

**T.C.**  
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**  
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

**RASTER VERİ YAPISINDAKİ**  
**HARİTALARIN DALGACIK (WAVELET)**  
**TRANSFORMU İLE SIKIŞTIRILMASI**

**Uğur Kemal ÇELİK**  
**YÜKSEK LİSANS TEZİ**  
**JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ**  
**ANABİLİM DALI**

**GEBZE**  
**2004**

T.C.  
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ  
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

RASTER VERİ YAPISINDAKİ  
HARİTALARIN DALGACIK (WAVELET)  
TRANSFORMU İLE SIKIŞTIRILMASI

Uğur Kemal ÇELİK  
YÜKSEK LİSANS TEZİ  
JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ  
ANABİLİM DALI

TEZ DANIŞMANI  
Yrd.Doç.Dr.Halis SAKA

GEBZE  
2004

## ÖZET

Bu tezde bilgisayar hafızasında çok büyük miktarlarda yer kaplayan raster haritaları (sayısal orthofoto ve grafik kadastral paftalar) depolama problemini çözmek için Wavelet Dönüşümü ile görüntü sıkıştırma üzerinde çalışıldı.

Tezin uygulamasında, Visual Basic Programlama Dili ile bir Görüntü Sıkıştırma programı geliştirildi. Programda, wavelet dönüşümlerinden biri olan Haar Wavelet Dönüşümü kullanıldı. Program vasıtasıyla çeşitli raster görüntüler, farklı sıkıştırma katsayılarında sıkıştırıldıktan sonra, bu sıkıştırma katsayıları ile sıkıştırılan görüntülerin kullanılabilirlikleri tartışıldı.

## **SUMMARY**

In this thesis, in order to analyse storage problem of the raster maps (digital orthophoto and graphic cadastre hardcopy maps) which are covered with huge amount of memories in a computer, it was worked on image compression using Wavelet Transformation.

In the application of this study, an image compression program was developed with Visual Basic Programming Language. In the program, Haar Wavelet Transformation which is a kind of wavelet transformations was used. By means of the program, After raster images were compressed in various compression coefficients, the usages of the compressed images obtained from images were argued in this compression rates.

## TEŞEKKÜR

Bu tezin geliştirilme ve yazım aşamalarında benden desteğini esirgemeyen değerli hocam Yrd.Doç.Dr. Halis SAKA'ya , tez çalışmam boyunca benden maddi ve manevi hiçbir desteği esirgemeyen sevgili aileme, Tez yazımında bana yardımcı olan sevgili ablam Diş Hekimi Ülkü ÇELİK 'e, Orthofoto görüntüleri elde etmemde bana yardımcı olan M.N.G Firması Çalışanları'na ve grafik kadastral paftaları elde etmemde bana yardımcı olan Nallıhan Kadastro Müdürlüğü Çalışanları'na teşekkürü bir borç bilirim.

# İÇİNDEKİLER DİZİNİ

|                                                          | <b><u>Sayfa</u></b> |
|----------------------------------------------------------|---------------------|
| ÖZET                                                     | iv                  |
| SUMMARY                                                  | v                   |
| TEŞEKKÜR                                                 | vi                  |
| İÇİNDEKİLER DİZİNİ                                       | vii                 |
| ŞEKİLLER DİZİNİ                                          | ix                  |
| <br>                                                     |                     |
| 1. GİRİŞ                                                 | 1                   |
| 2. DALGACIKLAR                                           | 3                   |
| 2.1. Haar Bazı (Haar Ölçekleme Fonksiyonu)               | 3                   |
| 2.2. Haar Dalgacık Fonksiyonu                            | 4                   |
| 2.3. Haar Bazının Yapısı                                 | 4                   |
| 2.4. Açık Bir Bazda Sinyalin Genişlemesi                 | 5                   |
| 2.5. Bir Sinyalin Fourier Analizi                        | 5                   |
| 2.6. Bir Sinyalin Wavelet Dönüşümü                       | 6                   |
| 2.7. Ortalama ve Fark Alma İşlemlerinin Matematiği       | 6                   |
| 2.8. Dalgacıklar ve Filtreler                            | 9                   |
| 2.9. Görüntüyü Alt Bandlara Ayırıştırma (Subband Coding) | 10                  |
| 2.10. Çok Spektrumlu Görüntü Sıkıştırma Dalgacıklar      | 11                  |
| 2.11. Örnek Uygulama                                     | 13                  |
| 2.11.1. Ortalama ve Fark Alma Süreçleri                  | 13                  |
| 2.11.2. Temsili Görüntüyle Uygulama                      | 14                  |
| 2.11.3. Lineer Cebir Kullanarak Ortalama ve Fark Alma    | 18                  |
| 3. DİJİTAL GÖRÜNTÜLER                                    | 22                  |
| 3.1. Orthofoto Görüntüler                                | 23                  |
| 3.2. Grafik Kadastral Paftalar                           | 25                  |
| 4. GÖRÜNTÜ SIKIŞTIRMA PROGRAMI                           | 28                  |
| 4.1. Programın Arayüzü                                   | 28                  |
| 4.2. Programdaki Butonların Görevleri                    | 30                  |
| 4.2.1. Görüntü Yükle Butonu                              | 30                  |
| 4.2.2. Gri Tonlamalı Görüntüyü Yükle Butonu              | 30                  |

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| 4.2.3. Gerçek Görüntü Matrisi Butonu                                                       | 30 |
| 4.2.4. Sıkıştır Butonu                                                                     | 31 |
| 4.2.5. Sıkıştırılmış Görüntü Butonu                                                        | 32 |
| 4.2.6. .SKM Formatında Kaydet Butonu                                                       | 32 |
| 4.2.7. Transforme Olmuş Görüntü Butonu                                                     | 32 |
| 4.2.8. Transforme Olmuş Görüntü Matrisi Butonu                                             | 33 |
| 4.2.9. Sıkıştırılmış Görüntüyü Dosyadan Yükle Butonu                                       | 33 |
| 4.2.10. BİTMAP Formatında Kaydet Butonu                                                    | 34 |
| 4.2.11. Çıkış Butonu                                                                       | 34 |
| 4.2.12. Temizle Butonu                                                                     | 34 |
| 4.3. Programın Çalışma Prensipleri                                                         | 35 |
| 4.3.1. Görüntünün Imagebox'a Yüklenmesi                                                    | 35 |
| 4.3.2. Spektral Dönüşüm                                                                    | 36 |
| 4.3.3. Görüntüyü Sıkıştırma Süreci                                                         | 37 |
| 4.3.4. Sıkıştırılmış Görüntünün Bilgisayara Kaydedilmesi                                   | 42 |
| 4.3.4.1. Kayıt Formatı                                                                     | 42 |
| 4.3.5. Daha Önce .SKM Formatında Kaydedilmiş Bir Görüntünün<br>Bilgisayarda Görüntülenmesi | 45 |
| 4.3.6. Görüntü Matrislerinin MsFlexGrid ile Görüntülenmesi                                 | 47 |
| 5. UYGULAMA                                                                                | 49 |
| 5.1. Orthofoto Görüntülerin Sıkıştırılması                                                 | 49 |
| 5.2. Grafik Kadastral Paftaların Sıkıştırılması                                            | 52 |
| 6. SONUÇ VE ÖNERİLER                                                                       | 58 |
| KAYNAKLAR                                                                                  | 60 |
| ÖZGEÇMİŞ                                                                                   | 62 |
| EKLER                                                                                      |    |
| EK- 1: Görüntü Sıkıştırma Programı                                                         |    |

