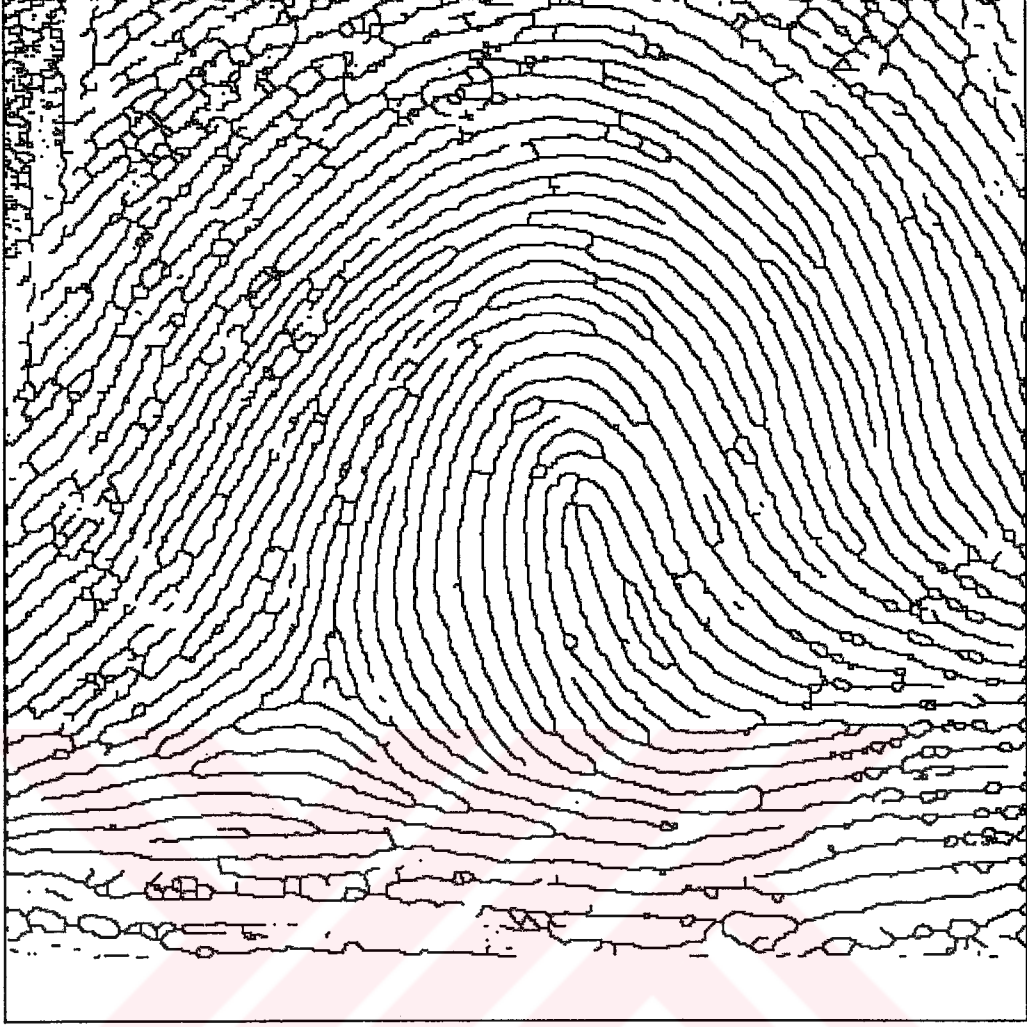
Çizelge 7.2 I.Algoritmanın uygulanması ile resimden silinen piksel sayıları

	ZS Algoritması	LW Algoritması	RD Algoritması	ZW Algoritması
Orijinal Resim	11470	11385	12375	11961
İyileştirilmiş resim	9365	9224	9431	9432

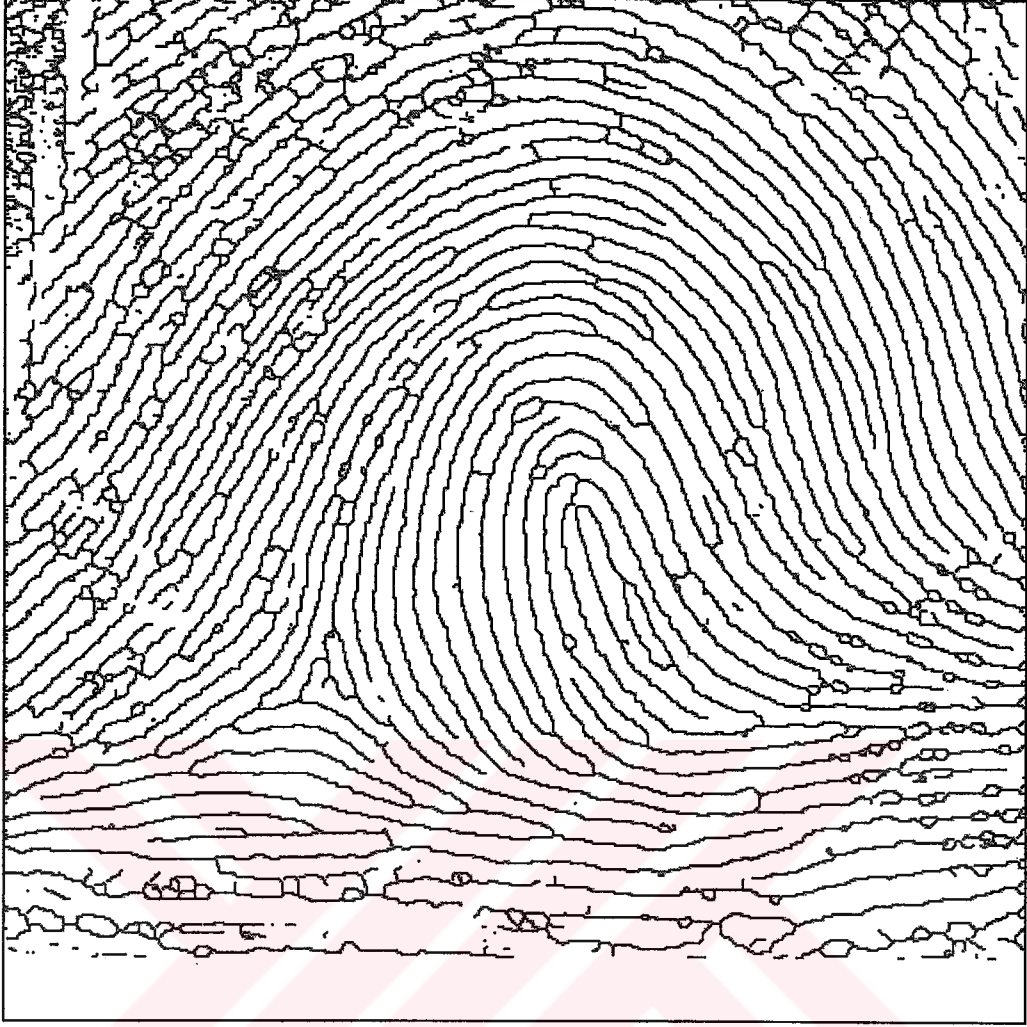
Çizelge 7.3 II. Algoritmanın uygulanması ile resimden silinen piksel sayıları

	ZS Algoritması	LW Algoritması	RD Algoritması	ZW Algoritması
Orijinal Resim	11646	11607	12516	12133
İyileştirilmiş resim	9706	9221	9743	9776

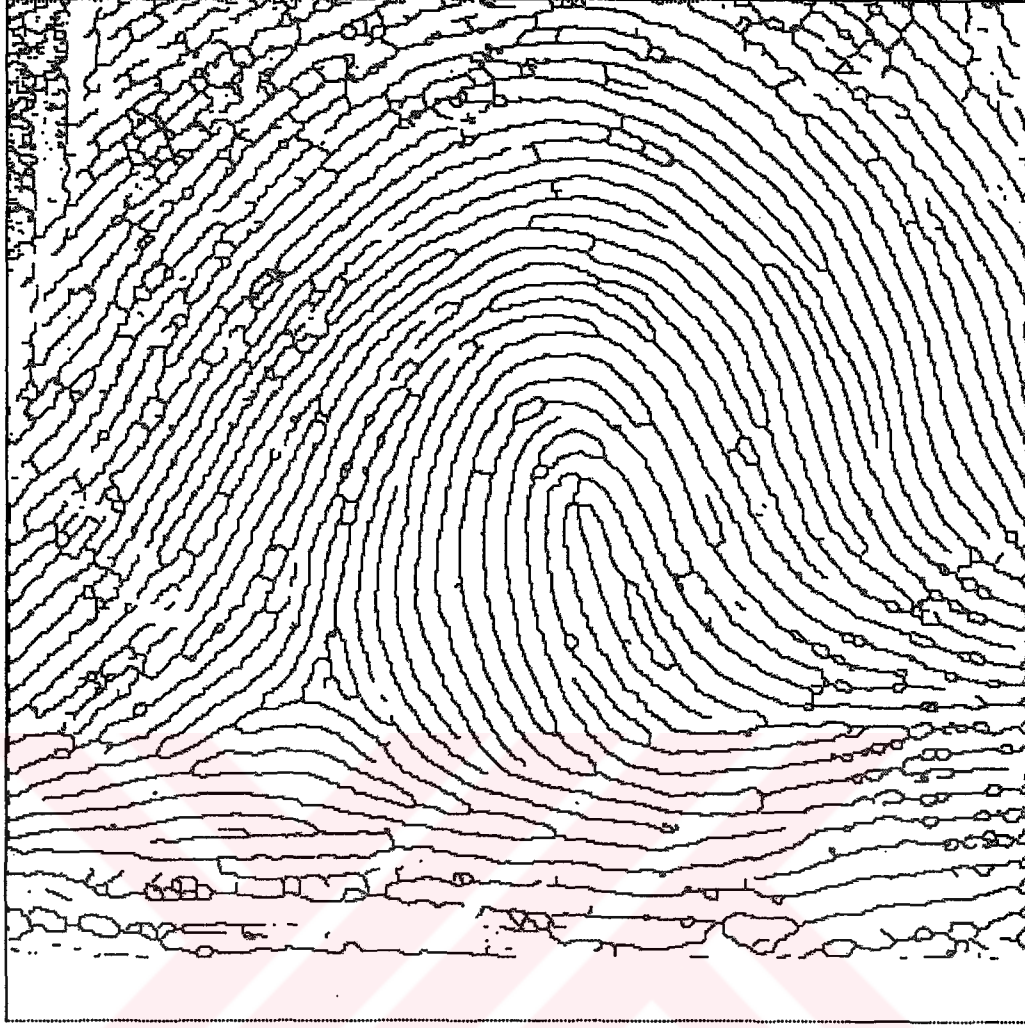
Çizelgeler incelendiğinde algoritmaların en az sayıda pikseli LW Algoritması ile inceltmiş resimden sildiği sonucuna varılabilir. Buradan LW Algoritması ile inceltmiş resimlerin diğer üç inceltme algoritmasına göre tek piksel genişliğine daha yakın olduğu yorumu yapılabilir. Ama algoritmanın çok piksel kaldırmış olması düzgün bir inceltme yaptığı anlamına gelmemektedir. Çünkü kaldırılan pikseller kaldırılmaması gereken yani süreksizliğe yol açan pikseller veya resmi kısaltan, resmin iskeletine ait olan pikseller olabilir. Nitekim ZW Algoritması daha az nokta kaldırmış olmasına rağmen resimler incelendiğinde çizgilerin daha sürekli ve çizgiler arasındaki gereksiz uzantıların daha az olduğu görülecektir. Ayrıca I. Ve II. Algoritmanın sildiği pikseller arasındaki fark en fazla 344 pikseldir. Tek piksel genişliğine indirildikten sonra resimdeki siyah piksel sayısı çizelgeler incelendiğinde 15662 ile 20437 arasında değiştiği hesaplanabilir. Yaklaşık 15000 ile 20000 tane siyah pikselin olduğu resimlerde 344 pikselin değerinin değişmesinin gözle farkedilmesi zordur. Yine de bu farkın resimlerin üzerinde bir etkisinin olacağını düşünmekteyiz. Önerdiğimiz algoritmalarının bazı durumlarda resimler üzerinde süreksizliğe sebep olduğunu ve inceltme algoritmalarından kalan bazı gereksiz pikselleri silmediğini yaptığımız çalışmalardan gördük. Bu süreksizliklerin parmakizi resimlerinin özellikli noktalarının bulunmasına nasıl etkisinin olacağını araştırılması ilerki çalışmalar olarak düşünülmektedir.



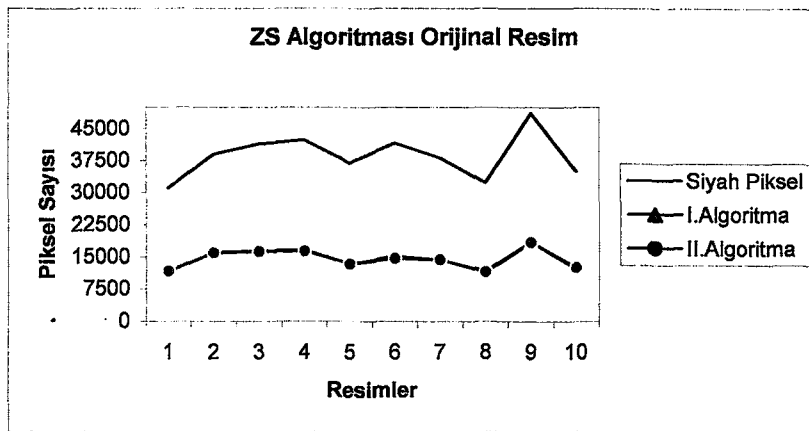
Şekil 7.1 ZS Algoritması ile inceltilmiş orijinal parmakizi resmi



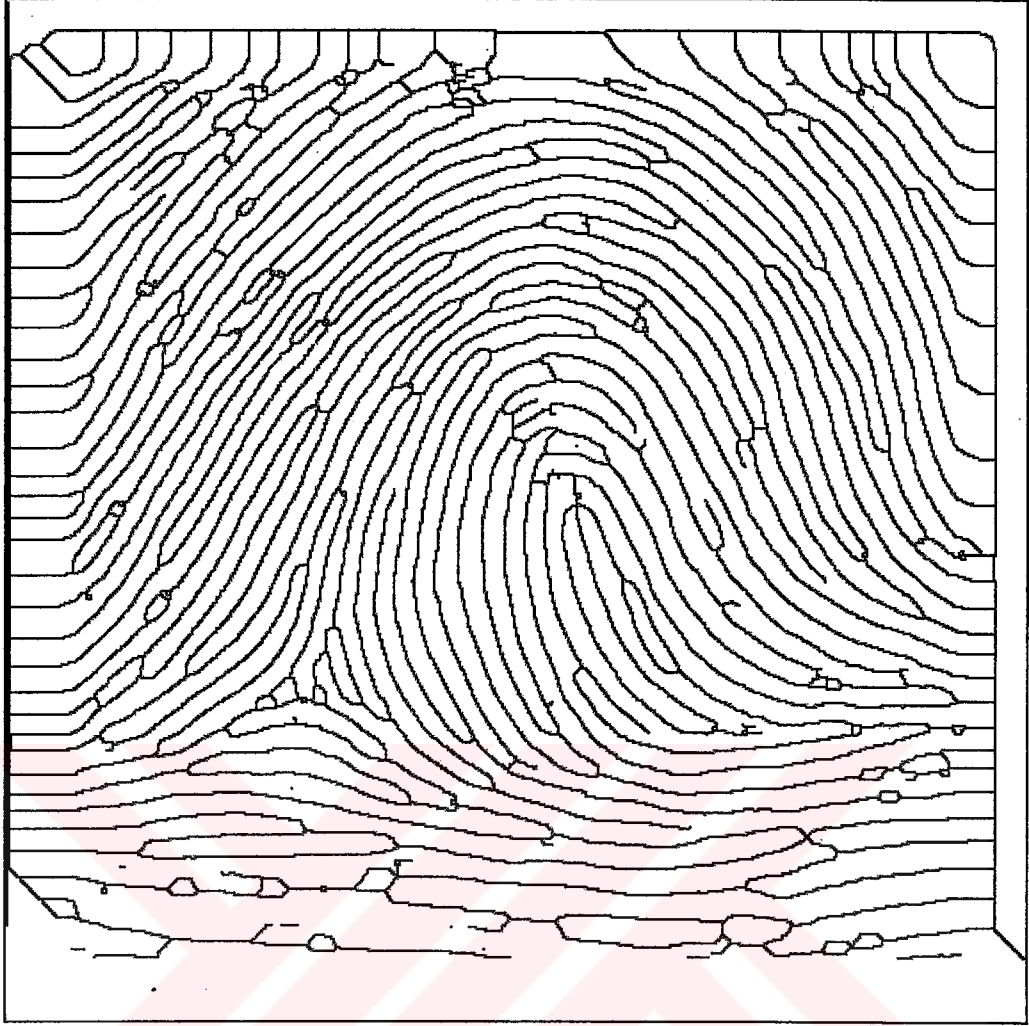
Şekil 7.2 Şekil 7.1 'deki inceltilmiş resmin I. Algoritma'dan geçirilmiş hali.



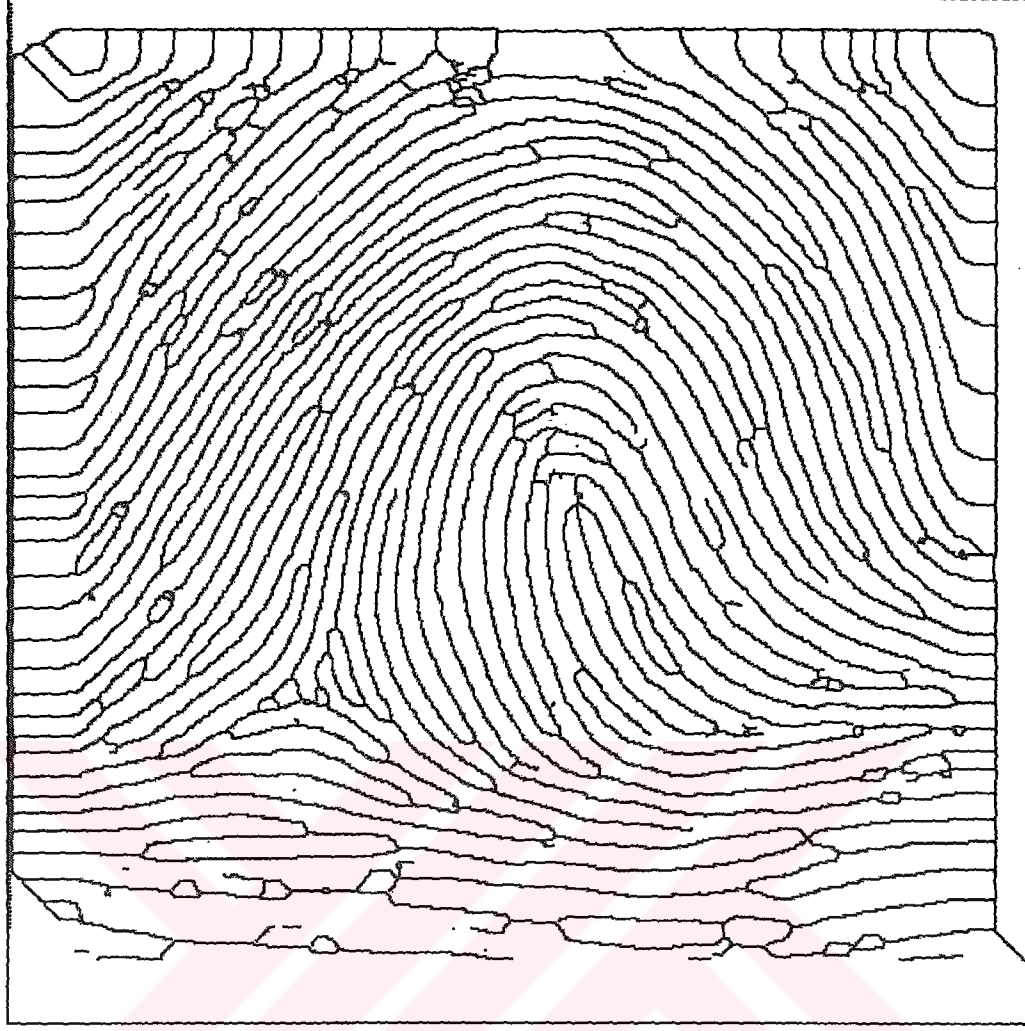
Şekil 7.3 Şekil 7.1 'deki resmin II. Algoritma'dan geçirilmiş hali.



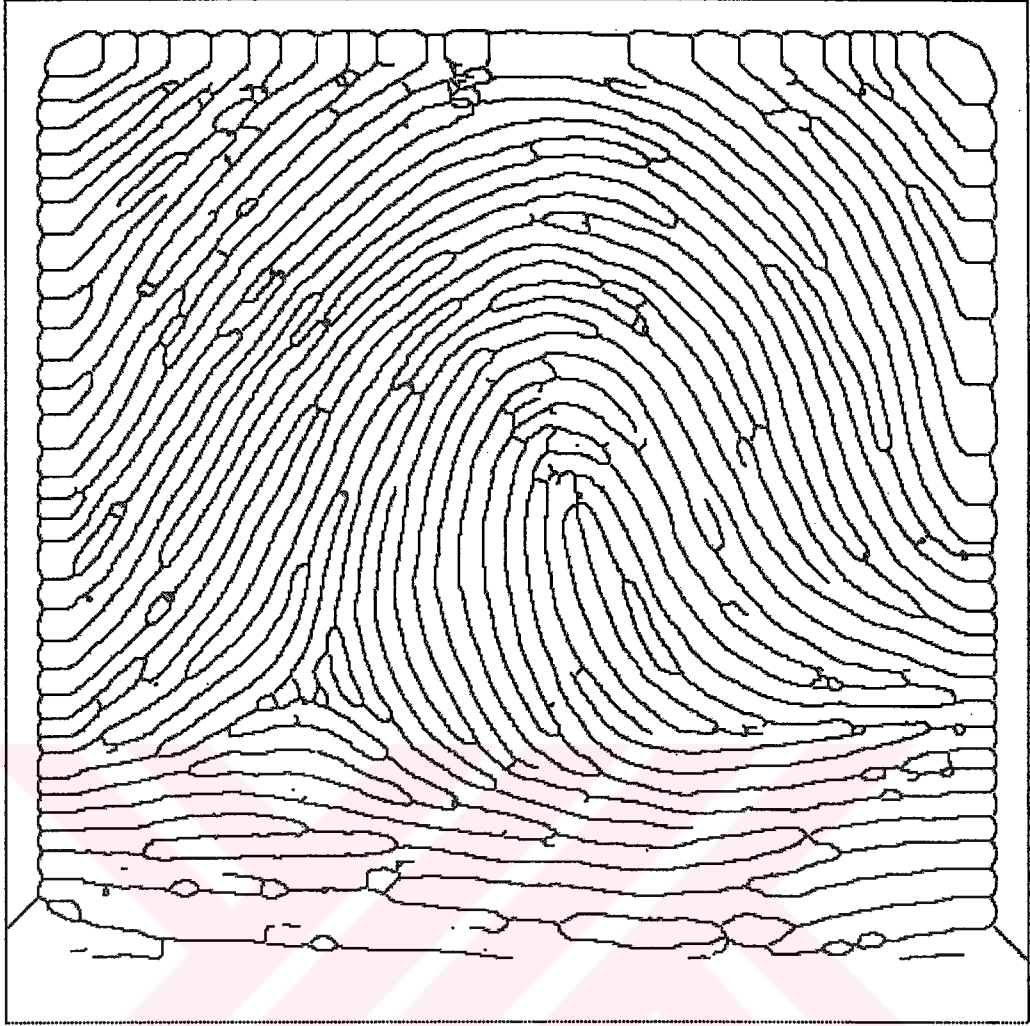
Şekil 7.4 İncelenen on farklı resim için ZS Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları



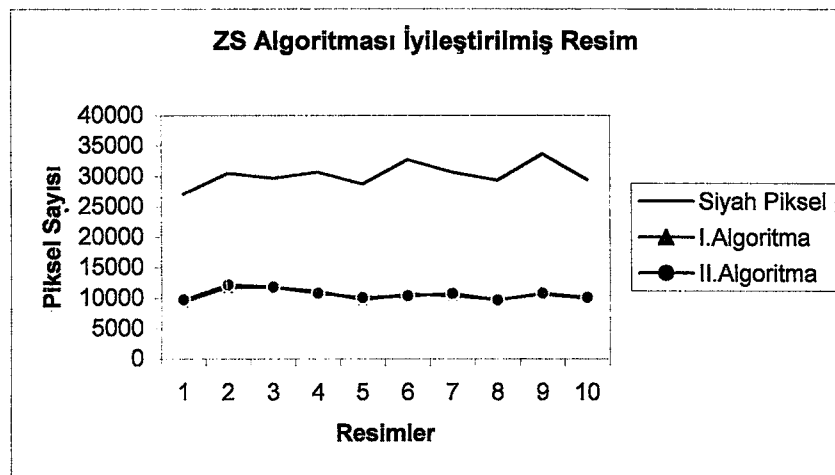
Şekil 7.5 Yönlü Fourier Filtresi ile iyileştirilmiş test resminin ZS Alg. ile inceltilmiş hali