## ŞEKİLLER DİZİNİ

| <b><u>Sekil</u></b>                                                                  | <b><u>Sayfa</u></b> |
|--------------------------------------------------------------------------------------|---------------------|
| 2.1. Bir Görüntünün Alt Bandlara Dalgacık Çözümü İle Ayırıştırılması                 | 10                  |
| 2.2. Çok Spektrumlu Görüntülerin Ayırıştırılması                                     | 12                  |
| 2.3. Dalgacık Dönüşümünün Yapısı                                                     | 12                  |
| 2.4. A Matrisi                                                                       | 15                  |
| 2.5. A Matrisinin Görüntüsü                                                          | 15                  |
| 2.6. Yaklaşım Matrisinin Görüntüsü                                                   | 17                  |
| 3.1. 256x256 Piksellik (6.75x6.75 cm <sup>2</sup> ) Orthofoto Görüntü Örnekleri      | 25                  |
| 3.2. 1790x1280 Piksellik (33.7x47.28 cm <sup>2</sup> ) Grafik Kadastral Pafta Örneği | 26                  |
| 3.3. Elektromanyetik Spektrum                                                        | 27                  |
| 4.1. Programın Arayüzü                                                               | 29                  |
| 4.2. Programın MsFlexGrid Bileşeni                                                   | 31                  |
| 4.3. Segmentlerine Ayırıştırılmış Bir İnsan Görüntüsü                                | 33                  |
| 5.1. 256x256 Piksellik (192 kilobayt) Orijinal Orthofoto Görüntü                     | 49                  |
| 5.2. $\delta = 9$ , %94'lük Sıkıştırma yüzdesi ile (11.5 kilobayt)                   | 50                  |
| 5.3. $\delta = 7$ , %90'lık Sıkıştırma yüzdesi ile (19 kilobayt)                     | 50                  |
| 5.4. $\delta = 5$ , %82'lik Sıkıştırma yüzdesi ile (34,5 kilobayt)                   | 51                  |
| 5.5. $\delta = 3$ , %69'luk Sıkıştırma yüzdesi ile (59,5 kilobayt)                   | 51                  |
| 5.6. $\delta = 5$ katsayısı ile sıkıştırılmış görüntü %85'lik sıkıştırma (981 kb)    | 54                  |
| 5.7. $\delta = 7$ katsayısı ile sıkıştırılmış görüntü %88'lik sıkıştırma (846 kb)    | 55                  |
| 5.8. $\delta = 15$ katsayısı ile sıkıştırılmış görüntü %93'lük sıkıştırma (466 kb)   | 56                  |
| 5.9. $\delta = 22$ katsayısı ile sıkıştırılmış görüntü %96'lık sıkıştırma (297 kb)   | 57                  |

# 1. GİRİŞ

17. Yüzyılda Fransız matematikçi J. Fourier, herhangi bir priyodik fonksiyonun, periyodik kompleks üs fonksiyonlarının sonsuz bir toplamı olarak ifade edilebileceğini göstermiştir. Periyodik fonksiyonların bu önemli özelliği keşfedildikten yıllar sonra, Fourier'in fikirleri periyodik olmayan ilk fonksiyonlar için üretildi ve böylece periyodik veya periyodik olmayan ayırık zamanlı sinyaller geliştirildi. 1965'de Hızlı Fourier Dönüşümü olarak isimlendirilen yeni bir algoritma geliştirildi ve Fourier Transformu ve bu transformun uzantıları, sinyal analizleri ve temsilleri için bir araç oldu. 1970'den beri dönüşüm kodlayan uygulamalar, özellikle gerçek baz tabanlı discrete cosine transform üzerinde çalışıldı. Bu dönüşümü kullanan algoritmalara JPEG, MPEG I, MPEG II örnek olarak verilebilir [Sönmez A.Gürkan,1994,s.2,3].

Dalgacık Dönüşümleri (Wavelet Transforms) son yıllarda, sıkıştırma performansı ve artan fonksiyonellik desteğiyle görüntü kodlamada iyi bir konuma gelmiştir. Bundan dolayı; dalgacık tabanlı görüntü sıkıştırma algoritmaları, JPEG-2000 'in (The Joint Photographic Experts Group) kullandığı bir yöntem olmuştur [Auwera G. Van der, Munteanu A. and Cornelis J., 2003, s.1].

Dalgacık Dönüşümü, işaretlerin değişik ölçeklerde incelenmesi amacıyla yapılan çalışmalardan doğmuştur. 1980 'li yıllarda Fransız araştırmacılarından J. Morlet, A. Grossman ve Y. Meyer tarafından ortaya atılmıştır. Daha sonraları, I. Daubechies ve S.G. Mallat, dalgacık teorisini işaret işleme alanında geliştirmişlerdir. Günümüzde, birbirinden farklı birçok alanda dalgacık dönüşümünden yararlanılmaktadır [Sarı Tunç, 2003, s.1].

Dalgacık Dönüşümleri; radyoaktiflik içeren bilgisayar grafikleri, yüksek çözünürlüklü resimleme, eğri tasarımı, dişli (mesh) optimizasyonu, ses dalgalarının görüntülenmesi, görüntü arama, animasyon kontrolü ve bunların içerisinde ilk uygulanan ve en önemlilerinden biri olan görüntü sıkıştırma gibi çok geniş uygulama alanlarında kullanılmaktadır [Kopp Manfred, 2003, s.1].

Bu kullanımlara ek olarak, parmak izi sınıflandırma problemi de örnek olarak gösterilebilir. Amerika Birleşik Devletleri bünyesinde bulunan F.B.I örgütü, parmak izlerindeki gri gölgeleri belirlemek amacıyla piksel başına 8 bit kullanır ve inch başına 500 piksel depolar. Böylece elektronik formda parmak izlerini depolamak için tek parmak için 700.000 piksellik hafıza kullanır [Wickerhauser Mladen Victor, 1994, s.387].

Bu tezde birkaç spektral dalga boyundan bilgi içeren orthofoto ve bilgisayar ortamına raster veri tipinde aktarılan grafik kadastral paftalardan oluşan görüntüler üzerinde çalışıldı. Sıkıştırılmamış bir görüntüyü bilgisayar ortamında depolamak için oldukça büyük miktarda hafıza gerekir. Bu bilgiyi internet ortamında bir network ağı sayesinde bir yerden başka bir yere yollamak çok zaman harcanmasına sebebiyet verir. Bu nedenlerden dolayı raster görüntülerin içerdiği gereksiz bilgileri yok etmek ve çok büyük miktardaki verilerin boyutunu düşürmek için etkili kodlama sistemleri kullanılmalıdır.

Dalgacık analizleri sayesinde, sinyaller kendilerini oluşturan parçalara indirgenir. Bu özellik, veri sıkıştırma için dalgacık fonksiyonlarını uygun araçlar yapar.

Haar Wavelet, dalgacık dönüşümleri içerisinde en basit dönüşümlerden birisidir [Pinnamaneni Pujita, 2003, s.1]. Çok spektrumlu görüntüler, Haar Wavelet Dönüşümü sayesinde büyük oranlarda sıkıştırılabilmektedir. Bu tezin yazım amacı olan tez projesinde, çok spektrumlu orthofoto görüntüler ve raster veri tipine dönüştürülen grafik kadastral pafta görüntüleri kullanıldı. Bu görüntüler, Ankara'da bulunan M.N.G isimli fotogrametrik uygulamalar yapan bir şirketten ve Nallıhan Kadastro Müdürlüğü'nden elde edildi. Tez projesinin yapımında Visual Basic 6.0 programlama dili kullanıldı. Ayrıca uygulamaların bir bölümü de Matlab 6.5 programı ile geliştirildi.

Bu çalışma 6 ayrı bölümden oluşmaktadır. Bölüm1; "Giriş" kısmıdır. Bölüm 2'de; "Dalgacıklar ve örnek uygulama", Bölüm 3 'de "Dijital Görüntüler", Bölüm 4 'te "Görüntü Sıkıştırma Programı", Bölüm 5'te "Uygulama", Bölüm 6'da "Sonuç ve Öneriler" kısmı yer almaktadır.