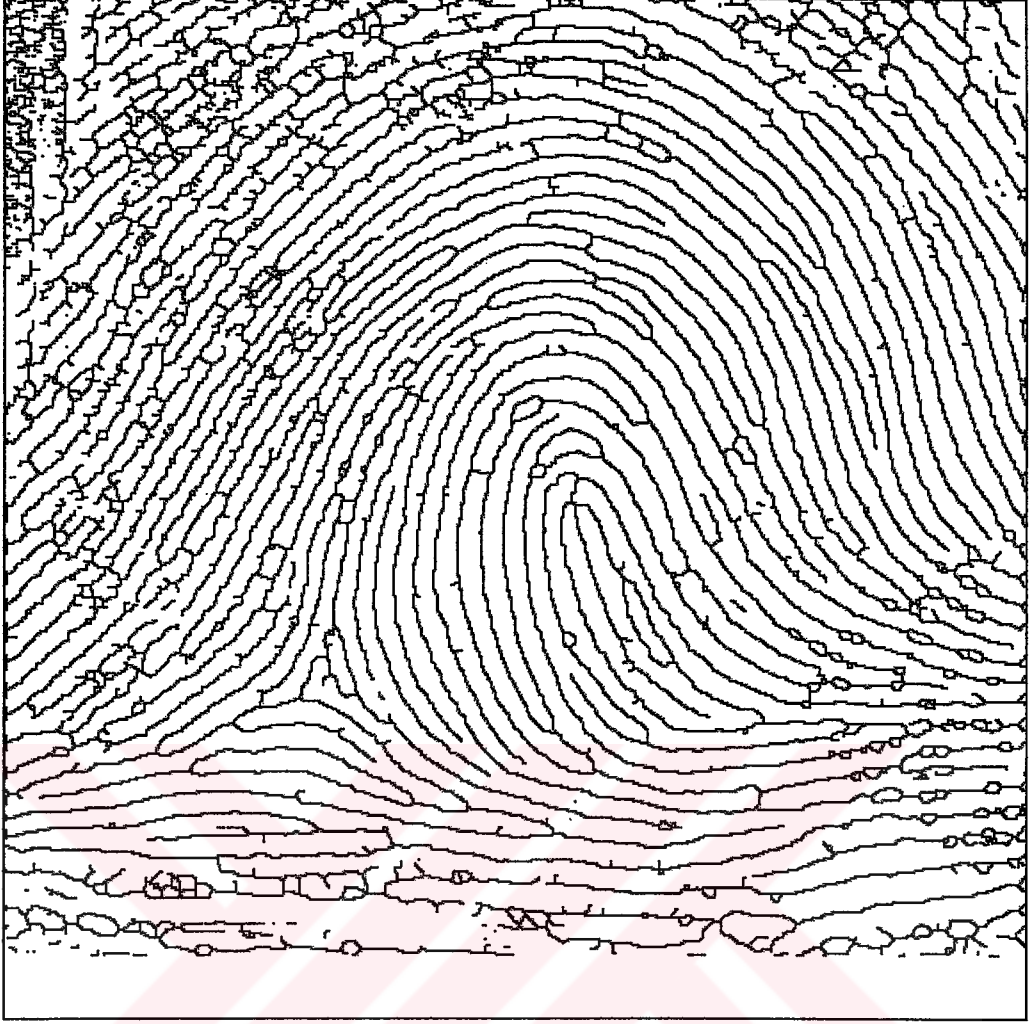
Şekil 7.6 Şekil 7.5 'in I. Algoritma'dan geçirilmiş hali



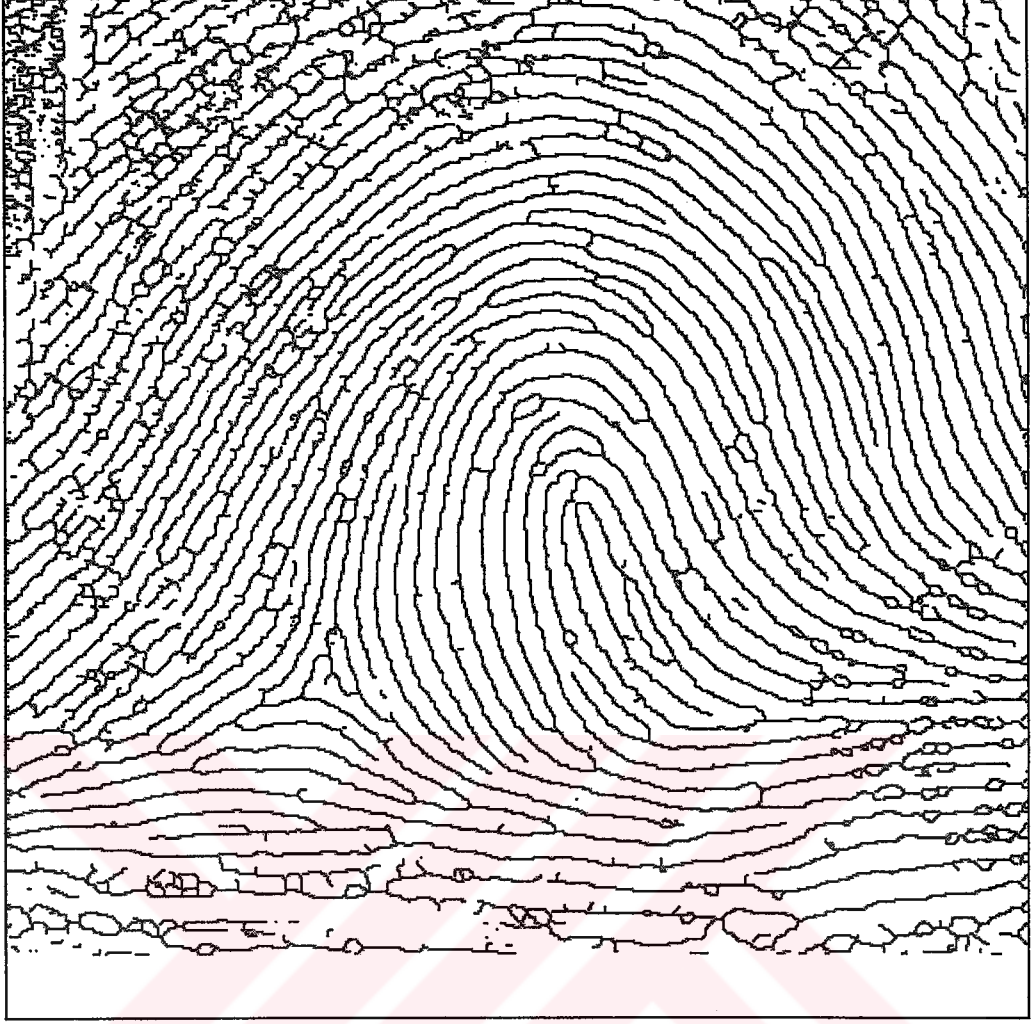
Şekil 7.7 Şekil 7.5 'in II. Algoritma'dan geçirilmiş hali



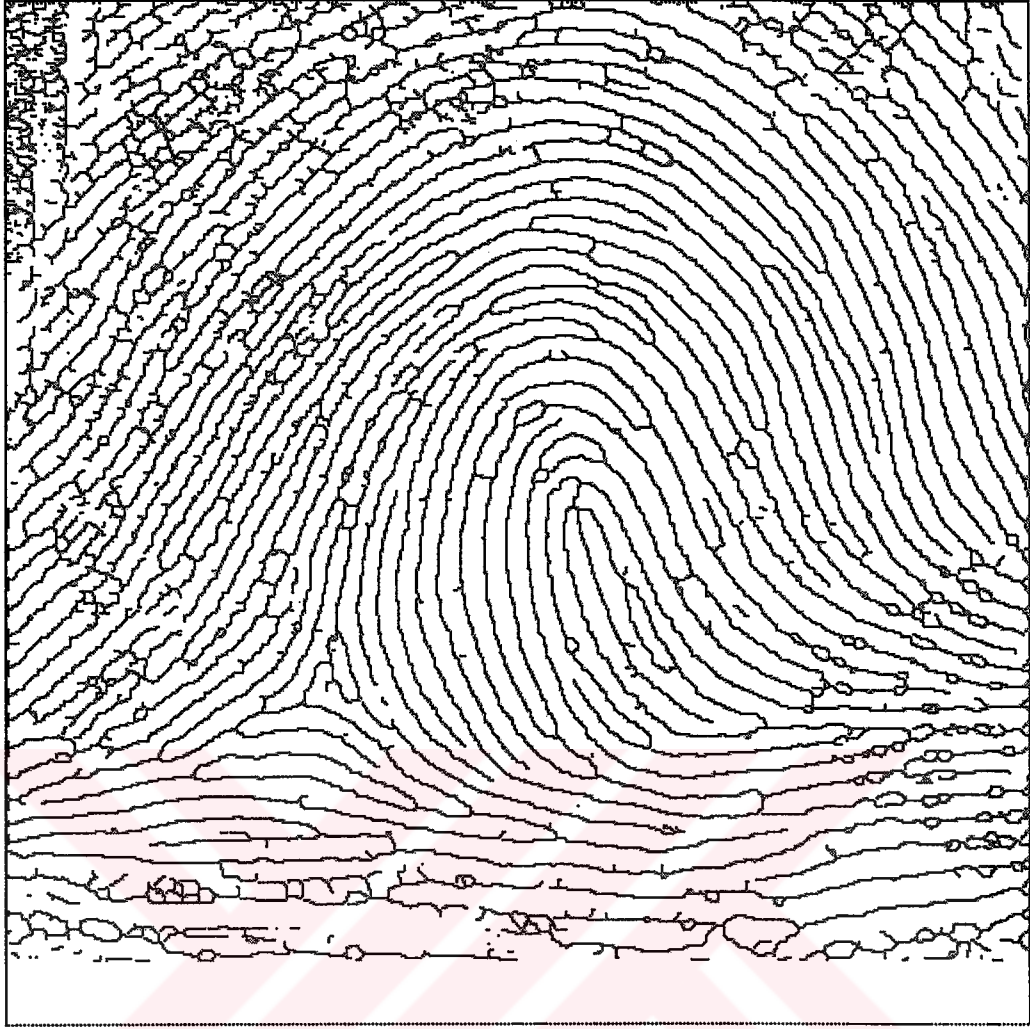
Şekil 7.8 İncelenen on farklı resmin iyileştirilmiş versiyonu için ZS Algoritması ile incelenen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları



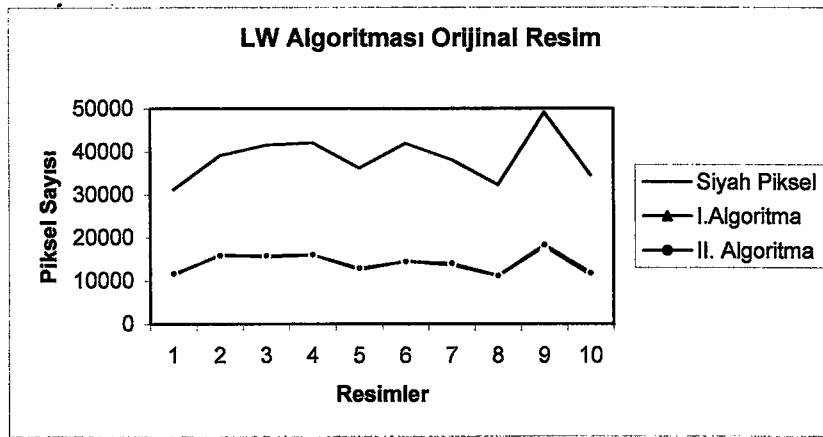
Şekil 7.9 Test resminin LW Algoritması ile inceltilmiş hali



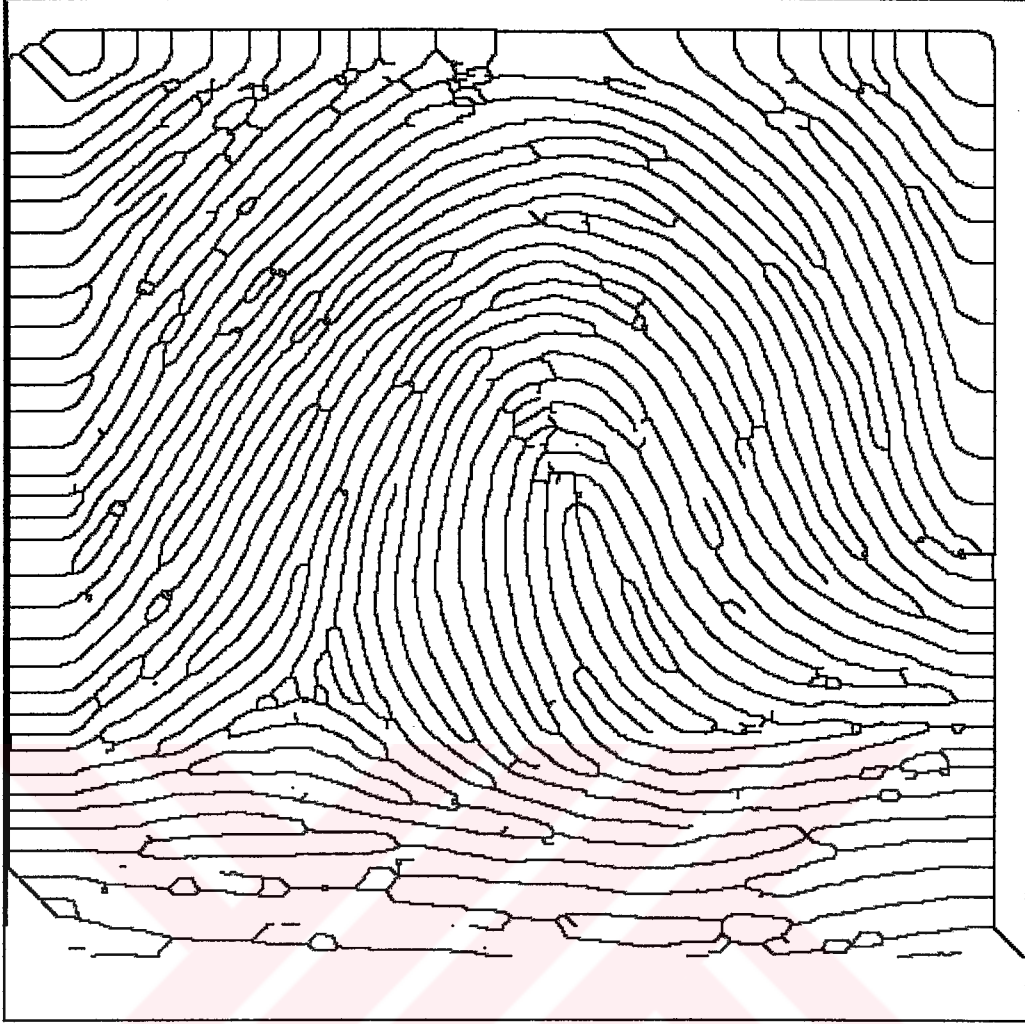
Şekil 7.10 Şekil 7.9 'nin I. Algoritma'dan geçirilmiş hali.



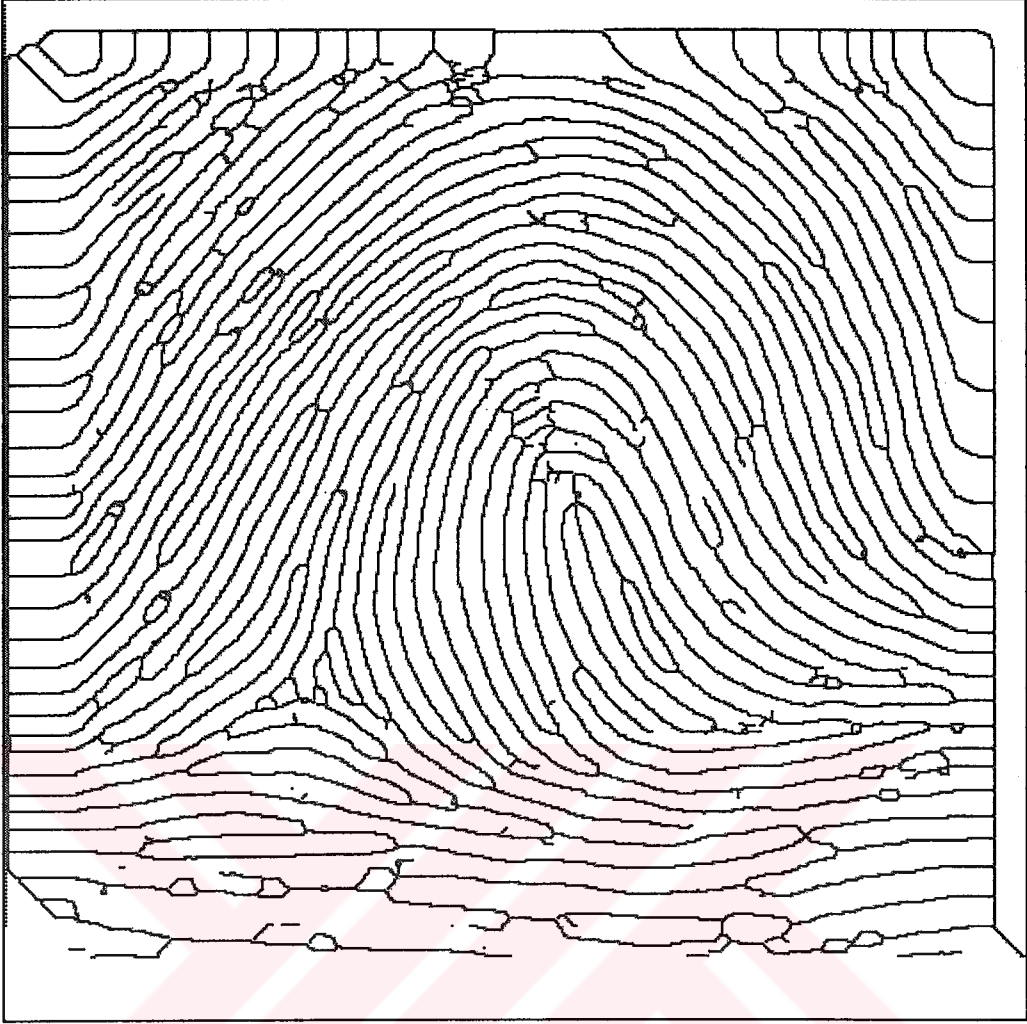
Şekil 7.11 Şekil 7.9 'zun II. Algoritma'dan geçirilmiş hali.



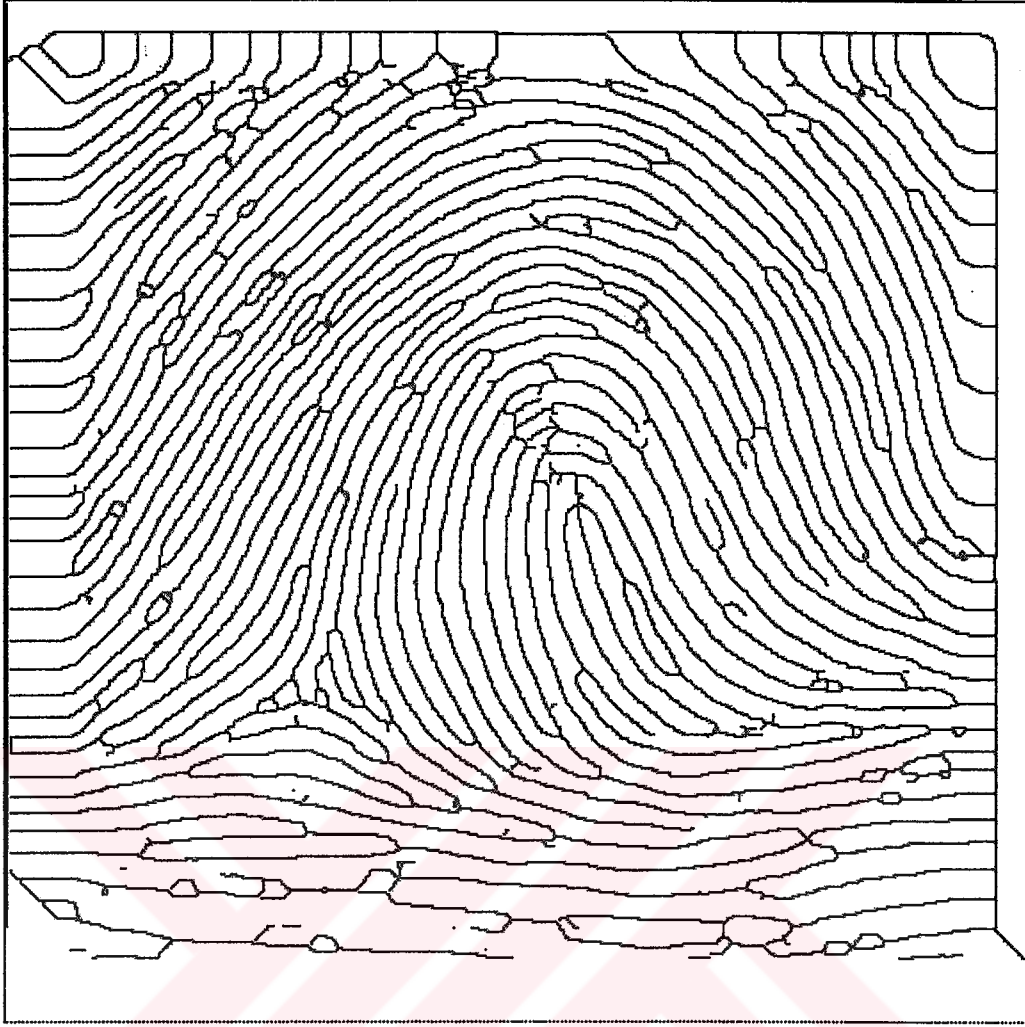
Şekil 7.12 İncelenen on farklı resim için LW Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları



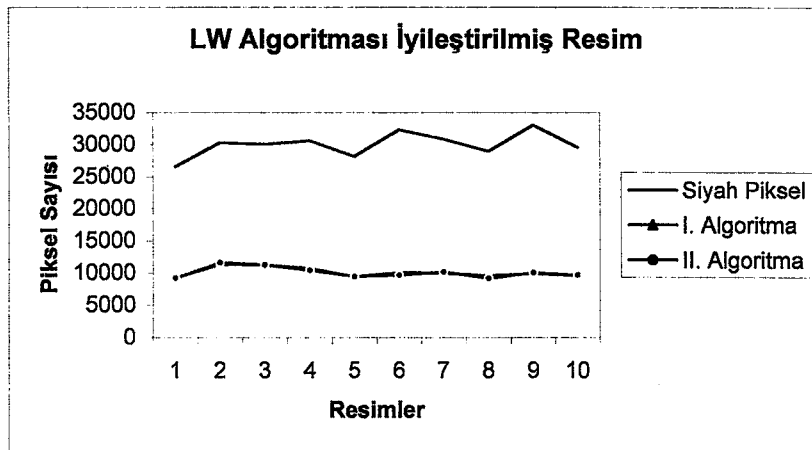
Şekil 7.13 YFF'den geçirilmiş test resminin LW Algoritması ile inceltilmiş hali.