## 2. DALGACIKLAR

### 2.1. Haar Bazı (Haar Ölçekleme Fonksiyonu)

Haar ölçekleme (scaling) fonksiyonu,  $\phi : R \rightarrow R$  olmak üzere,

$$\phi(t) = \begin{cases} 1 & t \in [0,1) \\ 0 & t \notin [0,1) \end{cases} \quad (2.1)$$

ifadesi tarafından belirlenir.  $\phi_i^j : R \rightarrow R$  olmak üzere

$$\phi_i^j(t) = \sqrt{2^j} \cdot \phi(2^j t - i), \quad j = 0,1,\dots \text{ ve } i = 0,1,\dots,2^j - 1 \quad (2.2)$$

formülü ile formülize edilir.  $V^j$  vektör uzayı ise,

$$V^j = sp\{\phi_i^j\}_{i=0,1,\dots,2^j-1} \quad (2.3)$$

ifadesi ile belirlenir. Burada  $sp$ , lineer aralığı (span) gösterir.  $j$  indeksi, genişleme (dilation) ve  $i$  indeksi ise öteleme (translation) anlamına gelmektedir.

$$\langle \phi_i^j, \phi_i^j \rangle = \int_0^1 \phi_i^j(t)^2 dt = 1 \text{ eşitliği sağlandığından dolayı } \sqrt{2^j}, \text{ normalizasyon}$$

sabiti olarak seçilir. Eğer ölçekleme fonksiyonu  $[0,1]$  aralığından başka bir aralıkta seçilirse normalizasyon sabiti de değişecektir. Ayrıca  $V^j$  uzayı için,

$$V^j \subseteq V^{j+1} \quad (2.4)$$

özelliliği sağlanır.

## 2.2. Haar Dalgacık Fonksiyonu

Haar dalgacık fonksiyonu,  $\psi : R \rightarrow R$  olmak üzere,

$$\psi(t) = \begin{cases} 1 & t \in [0, 1/2) \\ -1 & t \in [1/2, 1) \\ 0 & t \notin [0, 1) \end{cases} \quad (2.5)$$

ifadesi ile belirlenir.  $\psi_i^j(t) : R \rightarrow R$  olmak üzere

$$\psi_i^j(t) = \sqrt{2^j} \cdot \psi(2^j t - i), \quad j = 0, 1, \dots \text{ ve } i = 0, 1, \dots, 2^j - 1 \quad (2.6)$$

formülü ile formülize edilir.  $W^j$  vektör uzayı ise,

$$W^j = sp\{\psi_i^j\}_{i=0,1,\dots,2^j-1} \quad (2.7)$$

ifadesi ile belirlenir. Burada  $sp$ , lineer aralığı (span) gösterir.  $j$  indeksi, genişleme (dilation) ve  $i$  indeksi ise çeviri (translation) anlamına gelmektedir.

$$\langle \psi_i^j, \psi_i^j \rangle = \int_0^1 \psi_i^j(t)^2 dt = 1 \text{ eşitliği sağlandığından dolayı } \sqrt{2^j}, \text{ normalizasyon}$$

sabiti olarak seçilir. Eğer ölçekleme fonksiyonu  $[0, 1]$  aralığından başka bir aralıkta seçilirse normalizasyon sabiti de değişecektir. Ayrıca  $W^j$  uzayı için,

$$W^j \subseteq V^{j+1} \quad (2.8)$$

özellği sağlanır.

## 2.3. Haar Bazının Yapısı

Haar Bazı ve diğer dalgacık bazları,

$$V^{j+1} = V^j \oplus W^j \quad (2.9)$$

ifadesi gibi çok önemli bir özelliğe sahiptirler.

## 2.4. Açık bir Bazda (On-Basis) bir Sinyalin Genişlemesi

$f \in R^N$ ,  $N$  örnekli ( $N$  - boyutlu vektör),

$$f = (f[0], f[1], \dots, f[N-1]) \quad (2.10)$$

ifadesi ile verilen sonlu bir sinyali gösterir.  $\{V_k\}_{k=0,1,\dots,N-1}$ ,  $R^N$  uzayında açık bir baz olsun. Sinyal bu bazda,

$$f = \langle f, v_0 \rangle v_0 + \langle f, v_1 \rangle v_1 + \dots + \langle f, v_{N-1} \rangle v_{N-1}. \quad (2.11)$$

şeklinde genişler.

## 2.5. Bir Sinyalin Fourier Analizi

$$v_k = \frac{1}{\sqrt{N}} \left( 1, e^{j2\pi \frac{k}{N}}, e^{j2\pi \frac{2k}{N}}, \dots, e^{j2\pi \frac{(N-1)k}{N}} \right). \quad (2.12)$$

$v_k$  vektörünün  $(n+1)$ 'nci elemanı  $v_k[n]$  olarak gösterilirse,  $f$  fonksiyonu 2.11'de gösterildiği gibi genişletilebilir. Bu durumda,

$$\langle f, v_k \rangle = \sum_{n=0}^{N-1} f[n] \cdot v_k[n]^* = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f[n] \cdot e^{-j2\pi \frac{nk}{N}} \quad (2.13)$$

ifadesi elde edilir. Burada  $*$  işareti kompleks eylem çekimini (conjugation) gösterir.

Sayısal olarak Fast Fourier Transform (FFT) kullanımı ile  $\langle f, v_k \rangle$  hesaplanabilir.

## 2.6. Bir Sinyalin Wavelet Dönüşümü

Anlaşılabilirlik açısından 8 örneklı bir sinyal düşünölebilir. Ancak, anlatılacak prosedür, herhangi sonlu bir sinyal için de üretilebilir. Sinyalin boyutu  $k$  pozitif tamsayısı için  $2^k$  alınırsa, analiz basitleşir.

$f \in R^8$ 'dir ve bu sinyal baz vektörleri tarafından aralıklandırılan(span)  $V^0 \oplus W^0 \oplus W^1 \oplus W^2$  içerisinde genişletilirse,

$$\begin{aligned} f = & \langle f, \phi_{\cdot 0}^0 \rangle \phi_{\cdot 0}^0 + \langle f, \psi_{\cdot 0}^0 \rangle \psi_{\cdot 0}^0 + \langle f, \psi_{\cdot 0}^1 \rangle \psi_{\cdot 0}^1 + \\ & \langle f, \psi_{\cdot 1}^1 \rangle \psi_{\cdot 1}^1 + \langle f, \psi_{\cdot 0}^2 \rangle \psi_{\cdot 0}^2 + \langle f, \psi_{\cdot 1}^2 \rangle \psi_{\cdot 1}^2 + \\ & \langle f, \psi_{\cdot 2}^2 \rangle \psi_{\cdot 2}^2 + \langle f, \psi_{\cdot 3}^2 \rangle \psi_{\cdot 3}^2. \end{aligned} \quad (2.14)$$

yukarıdaki ifade elde edilir. Burada  $\langle f, \phi_{\cdot i}^j \rangle$  ve  $\langle f, \psi_{\cdot i}^j \rangle$  ifadeleri iç çarpımlardır. Bu iç çarpımlar prensipte integral alma işlemi ile veya ayrık durumlarda bir toplam işlemi ile gerçekleştirilebilir. Pratikte ise bu iç çarpımlar ortalama ve fark alma işlemi ile gerçekleştirilir.