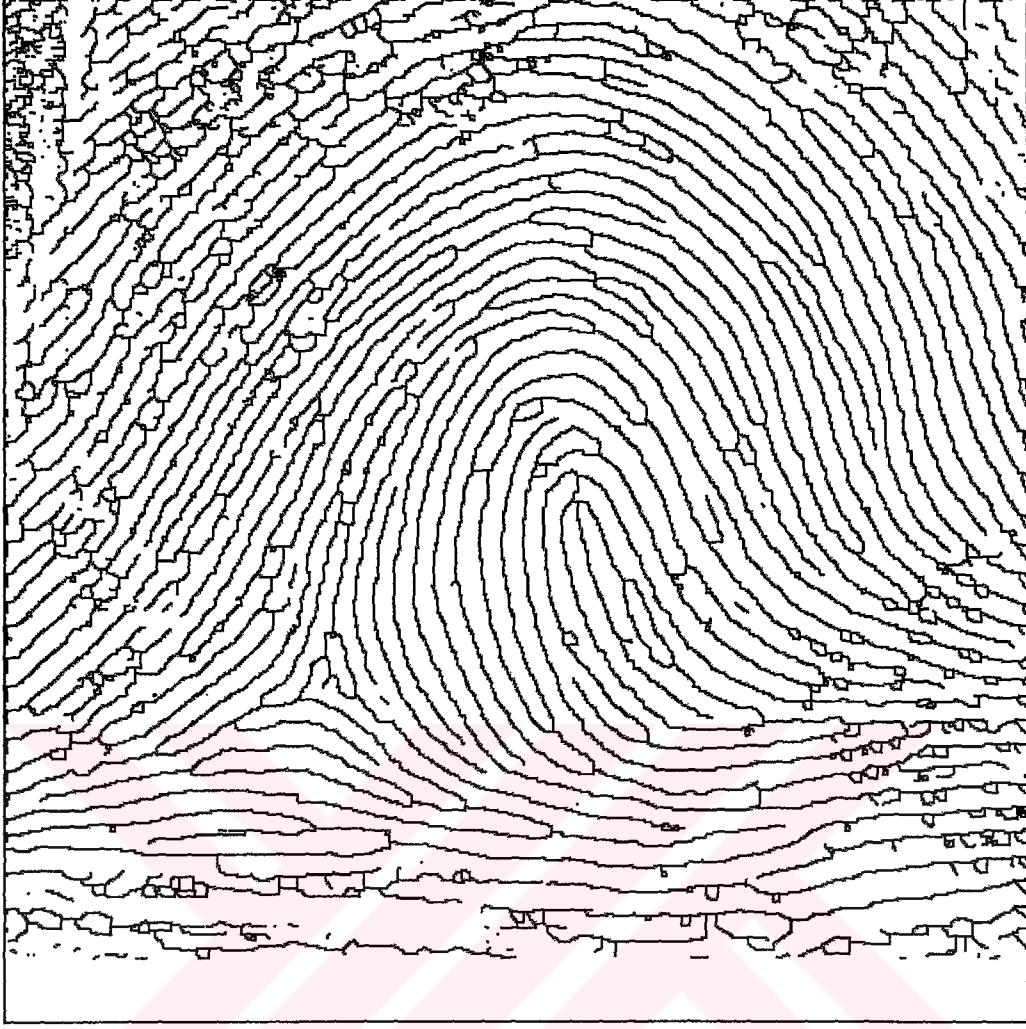
Şekil 7.14 Şekil 7.13 'ün I.Algoritma'dan geçirilmiş hali



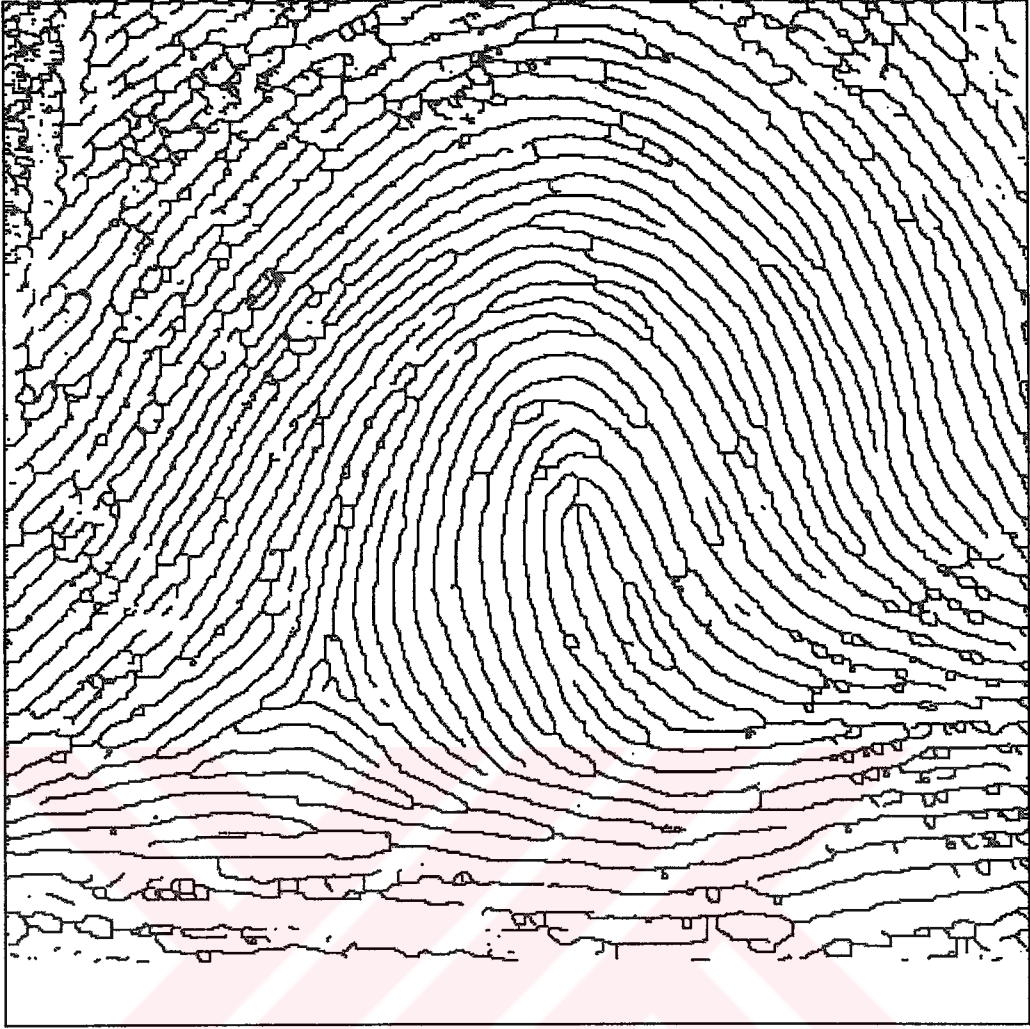
Şekil 7.15 Şekil 7.13 'ün II. Algoritma'dan geçirilmiş hali.



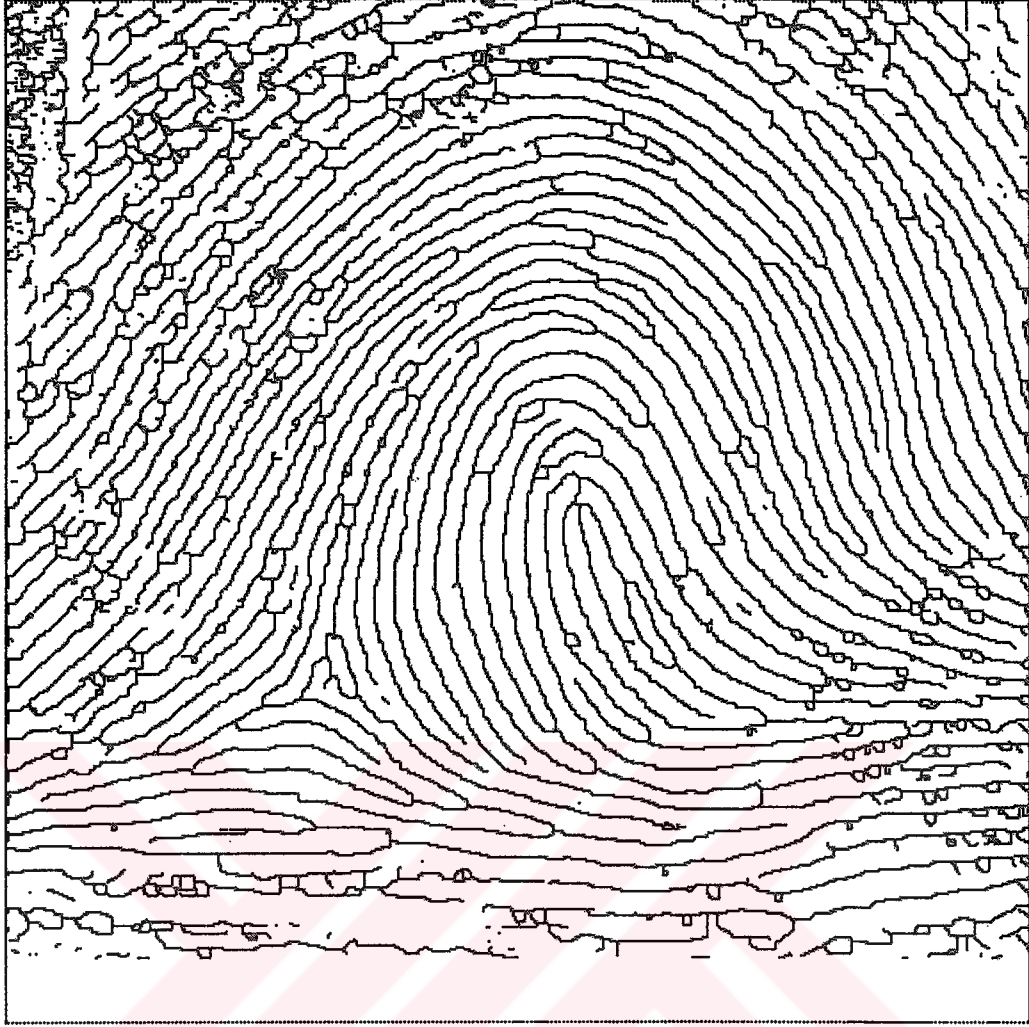
Şekil 7.16 İncelenen on farklı resmin iyileştirilmiş versiyonu için LW Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları



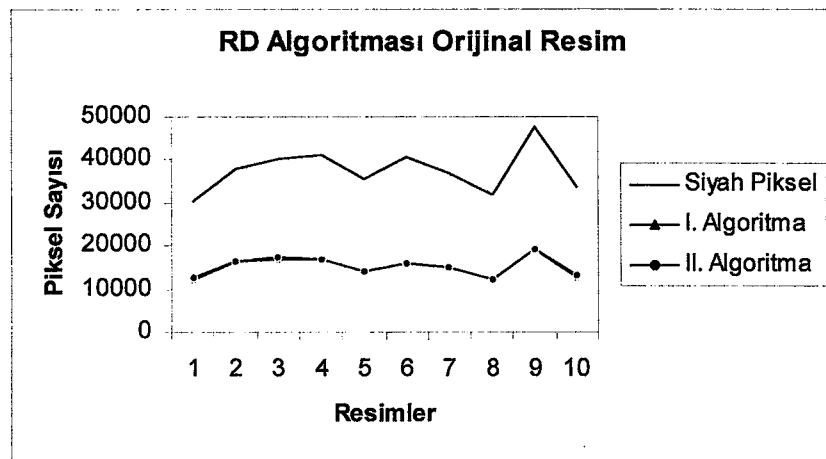
Şekil 7.17 Test resminin RD Algoritması ile inceltilmiş hali.



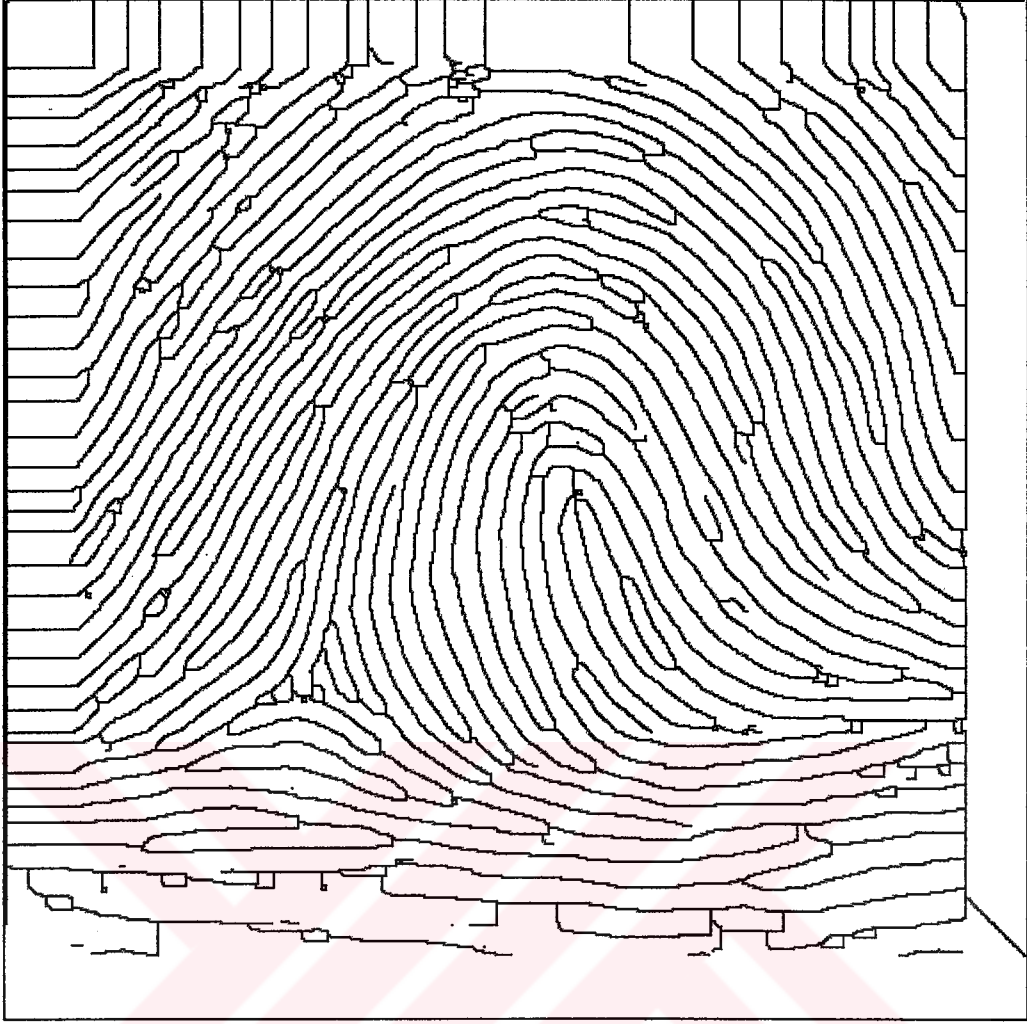
Şekil 7.18 Şekil 7.17 'nin I.Algoritma'dan geçirilmiş hali