## 2.7. Ortalama ve Fark Alma İşlemlerinin Matematiği

Sinyal Fonksiyonu  $f = (2, 5, 8, 9, 7, 4, -1, 1)$  olarak alınırsa, orijinal sinyalin katsayıları  $V^3$  'de genişleyen  $f$  vektörünün katsayılarıdır. Burada  $j = 3$  ve  $i = 2^j - 1 = 2^3 - 1 = 7$  olacaktır. Bu durumda,

$$\begin{aligned} \langle f, \phi_{\cdot 0}^3 \rangle = 2, \quad \langle f, \phi_{\cdot 1}^3 \rangle = 5, \quad \langle f, \phi_{\cdot 2}^3 \rangle = 8, \quad \langle f, \phi_{\cdot 3}^3 \rangle = 9, \quad \langle f, \phi_{\cdot 4}^3 \rangle = 7, \quad \langle f, \phi_{\cdot 5}^3 \rangle = 4, \\ \langle f, \phi_{\cdot 6}^3 \rangle = -1, \quad \langle f, \phi_{\cdot 7}^3 \rangle = 1 \end{aligned} \text{ iç çarpımları geçerli olacaktır.}$$

Burada amaç,  $f$  fonksiyonunu  $V^0 \oplus W^0 \oplus W^1 \oplus W^2$  ifadesi içinde genişletmektir. Bu genişletme süreci üç adımda yapılır.

## 1. Adım

Birinci adımda  $f$  fonksiyonu,  $V^2 \oplus W^2$  uzayında genişletilir.  $f_1 = W_1 f$  formülü ile matris çarpımı şeklinde bu işlem gerçekleştirilir.

$$W_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \quad (2.15)$$

Yukarıdaki  $W_1$  matrisinde ilk dört satır, yukarıdan aşağıya sırası ile  $\phi_0^2, \phi_1^2, \phi_2^2, \phi_3^2$  baz vektörleridir. Bu vektörler  $V^2$  uzayında konumlanırlar. Son dört satır yukarıdan aşağıya sıra ile  $\psi_0^2, \psi_1^2, \psi_2^2, \psi_3^2$  baz vektörleridir ve  $W^2$  uzayında konumlanırlar. Çarpım işleminden sonra;

$$f = \frac{1}{\sqrt{2}} (7\phi_0^2 + 17\phi_1^2 + 11\phi_2^2 + 0\phi_3^2 - 3\psi_0^2 - 1\psi_1^2 + 3\psi_2^2 - 2\psi_3^2) \quad (2.16)$$

ifadesi elde edilir.  $f$  fonksiyonu  $V^2 \oplus W^2$  uzayında 2.16 'da gösterilen şekilde genişler.

## 2. Adım

Bu adımda  $f$  fonksiyonu  $V^1 \oplus W^1 \oplus W^2$  'de genişletilir. Bu işlem için  $f_2 = W_2 f_1$  matris çarpımından faydalanılır.

$$W_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.16)$$

Yukarıdaki matriste yukarıdan aşağı ilk iki satır  $\phi_{\cdot 0}^1, \phi_{\cdot 1}^1$  baz vektörleridir. Bu iki vektör  $V^1$  uzayında konumlanır. Üçüncü ve Dördüncü satırlar  $W^1$  uzayında konumlanan  $\psi_{\cdot 0}^1, \psi_{\cdot 1}^1$  baz vektörleridir. Son dört satır ise  $W^2$  uzayında konumlanan  $\psi_{\cdot 0}^2, \psi_{\cdot 1}^2, \psi_{\cdot 2}^2, \psi_{\cdot 3}^2$  baz vektörleridir. Yukarıdaki çarpım yapıldıktan sonra

$$f = \frac{1}{\sqrt{2}} \left( \frac{24}{\sqrt{2}} \phi_{\cdot 0}^1 + \frac{11}{\sqrt{2}} \phi_{\cdot 1}^1 - \frac{10}{\sqrt{2}} \psi_{\cdot 0}^1 + \frac{11}{\sqrt{2}} \psi_{\cdot 1}^1 - 3\psi_{\cdot 0}^2 - 1\psi_{\cdot 1}^2 + 3\psi_{\cdot 2}^2 - 2\psi_{\cdot 3}^2 \right) \quad (2.17)$$

ifadesi elde edilir.

### 3. Adım

Son adımda  $f$  fonksiyonu  $V^0 \oplus W^0 \oplus W^1 \oplus W^2$  uzayında genişletilir.  $f_3 = W_3 f_2$  çarpımı uygulanır.

$$W_3 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.18)$$

Yukarıdaki matriste yukarıdan aşağıya sırası ile ilk satır  $V^0$  uzayında bulunan  $\phi_0^0$  baz vektörüdür, ikinci satır  $W^0$  uzayında bulunan  $\psi_0^0$  baz vektörüdür. Üçüncü ve dördüncü satırlar  $W^1$  uzayında bulunan  $\psi_0^1, \psi_1^1$  baz vektörleridir. Son dört satır ise  $W^2$  uzayında konumlanan  $\psi_0^2, \psi_1^2, \psi_2^2, \psi_3^2$  baz vektörleridir. Çarpım yapıldığı zaman,

$$f = \frac{1}{\sqrt{2}} \left( \frac{35}{2} \phi_0^0 + \frac{13}{2} \psi_0^0 - \frac{10}{\sqrt{2}} \psi_0^1 + \frac{11}{\sqrt{2}} \psi_1^1 - 3\psi_0^2 - 1\psi_1^2 + 3\psi_2^2 - 2\psi_3^2 \right)$$

$$\cong (12.4\phi_0^0 + 4.60\psi_0^0 - 5.00\psi_0^1 + 5.50\psi_1^1 - 2.12\psi_0^2 - 0.707\psi_1^2 + 2.12\psi_2^2 - 1.41\psi_3^2)$$

ifadesi elde edilir. Elde edilen ifade,  $f$  fonksiyonunun dalgacık dönüşümüdür.

## 2.8. Dalgacıklar ve Filtreler

Yukarıda ifade edilen ortalama ve fark alma metodları ayrıca, verinin bir çeşit filtreden geçirilmesi gibi düşünülebilir. Ortalama süreci, verinin bir low-pass filtreden geçirilmesi anlamına gelir. Low-pass filtre, yüksek frekanslı veriyi çıkarır. Verideki detaylar (keskin değişiklikler) yüksek frekanslara uyduğu için, ortalama prosedürü veriyi yumuşatma eğilimindedir. Low-pass filtre, Haar durumunda

$$\frac{1}{\sqrt{2}} (1,1) \text{ şeklinde ifade edilir.}$$

Fark alma prosedürü verinin bir çeşit high-pass filtreden geçirilmesi gibi düşünülebilir. Düşük frekansları çıkarır ve detaylar yüksek frekanslara karşılık geldiği için bir görüntünün detaylarını oluşturur. Ayrıca bir görüntüde gürültüyü (noise) ortaya çıkarır. Çünkü gürültü, yüksek frekanslarda konumlanmıştır. High-pass filtre Haar durumunda,  $\frac{1}{\sqrt{2}}(1,-1)$  olarak ifade edilir.

Ortalama ve Fark alma prosedürlerine “analiz” de denir. Bunun tersi duruma ise “sentez” ismi verilir. Fourier dönüşümü gibi, dalgacık dönüşümleri de yüksek ve düşük frekansları ayırır [Sandberg Kristian,1998,s.1...14].