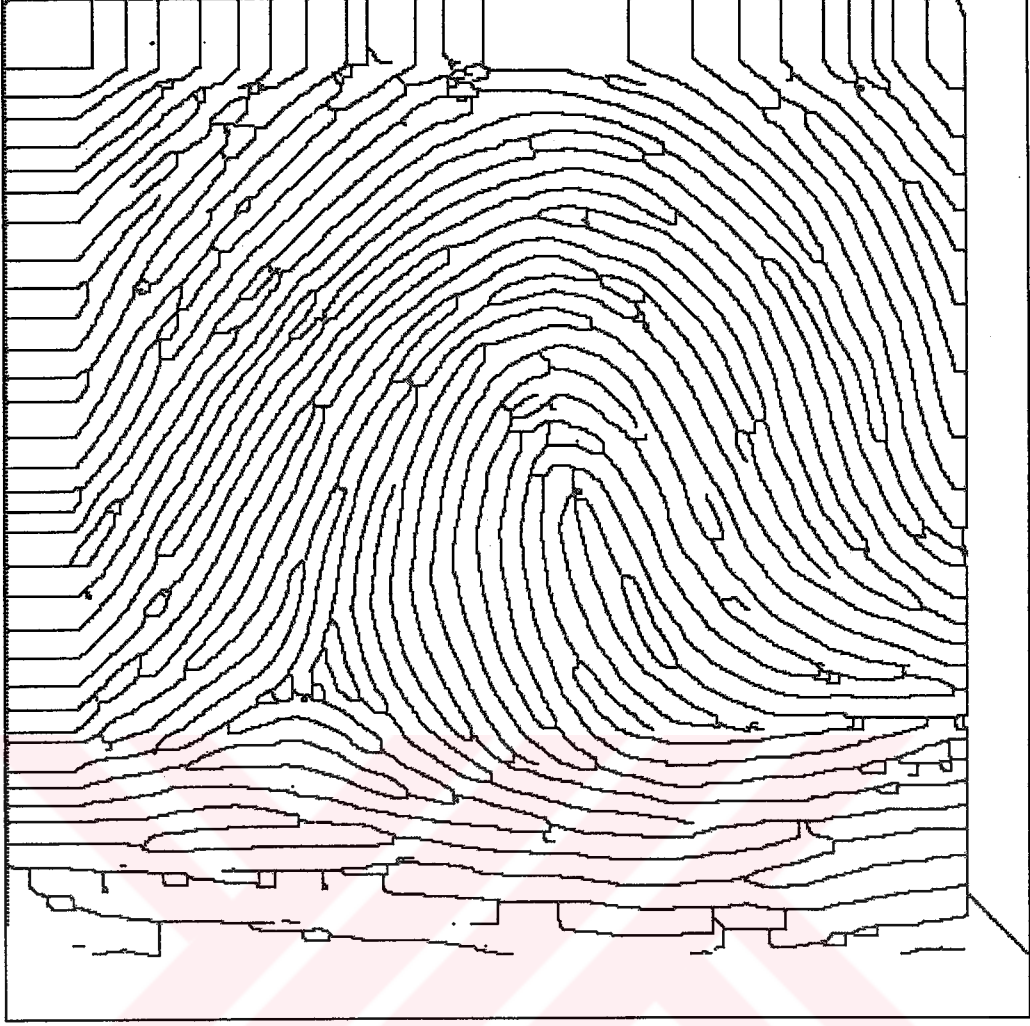
Şekil 7.19 Şekil 7.17 'nin II.Algoritma ile tek piksele indirgenmiş hali.



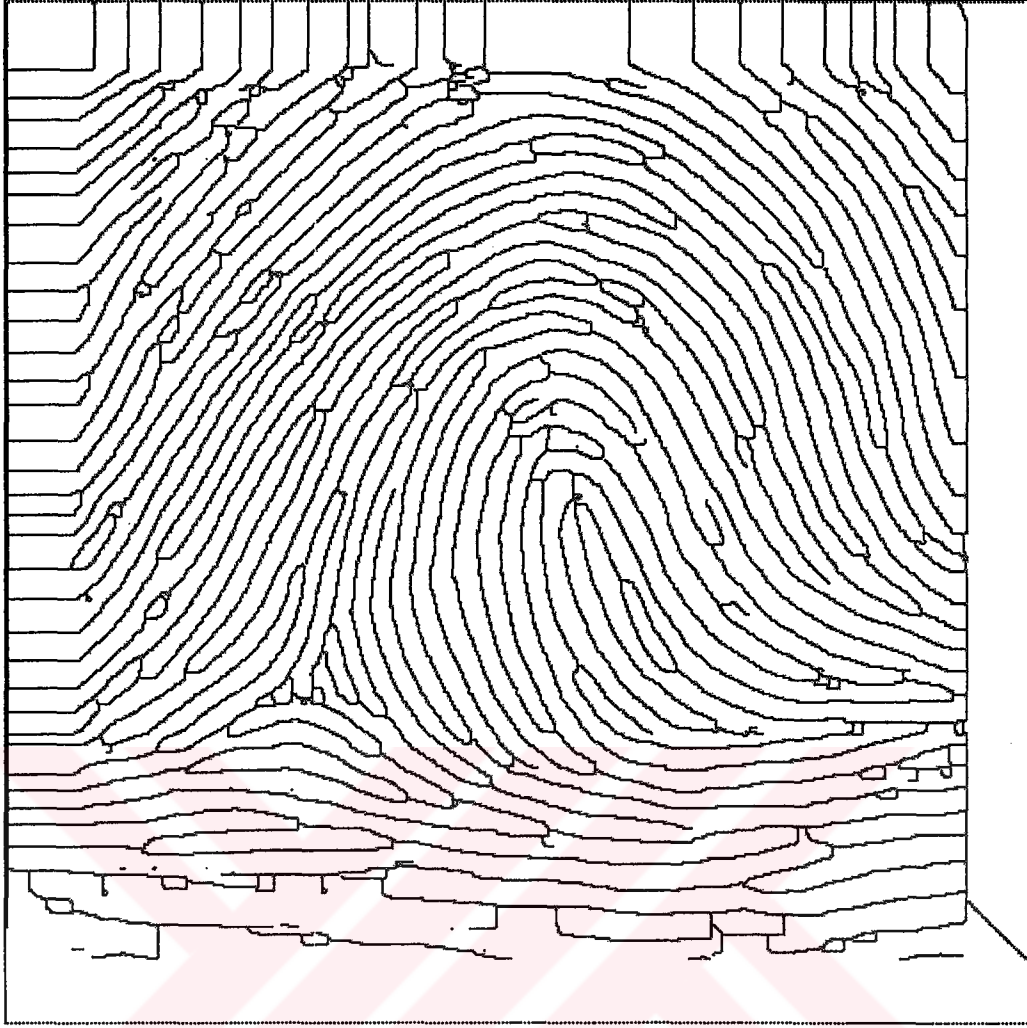
Şekil 7.20 İncelenen on farklı resim için RD Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları



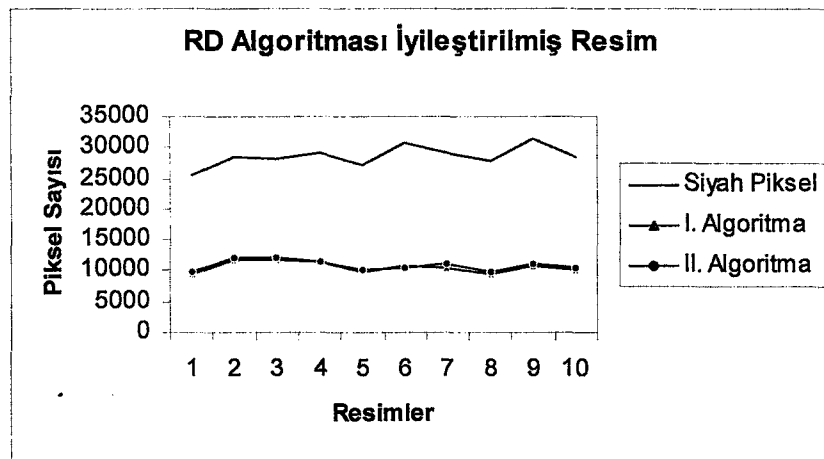
Şekil 7.21 YFF ile iyileştirilmiş test resminin RD Algoritması ile inceltilmiş hali.



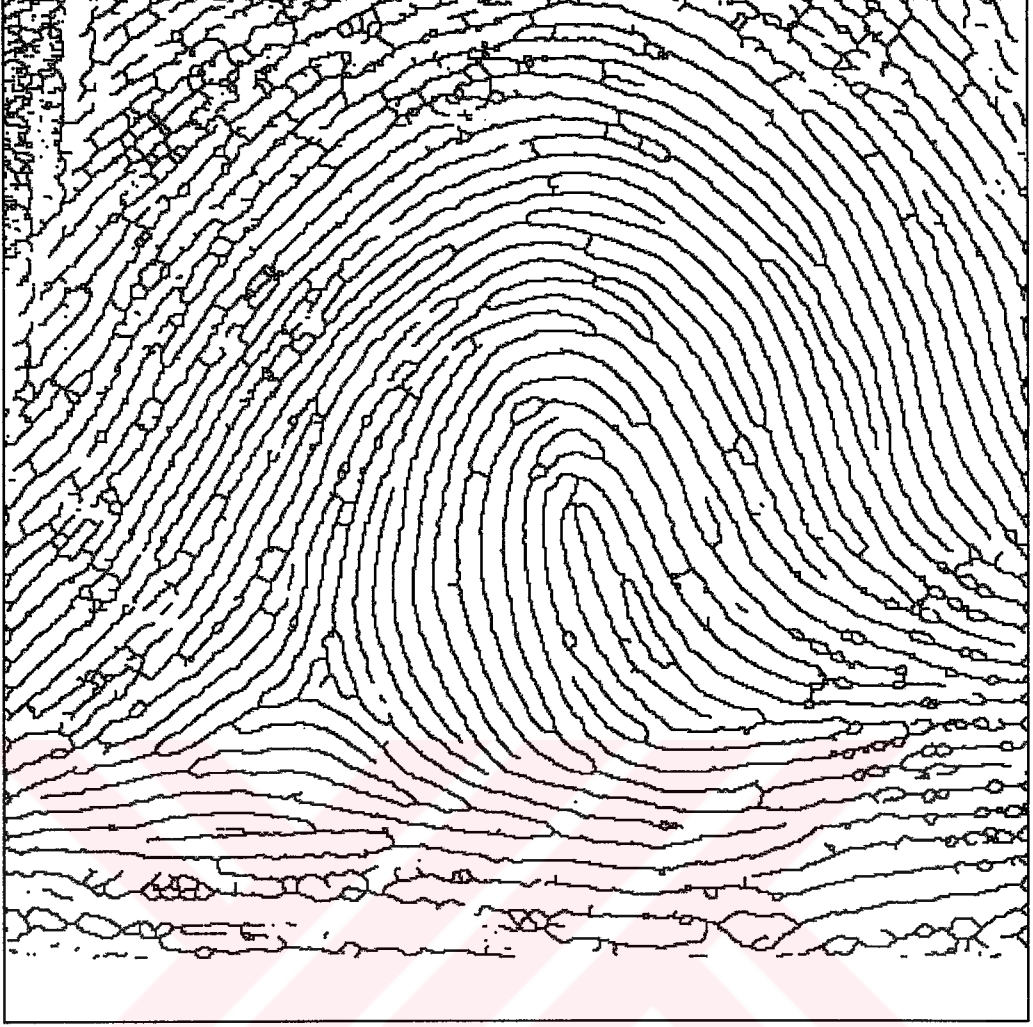
Şekil 7.22 Şekil 7.21 'rin I. Algoritma ile tek piksele indirgenmiş hali.