## 2.9. Görüntüyü Alt Bandlara Ayırıştırma (Subband Coding)

Görüntü sıkıştırmanın arkasında yatan temel fikir, bir sinyali frekans bandlarına ayırıştırma ve sonra her bir alt bandı kodlamaktır. Bu tekniğe subband coding denir. Bu durum temsili olarak Şekil 2.1 ‘de gösterilmiştir.

|           |           |       |
|-----------|-----------|-------|
| LL,<br>LL | LH,<br>LL | H , L |
| LL,<br>LH | LH,<br>LH |       |
| L , H     |           | H , H |

Şekil 2.1. Bir Görüntünün alt bandlara dalgacık çözümü ile ayrıştırılması  
[Bala Erdem,2001,s.31]

İki boyutlu (2-D) durum için görüntü ayrıştırmasının her adımında görüntü, bir öncekine göre ikiye bölünür.  $N \times N$  boyuttaki bir görüntüye sahip olduğu

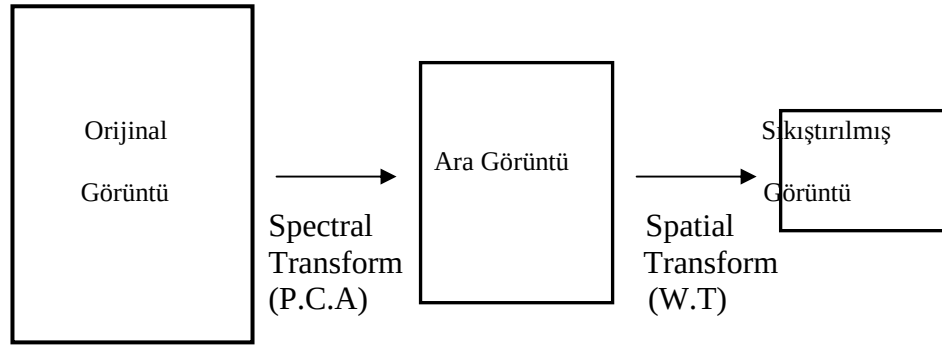
varsayılırsa, İlk olarak  $N$  satırın her biri low-pass ve high-pass parçalarına bölünür. Her bir kolon da low-pass ve high-pass parçalarına bölünür. Böyle bir ayrıştırmanın sonucu olarak,  $N/2 \times N/2$  boyutunda dört alt görüntü oluşur. Bu görüntüler; yatay (horizontal) low-pass / düşey (vertical) low pass, yatay (horizontal) high-pass / düşey (vertical) low-pass, yatay (horizontal) / düşey (vertical) high-pass ve yatay (horizontal) high pass / düşey (vertical) high-pass görüntülerdir.

Dönüşümün sonraki adımında yatay (horizontal) low-pass / düşey (vertical) low-pass parçaları (yaklaşma katsayıları) ve diğer parçalar (detay katsayıları) daha fazla analiz edilemeyecek duruma gelene kadar aynı tarzda ayrıştırılırlar. Şekil 2.1, herhangi bir görüntünün dalgacık ayrıştırmasındaki iki genel adımı gösterir [Engeny Krestyannikov, 2002, s.7].

## 2.10. Çok Spektrumlu Görüntü Sıkıştırma Dalgacıklar

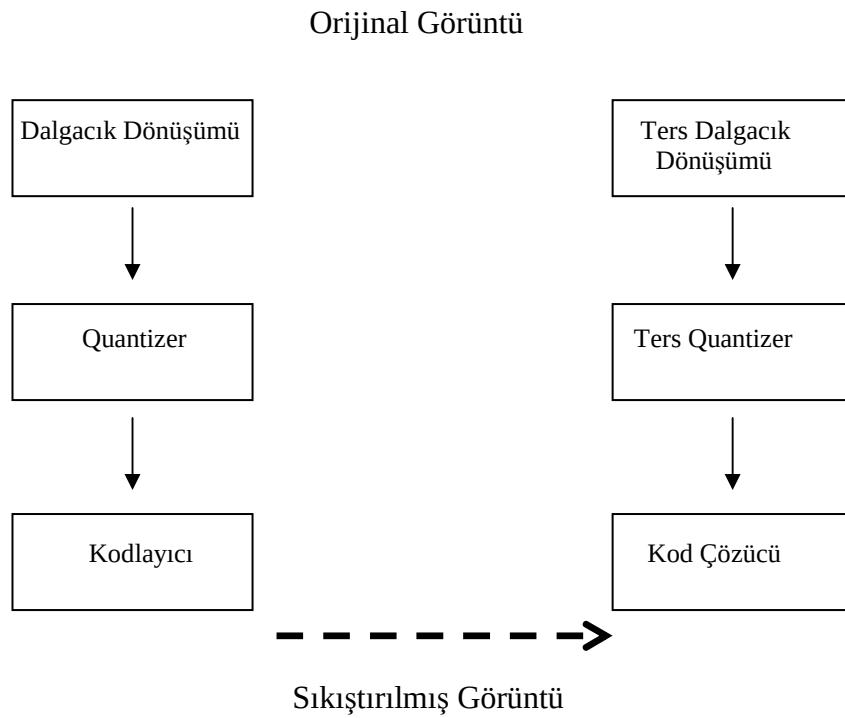
Bugün bilimin farklı alanlarında ve endüstride birçok spektral dalga boyundan bilgi içeren görüntülere başvurulur. Çok spektrumlu görüntüler genellikle yer bilimleri ve uzaktan algılamada (remote sensing) kullanılır. Uydulardan elde edilen her görüntüde yüzlerce spektral band çok büyük disk alanları gerektirir. Bu alanların boyutunu düşürmek için veriler sıkıştırılmalıdır [Evgeny Krestyannikov, 2002, s.11].

Çok spektrumlu görüntüleri sıkıştırmak için birkaç yaklaşım bulunmaktadır. Bu yaklaşımların her biri spektral ve spatial boyutlarda gereksiz çoklukları görüntüden kaldırır veya temizler. Spektral boyuttan kastedilen görüntünün hangi spektral bantlardan oluştuğudur. Spatial boyut ise, bu spektral değerlerin dijital ortama kayıt edilme şeklidir. Görüntülerin sıkıştırılmasında Principal Component Analysis (P.C.A) uygulanarak görüntüdeki gereksiz spektral değerler kaldırılır. Wavelet Transform (W.T) kullanılarak da görüntüdeki çok küçük ihmal edilebilecek spektral değerler sıfır yapılarak görüntü sıkıştırılır. Şekil 2.2, şematik olarak bu durumu göstermektedir [Evgeny Krestyannikov, 2002, s.11].



Şekil 2.2. Çok Spektrumlu Görüntünün Ayrıştırılması

Etkili şekilde kullanılan çok farklı dalgacık sistemleri vardır. Belirli uygulamalar için dalgacıkların seçiminde çok büyük sayıdaki esneklik, en uygun dalgacık kullanımına izin verir. Şekil 2.3 , ideal bir dalgacık dönüşüm şemasını göstermektedir [Mert Funda Yıldırım,1999,s.4].



Şekil 2.3. Dalgacık Dönüşümünün Yapısı. [Mert Funda Yıldırım,1999,s.5].

## 2.11. Örnek Uygulama

### 2.11.1. Ortalama ve Fark Alma Süreçleri

Ortalama ve fark alma (Averaging and Differencing) sürecini tanımlamada kolaylık olması için,  $8 \times 8$  piksel karelik görüntü matrisinin ilk satırı kullanılacaktır. Bu satır aşağıda gösterilmiştir. Görüntü matrisi  $8 \times 8$  piksel karelik bir matris olduğundan, sıkıştırma süreci üç ( $2^n = 8$ ,  $n = 3$ ) adımda gerçekleşecektir.

$$[ 3 \quad 5 \quad 4 \quad 8 \quad 13 \quad 7 \quad 5 \quad 3 ]$$

İlk adımda matris elemanlarının ikişerli gruplar halinde aritmetiksel ortalamaları alınır. Birinci elemanla ikincinin, üçüncüyle dördüncünün, beşinciyle altıncının ve yedinci ile sekizincinin toplamaları ikiye bölünür. Üretilen bu dört sonuç yine böyle bir matrisin ilk dört sütununa yazılır. Bu adıma Ortalama (Averaging) Süreci denir. Aynı şekilde birinci elemandan ikincinin, üçüncü elemandan dördüncünün, beşinci elemandan altıncının ve yedinci elemandan sekizincinin farkları alınıp ikiye bölünür. Çıkan dört yeni sonuç, yeni oluşan satır matrisinin son dört sütununa sırası ile yazılır. Bu işleme Fark Alma (differencing) Süreci denir. Birinci adımda üretilen sonuçlar aşağıda gösterilmiştir.

$$[ 4 \quad 6 \quad 10 \quad 4 \quad -1 \quad -2 \quad 3 \quad 1 ]$$

Yukarıda koyu olarak yazılmış matris elemanları, fark alma işlemi sonucu oluşmuştur. Bu dört elemana detay katsayıları (detail coefficient) denir.

İkinci adımda; birinci adımdan sonra oluşan matrisin son dört elemanı olan detay katsayılarına dokunulmaksızın ilk dört eleman üzerinde ilk adımdaki işlemlerin aynısı tekrarlanır. Birinci elemanla ikinci elemanın, üçüncü elemanla dördüncü elemanın kendi aralarında aritmetiksel ortalamaları alınır ve ilk iki sütuna yazılır. Birinci ve ikinci elemanın ve üçüncüyle dördüncü elemanın farklarının kendi

aralarında ikiye bölümü yapılır. Çıkan sonuçlar da üçüncü ve dördüncü sütuna yazılır. İlk adımda elde edilen detay katsayıları ise aynı şekilde geri taşınır. İkinci adımdan sonra oluşan matris aşağıda verilmiştir.

$$\begin{bmatrix} 5 & 7 & -1 & 3 & -1 & -2 & 3 & 1 \end{bmatrix}$$

Yukarıda koyu olarak yazılan elemanlar detay katsayılarıdır. Diğer iki eleman yaklaşım katsayısıdır.

Üçüncü ve son adımda, ortalama ve fark alma süreci sadece ilk iki elemana uygulanır. Birinci ve ikinci elemanın aritmetiksel ortalaması alınır. Çıkan sonuç ilk sütuna yazılır. Yine aynı elemanların farklarının ikiye bölümü hesaplanır. Çıkan sonuç ikinci sütuna yazılır. Birinci ve ikinci adımdan sonra oluşan detay katsayılarına dokunulmadan önceki adımlardaki gibi aynı şekilde matrise geri taşınır. Son adımdan sonra oluşan matris aşağıda gösterilmiştir.

$$\begin{bmatrix} 6 & -1 & -1 & 3 & -1 & -2 & 3 & 1 \end{bmatrix}$$

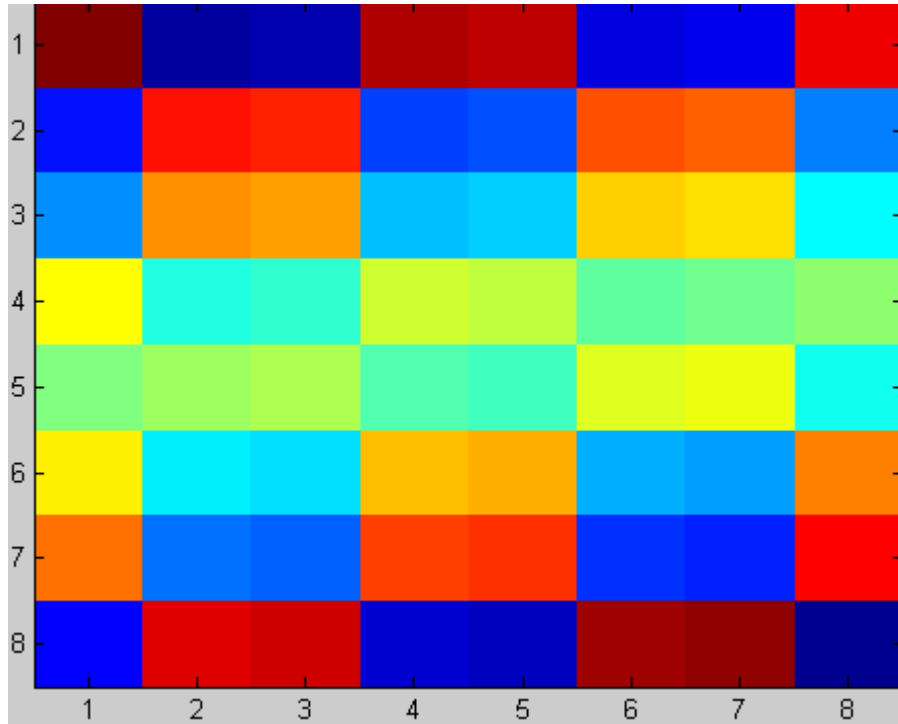
Yukarıda koyu renk ile yazılan elemanlar detay katsayılarıdır. İlk sütundaki eleman olan 6 sayısı ise yaklaşım sayısıdır. Son durum, yedi adet detay katsayısı tarafından takip edilen bir satır ortalamasıdır.

### 2.11.2. Temsili Görüntüyle Uygulama

Ortalama ve fark alma süreci, temsili olarak  $8 \times 8$  piksel karelik ufak bir görüntü matrisine yukarıda edinilen bilgiler ışığında uygulanabilir. Aşağıda verilen A matrisi Şekil 2.5 'deki görüntüyü temsil etmektedir. A matrisinde diğer sayılara göre daha büyük olan sayıların daha koyu renkleri temsil ettiği Şekil 2.4 ve Şekil 2.5 aracılığıyla görülebilir.

$$\begin{pmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 43 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{pmatrix}$$

Şekil 2.4. A matrisi



Şekil 2.5. A matrisinin görüntüsü

Eğer  $A$  matrisinin her bir satırına, ortalama ve fark alma süreçleri uygulanırsa, ilk kolon satır ortalamaları olur ve satırlardaki fark alma işleminden kaynaklanan detay katsayıları ise diğer kolonlarda bulunur. Bu işlemlerin sonucu, aşağıda görülmektedir.

$$\begin{pmatrix} 32.5 & 0 & .5 & .5 & 31 & -29 & 27 & -25 \\ 32.5 & 0 & -.5 & -.5 & -23 & 21 & -19 & 17 \\ 32.5 & 0 & -.5 & -.5 & -15 & 13 & -11 & 9 \\ 32.5 & 0 & .5 & .5 & 7 & -5 & 3 & -1 \\ 32.5 & 0 & .5 & .5 & -1 & 3 & -5 & 7 \\ 32.5 & 0 & -.5 & -.5 & 9 & -11 & 13 & -15 \\ 32.5 & 0 & -.5 & -.5 & 17 & -19 & 21 & -23 \\ 32.5 & 0 & .5 & .5 & -25 & 27 & -29 & 31 \end{pmatrix}$$

Eğer ortalama ve fark alma işlemleri kolonlara da uygulanırsa, aşağıdaki matris elde edilir.

$$\begin{pmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & .5 & .5 & 27 & -25 & 23 & -21 \\ 0 & 0 & -.5 & -.5 & -11 & 9 & -7 & 5 \\ 0 & 0 & .5 & .5 & -5 & 7 & -9 & 11 \\ 0 & 0 & -.5 & -.5 & 21 & -23 & 25 & -27 \end{pmatrix}$$