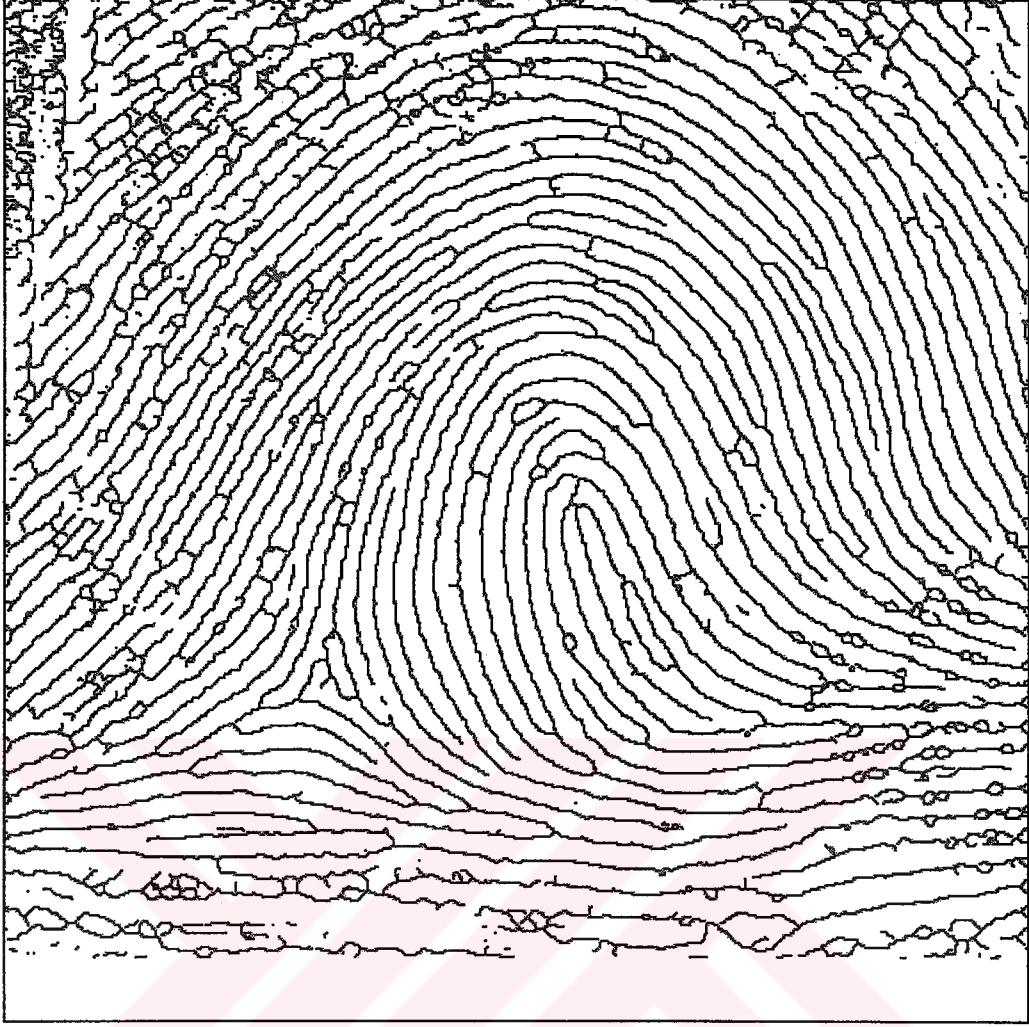
Şekil 7.23 Şekil 7.21 'in II. Algoritma ile tek piksele indirgenmiş hali



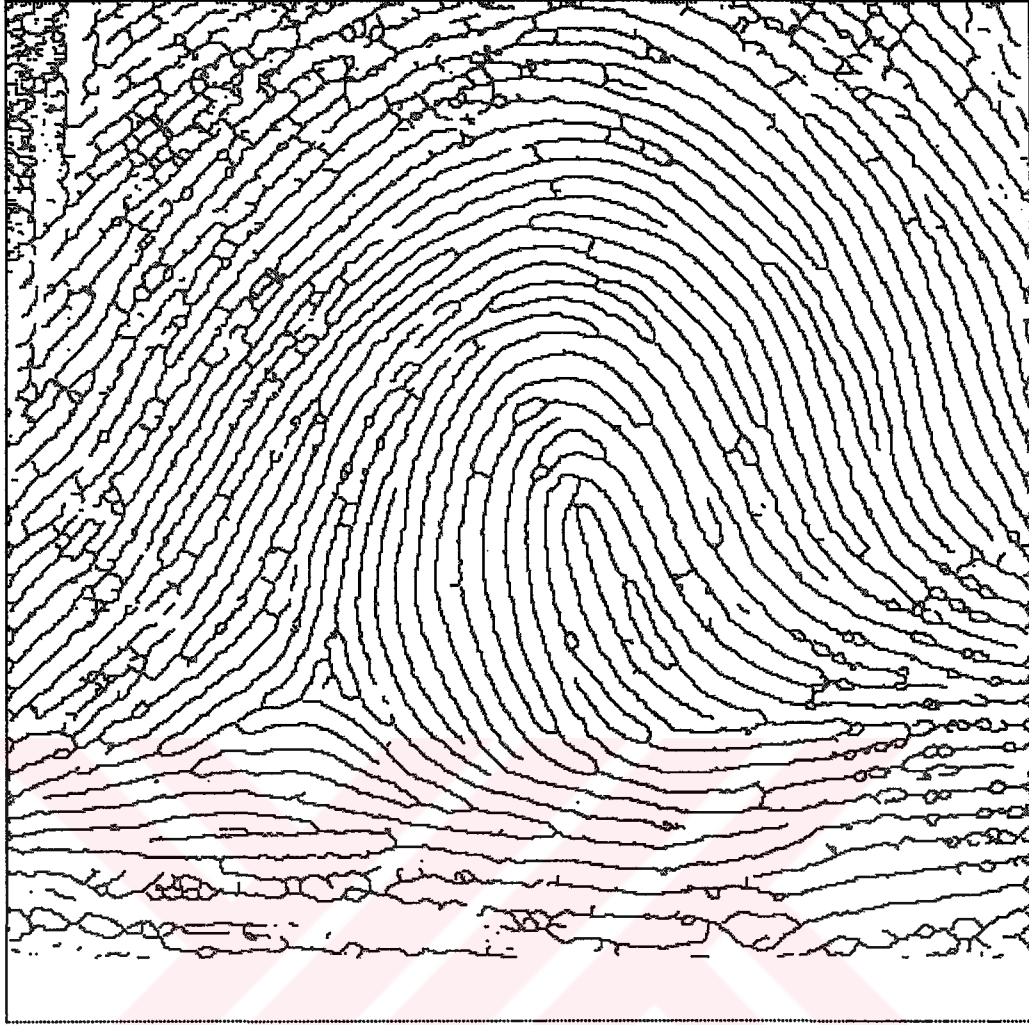
Şekil 7.24 İncelenen on farklı resmin iyileştirilmiş versiyonları için RD Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları



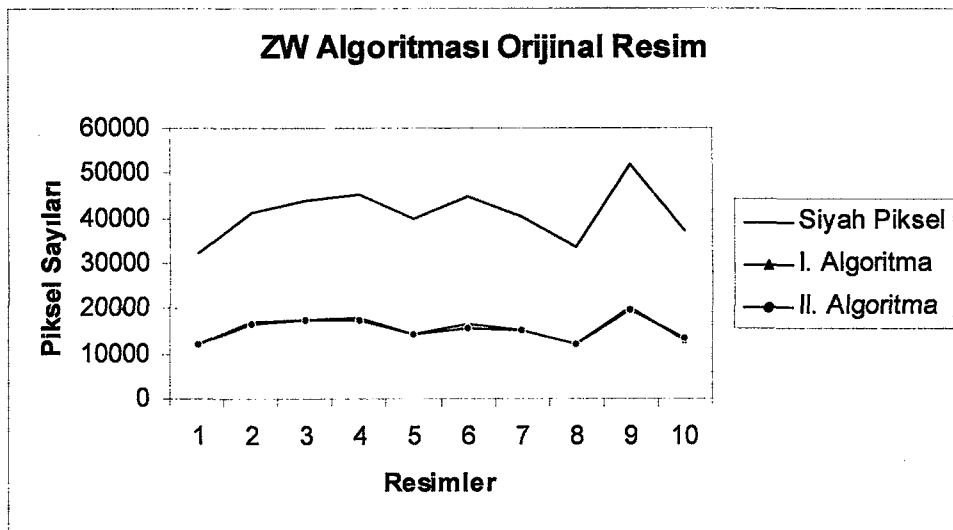
Şekil 7.25 Test resminin ZW Algoritması ile inceltilmiş hali.



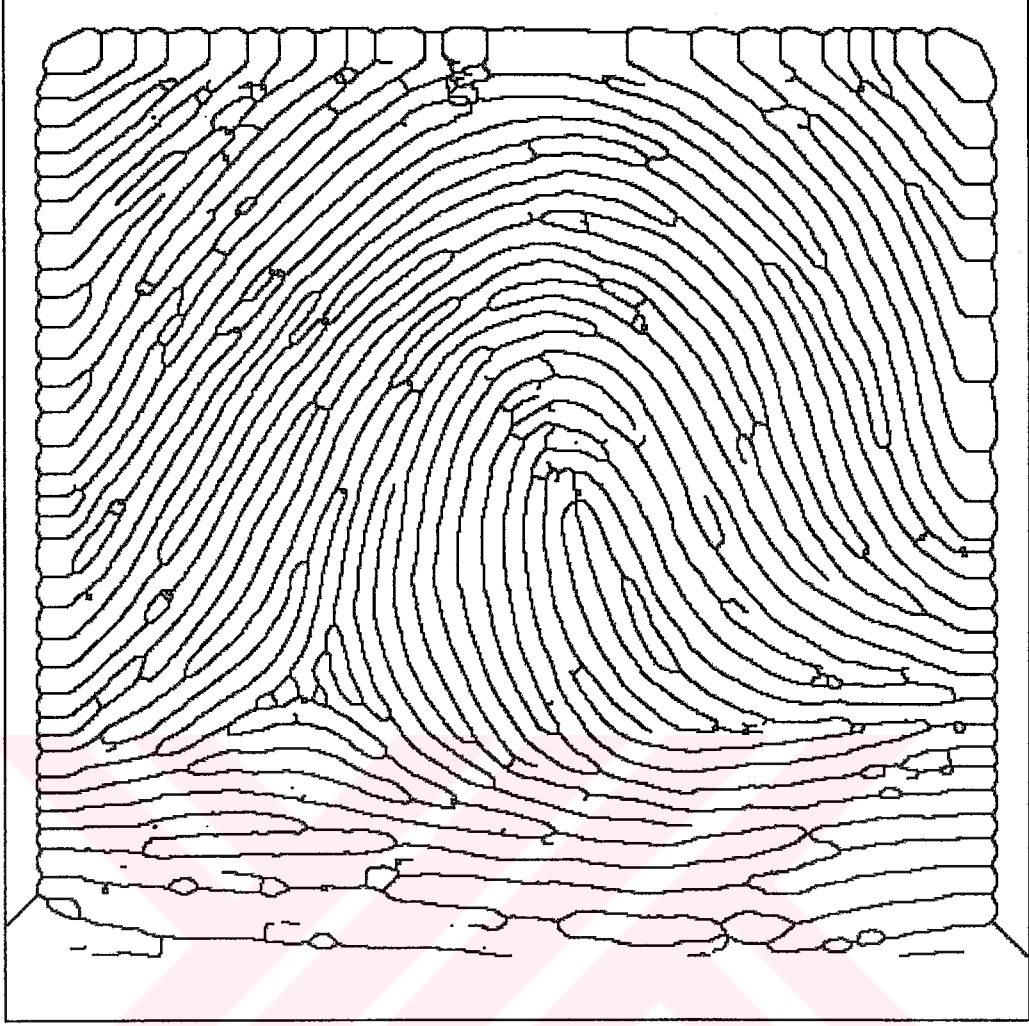
Şekil 7.26 Şekil 7.25 'in I. Algoritma ile tek piksele indirgenmiş hali.