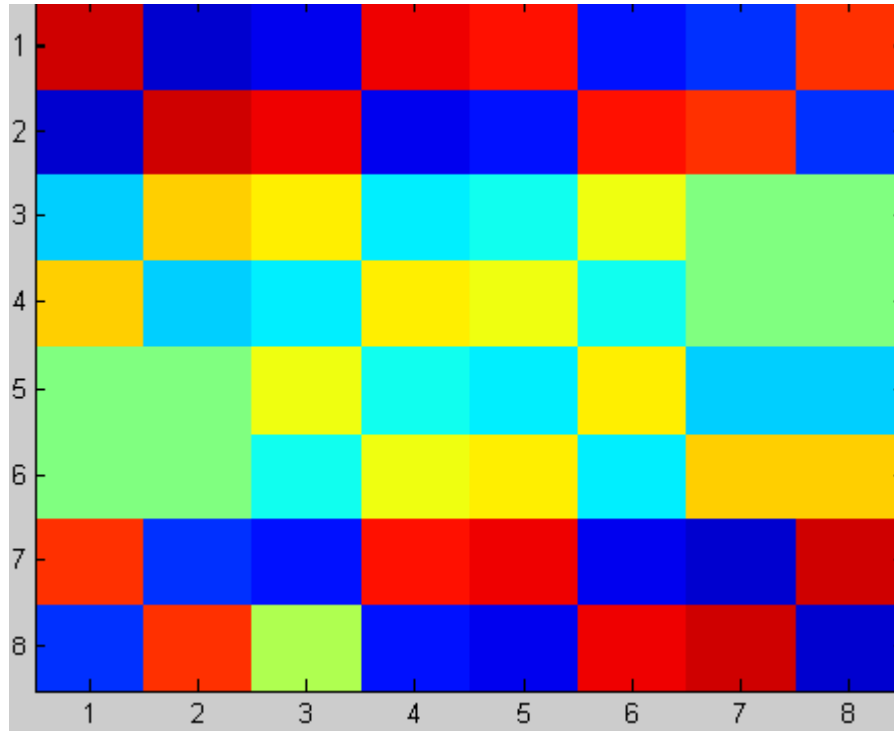
Yukarıda elde edilen matrisin sol üst köşesindeki 32.5 ortalaması, orijinal matrisimizi en öz tarzda temsil eder. Matriste geriye kalan diğer elemanlar, görüntüdeki bazı detayları temsil eden detay katsayılarıdır. Bu bilindiği için, matristen bazı gereksiz bilgiler elimine edilebilir. Bu işlemi yapmak için bir  $\delta$  sayısı seçilir ve büyüklüğü  $\delta$  'dan daha küçük olan değerler 0 (sıfır) yapılır.  $\delta$  'nın 5 ( $\delta = 5$ ) seçilmesiyle 18 eleman sıfırlanır. Aşağıda koyu renk yazı ile yazılan elemanlar  $\delta$  'nın 5 olarak alınmasıyla sıfırlanan elemanlardır.

$$\begin{pmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 27 & -25 & 23 & -21 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & -11 & 9 & -7 & \mathbf{0} \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 7 & -9 & 11 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 21 & -23 & 25 & -27 \end{pmatrix}$$

Orijinal görüntünün iyi bir yaklaşımını almak için ortalama ve fark alma işlemlerinin tersi, yukarıdaki matrise uygulanırsa aşağıdaki matris elde edilir.

$$\begin{pmatrix} 59.5 & 5.5 & 7.5 & 57.5 & 55.5 & 9.5 & 11.5 & 53.5 \\ 5.5 & 59.5 & 57.5 & 7.5 & 9.5 & 55.5 & 53.5 & 11.5 \\ 21.5 & 43.5 & 41.5 & 23.5 & 25.5 & 39.5 & 32.5 & 32.5 \\ 43.5 & 21.5 & 23.5 & 41.5 & 39.5 & 25.5 & 32.5 & 32.5 \\ 32.5 & 32.5 & 39.5 & 25.5 & 23.5 & 41.5 & 21.5 & 21.5 \\ 32.5 & 32.5 & 25.5 & 39.5 & 41.5 & 23.5 & 43.5 & 43.5 \\ 53.5 & 11.5 & 9.5 & 55.5 & 57.5 & 7.5 & 5.5 & 59.5 \\ 11.5 & 53.5 & 35.5 & 9.5 & 7.5 & 57.5 & 59.5 & 5.5 \end{pmatrix}$$

Yukarıdaki matris uygun bir program yardımıyla bilgisayar ortamında görüntülenirse, orijinal görüntüye yakın bir görüntü elde edilir. Matlab 6.5 programı yardımıyla yukarıdaki yaklaşım matrisi Şekil 2.6 'da görüntülenmiştir.



Şekil 2.6. Yaklaşım Matrisinin Görüntüsü.

### 2.11.3. Lineer Cebir Kullanarak Ortalama ve Fark Alma

Lineer cebir kullanarak, yukarıda bahsedilen ve boyutları  $8 \times 8$  piksel kare olan görüntünün ortalama ve fark alma işlemleri, bilgisayar ve programlama tekniklerinin gelişimine paralel olarak daha basit bir şekilde birkaç matrisin birbirleri ile çarpımı sonucunda elde edilebilir. Görüntü  $8 \times 8$  piksel kare boyutlarında olduğundan ( $2^n = 8$  ve  $n = 3$ ), uygulama üç adımda gerçekleştirilir. Dolayısıyla üç ayrı matris kullanılır. Bu işlemler için kullanılacak matrisler  $A1, A2, A3$  matrisleridir.

Birinci adımda, görüntü matrisinin ilk satırı  $A1$  matrisi ile çarpılır. Bu matris görüntü matrisinin ilk satırının her bir çiftinin ortalamasını alır ve yeni oluşan satır matrisinin ilk dört sütununa bu dört ortalamayı yazar. Daha sonra yine aynı satıra fark alma prosedürünü uygular ve yeni oluşan satırın son dört sütununa bu farkların ikiye bölümünü yazar. Yukarıda bahsedilen süreç ve ürünü aşağıda gösterilmiştir.

$$= [3 \quad 5 \quad 4 \quad 8 \quad 13 \quad 7 \quad 5 \quad 3] \cdot \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

$$= [4 \quad 6 \quad 10 \quad 4 \quad -1 \quad -2 \quad 3 \quad 1]$$

İkinci adımda,  $A2$  matrisi aynen birinci adımdaki  $A1$  matrisi gibi çalışır ve önceki adımdan oluşan matrisin ilk dört elemanının ortalama ve fark alma işlemlerini gerçekleştirir. İlk dört elemanın ikiye bölünmüş çiftler halinde aritmetiksel ortalamalarını alır ve oluşan sonuçları yeni oluşan matrisin ilk iki sütununa yazar. Daha sonra fark alma işlemini gerçekleştirir ve oluşan sonuçları yeni oluşan matrisin üçüncü ve dördüncü sütununa yazar. İlk adımda oluşan matrisin son dört elemanına da hiçbir işlem gerçekleştirilmeden, olduğu gibi yeni oluşan matrise taşır. İkinci adım ve oluşan sonuçlar aşağıda gösterilmiştir.

$$=[4 \quad 6 \quad 10 \quad 4 \quad -1 \quad -2 \quad 3 \quad 1] \cdot \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$=[5 \quad 7 \quad -1 \quad 3 \quad -1 \quad -2 \quad 3 \quad 1]$$

Üçüncü adımda  $A3$  matrisi, en son oluşan satır matrisi ile çarpılır.  $A3$  matrisi de yine  $A1$  ve  $A2$  matrisleri gibi çalışır. Son adım ve bu adımın ürünü aşağıda gösterilmiştir.

$$=[5 \quad 7 \quad -1 \quad 3 \quad -1 \quad -2 \quad 3 \quad 1] \cdot \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$=[6 \quad -1 \quad -1 \quad 3 \quad -1 \quad -2 \quad 3 \quad 1]$$

Bu üç adım ayrı ayrı uygulanmak yerine tek bir adımda gerçekleştirilebilir. Tek bir transformasyon matrisi;  $A1$ ,  $A2$  ve  $A3$  matrislerinin birbirleri ile çarpımı sonucu oluşturulur ve oluşan bu dönüşüm matrisi ile görüntü matrisinin ilk satırı direkt olarak çarpılarak ortalama ve fark alma işlemleri tek bir adımda gerçekleştirilebilir. Bu süreç aşağıda gösterilmiştir.