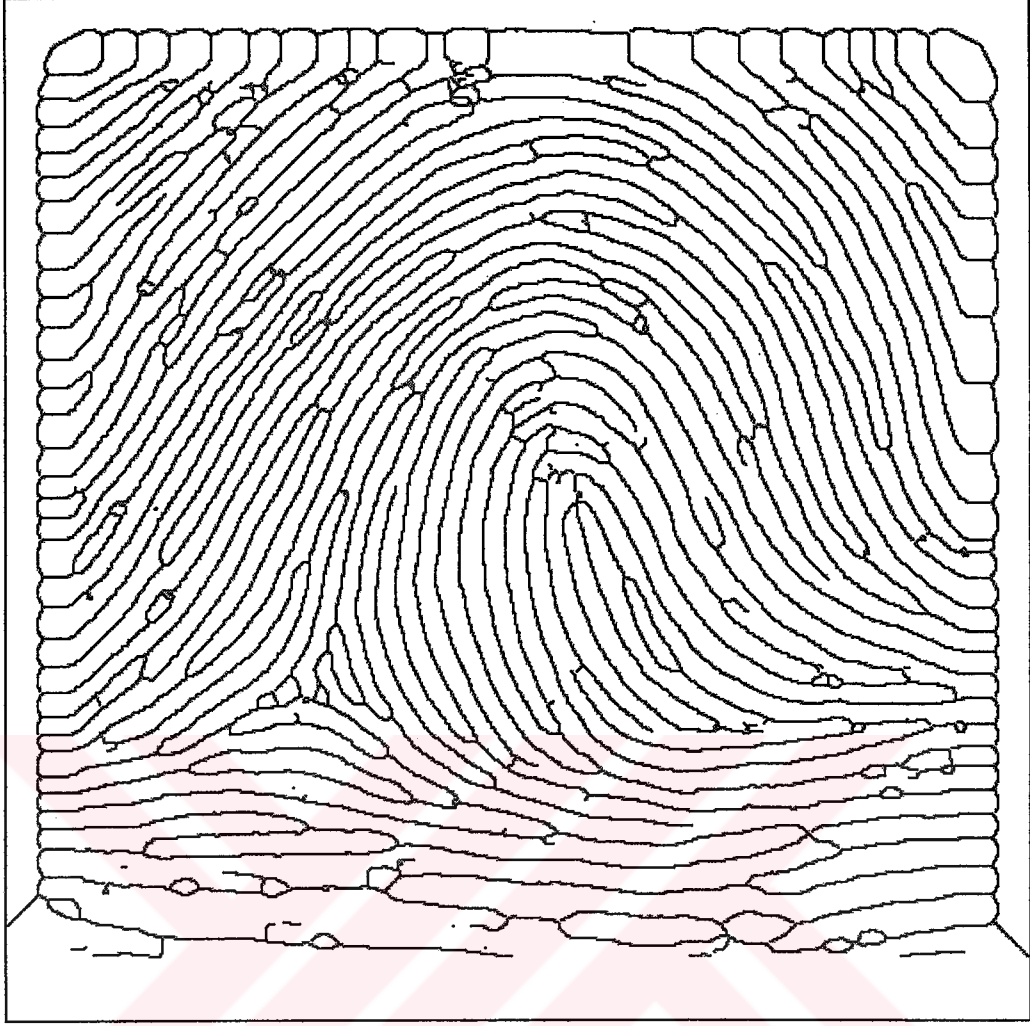
Şekil 7.27 Şekil 7.25 'in II. Algoritma ile tek piksele indirgenmiş hali.



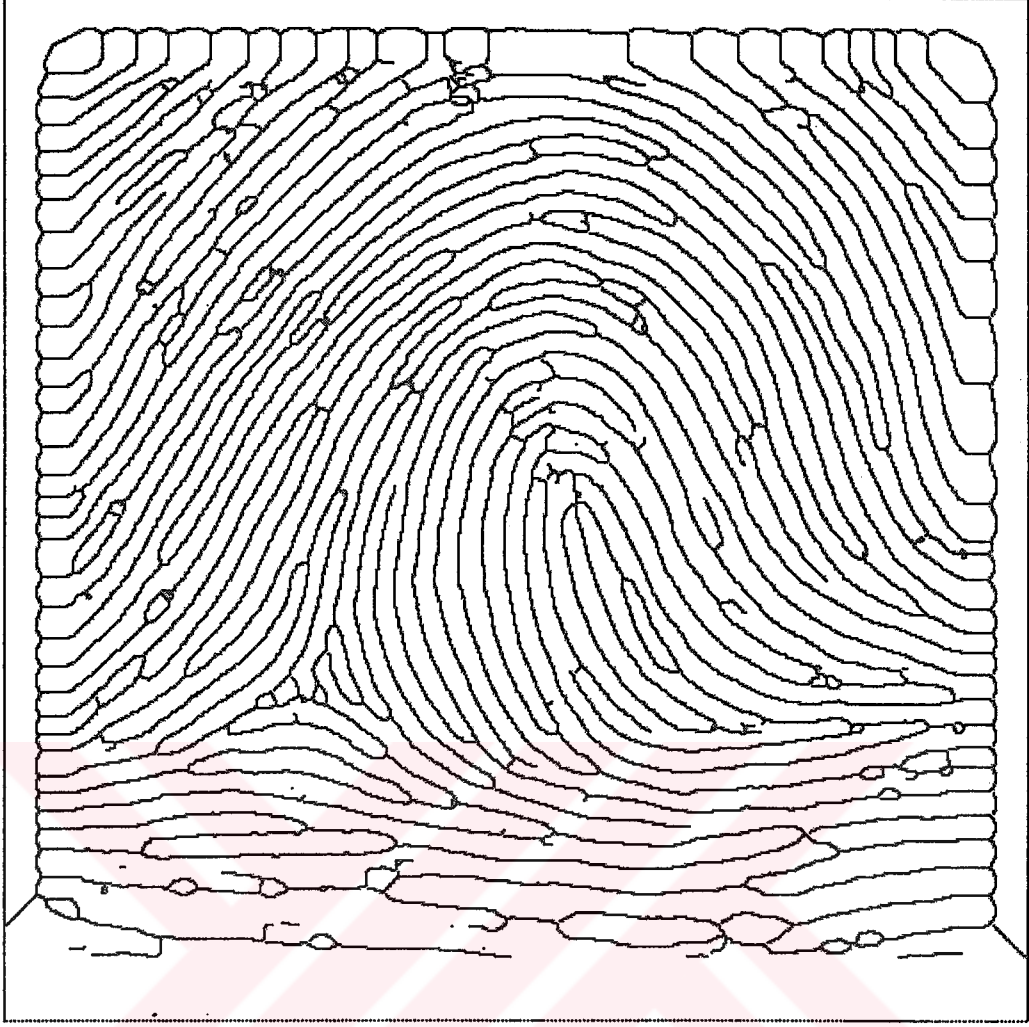
Şekil 7.28 İncelenen on farklı resim için ZW Algoritması ile inceltelen resimlerdeki siyah piksel sayısı ve I.,II. Algoritma ile bu resimlerden silinen piksel sayıları



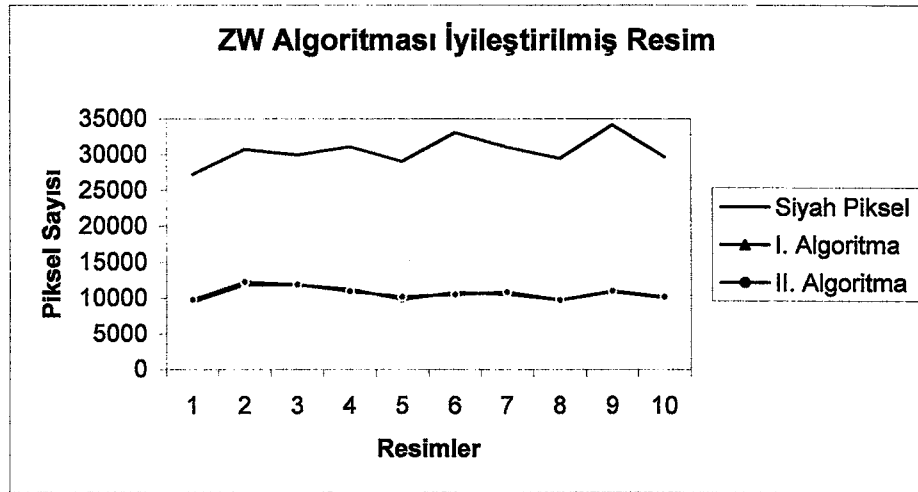
Şekil 7.29 YFF ile iyileştirilmiş test resminin ZW Algoritması ile inceltilmiş hali.



Şekil 7.30 Şekil 7.29 'zun I. Algoritma ile tek piksele indirgenmiş hali.



Şekil 7.31 Şekil 7.29 'zun II. Algoritma ile tek piksele indirgenmiş hali.



Şekil 7.32 İncelenen on farklı resmin iyileştirilmiş versiyonları için ZW Algoritması ile incelenen resimlerdeki siyah piksel sayısı ve I., II. Algoritma ile bu resimlerden silinen piksel sayıları

KAYNAKLAR

- Andrews, H.C., ve Hunt, BR. (1977). *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, N.J.
- Ballard B. ve Brown C. (1982) *Computer Vision*, Prentice-Hall, 1982, pp 24 - 30.
- Castleman, K., R., (1996) "Digital Image Processing", Prentice Hall International, USA
- Emiroğlu, I., (1997) "Fingerprint Image Enhancement & Recognition", Phd Thesis, University of Hertfordshire, October 1997
- Emiroğlu, I. ve Akhan, M., B., (1997a), "Pre-processing of Fingerprint Images", European Conference on Security and Detection, 1997, 147-151
- Gonzales R. ve Woods R., (1993), *Digital Image Processing*, Addison-Wesley Publishing Company.
- Jain, A., K., (1989), "Fundamentals of Digital Image Processing", Prentice-Hall, Englewood Cliffs, N.J.
- Kwok, P., C., K., (1988), "A Thinning Algorithm by Contour Generation", *Communications of the ACM*, Volume 31, Number 11, 1314-1324
- Kwon, J., S., Gi, J. ve W., Kang, E., K. (2001), "An Enhanced Thinning Algorithm Using Parallel Processing", *International Conference on Image Processing*, Thessaloniki, Greece, 7-10 October 2001, 752-755
- Lam, L., Lee, W. ve Suen, C., Y., (1992), "Thinning Methodologies-A Comprehensive Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 9, 869-887.
- Mehre B. M., Murthy N. N., Kapoor S., (1987) "Segmentation of Fingerprint Images using the Directional Image", *Pattern Recognition*, Vol. 20, No. 4, pp. 429-435
- Mehre B. M. ve Chatterjee B., (1989), "Segmentation of Fingerprint Images a Composite Method", *Pattern Recognition*, Vol. 22, No. 4, pp. 381-385.
- Pitas, I., (1993), *Digital Image Processing Algorithms*, Prentice Hall International (UK), Great Britain
- Raju, G., V., S. ve Xu, Y., (1991), "Study of Parallel Thinning Algorithms", 'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on , 13-16 Oct. 1991
- Sherlock, B., G., Monroe, D., M. ve Millard, K., (1992), "Algorithm for Enhancing Fingerprint Images", *Electronic Letters*, 28, 18, 1720-1721.
- Sherlock, B., G., Monroe, D., M. ve Millard, K., (1994), "Fingerprint Enhancement by Directional Fourier Filtering", *IEE Proceeding Vision, Image and Signal Processing*, 141, 2, 87-94
- Watson, C., I. ve Wilson, C., L. (1992), "NIST Special Database 4, Fingerprint Database", National Institute of Standards and Technology, Advanced Systems Division, Image Recognition Group, 17.03.1992

Zhang, Y., Y. ve Wang, P., S., P., (1992), "Analysis of Thinning Algorithms", Vol.III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on , 30 Aug.-3 Sept. 1992

Zhang, T., Y. ve · Suen, C., Y., (1984), "A Fast Parallel Algorithm for Thinning Digital Patterns", Communications of the ACM, Volume 27, Number 3, 236-239



ÖZGEÇMİŞ

Doğum tarihi	07.12.1980	
Doğum yeri	İstanbul	
Lise	1994-1997	Süleyman Nazif Lisesi
Lisans	1997-2002	Yıldız Teknik Üniversitesi Kimya-Metalurji Fak. Matematik Mühendisliği Bölümü
Yüksek Lisans	2002-2004	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Müh. Anabilim Dalı
Çalıştığı kurum(lar)		
2002 -	Devam ediyor	YTÜ Kimya-Metalurji Fak. Araştırma Görevlisi