$$=[3 \quad 5 \quad 4 \quad 8 \quad 13 \quad 7 \quad 5 \quad 3] \cdot \begin{pmatrix} 1/8 & 1/8 & 1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 0 & -1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & -1/2 & 0 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & 1/2 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & -1/2 & 0 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & 1/2 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

$$=[6 \quad -1 \quad -1 \quad 3 \quad -1 \quad -2 \quad 3 \quad 1]$$

Aşağıdaki eşitlikler ortalama ve fark alma prosedürlerini ve daha sonra bu operasyonların tersini uygulamayı kolaylaştıran matris çarpımlarını göstermektedir.

$$\begin{aligned}
 1. \quad T &= \left( (A.W)^T . W \right)^T \\
 &= \left( W^T . A^T . W \right)^T \\
 &= W^T . (A^T)^T . (W^T)^T \\
 &= W^T A W
 \end{aligned}$$

$$\begin{aligned}
 2. \quad (W')^{-1} . T . W^{-1} &= A \\
 (W^{-1})' . T . W^{-1} &= A
 \end{aligned}$$

Yukarıdaki formüllerde  $A$ , görüntü matrisini;  $W$ , transformasyon matrisini;  $T$ , transforme olmuş görüntü matrisini temsil etmektedir. Bu formüller, yukarıda  $8 \times 8$  piksel karelik görüntüler için uygulandı. Ancak bağıntılar, daha büyük boyutlu görüntüler için de uygulanabilir. Bu matrisler boyut olarak bir kenarı ikinin tam katı olan kare şeklindeki tüm görüntüler için uygulanabilir [Morton Peggy and Petersen Arne, 1997, s.1.6).

Daha önce bahsedildiği gibi gerekli transform matrisleri ile çarpılan bir görüntü, ortalama ve fark alma (averaging and differencing) işlemlerinden geçer. Bu durumda görüntü, ayrıntı katsayılarına (detail coefficients) ayrışır. Bunlar, görüntüdeki ayrıntıları temsil ederler. Ayrıntı katsayılarına ayrıştırılmış görüntüyü  $T$  matrisi temsil eder. Bu matris içindeki diğer elemanlara göre ihmal edilebilecek kadar küçük değerler sıkıştırma katsayısı  $\delta$  'ya göre sıfırlandıktan sonra ortalama ve fark alma prosedürlerinin tersi aynı matrise uygulanırsa yeni bir  $A$  görüntü matrisi

elde edilir. Bu matris uygun bir bilgisayar programı sayesinde görüntülenirse, orijinal görüntüye yakın bir görüntü elde edilir.

Yeni görüntünün orijinal görüntüye yakınlığı seçilecek  $\delta$  sıkıştırma katsayısının büyüklüğü ile doğru orantılıdır.  $\delta$  ne kadar büyük olursa görüntü o denli sıkışır ve netlik azalır.

Haar Wavelet Dönüşümünü kullanan görüntü sıkıştırma uygulamalarında yukarıda bahsedilen özellikten faydalanılır.  $\delta$  sayısının büyüklüğüne göre  $T$  matrisinden o denli eleman sıfırlanır. Sıfırlanan bu elemanların bilgisayarda saklanmasına gerek yoktur. Sadece sıfırdan farklı elemanlar bilgisayarda uygun bir program yardımıyla uygun bir formatta saklanırsa, görüntü sıkıştırılmış olur. Buna eşdeğer bir anlatımla; görüntü sıkıştırma işlemi,  $T$  matrisi içinde ihmal edilebilecek kadar küçük değerdeki ayrıntı katsayılarının sıfırlanıp uygun bir program vasıtasıyla dijital ortamda saklanması işlemi diye tanımlanabilir [Morton Peggy and Petersen Arne, 1997, s.6..9).

### 3. DİJİTAL GÖRÜNTÜLER

Dijital Görüntülerin kaynakları, dijite edilmiş fotoğraflardan dijital kameralara ve elektro optik tarayıcılara kadar uzanır [Lillesand Thomas M. and Kiefer Ralph W., 1994, s.339]. Bu görüntüler, bilgisayar ortamına RGB renk sistemine göre kodlanır. RGB renk sisteminin açık yazılımı Red-Green-Blue ‘dur. Yani; kırmızı, yeşil ve mavi ışık tayflarının oluşturduğu üçlü bir renk sistemidir. Bu ışık tayflarının her birine band denir. İnsan gözü, bu üç bandtan gelen görüntüleri algılayabilir. Orthofoto görüntüler, bu üç bandtan gelen ışınlarla oluşturulan görüntülerdir. Ancak; insan gözünün algılayamadığı bandlar da mevcuttur.

Genellikle uydu görüntüleri üçten çok daha fazla bandı kapsayan görüntüler içerirler. Bu ekstra bandların amacı, bitki yapısını belirleme, jeolojik oluşumları izleme, petrol kaynaklarını belirleme, meteorolojik olaylar hakkında yorumlar yapma, çevre kirliliğini araştırma gibi bilimsel çalışmalar yapmaktır. Şekil 3.3, şematik olarak yukarıda bahsedilen elektromanyetik spektrumu (elektromanyetik tayf) göstermektedir.

Dijital bir görüntü bilgisayar ortamında bu üç bandtan elde edilen renk karışımı ile saklanır. RGB renk sisteminde red (kırmızı); şiddeti en az (dalga boyu en büyük), blue (mavi), şiddeti en yüksek (dalga boyu en düşük) olan bandlardır. Bilgisayar ortamı içinde 24 bitlik bir görüntüde en yüksek sekiz bit maviyi, ortadaki sekiz bit yeşili ve en sondaki sekiz bit ise kırmızı rengi temsil etmektedir. Örneğin;

|                 |                 |                 |
|-----------------|-----------------|-----------------|
| <u>10101110</u> | <u>10011101</u> | <u>01110001</u> |
| Mavi            | Yeşil           | Kırmızı         |

şeklinde bir 24 bitlik ikilik tabana göre kodlama durumunda; en yüksek 8 bit onluk tabana göre 174 değerinde mavi renk tonunu, ortadaki 8 bit, onluk tabana göre 157 değerinde yeşil renk tonunu, en düşük 8 bit ise onluk tabana göre 113 değerinde kırmızı renk tonunu temsil etmekte ve bu üç rengin karışım tonu RGB(113,157,174) şeklinde gösterilmektedir.

Eğer bu 24 bitlik değer bilgisayarın en küçük boyutlu görüntü birimi olan bir piksele yansıtılırsa, bahsedilen piksel bu üç rengin karışımı olan bir renk alacaktır. Eğer sadece tek bantlar ile çalışılıp, tek bandın değeri RGB formülünde yerine konulup diğer bantlar için sıfır yazılırsa ve bu değer bir piksele gönderilirse, bahsedilen piksel o bandın tonlarında görülür.

Örneğin; RGB(240,30,70) şeklinde bir renge düşünülürse, R = 240, G = 30 ve B=70 'dir. Eğer RGB(240,0,0) yazılırsa, piksel görüntüsü kırmızının bir tonu şeklinde, RGB(0,30,0) yazılırsa, piksel görüntüsü yeşilin bir tonu şeklinde, RGB(0,0,70) yazılırsa, piksel görüntüsü mavinin bir tonu şeklinde görünecektir. Eğer RGB(240,240,240), RGB(30,30,30) veya RGB(70,70,70) yazılırsa, aynı piksel, gri bir renk tonu şeklinde görüntülenecektir.

Yukarıda anlatılanlar ile görüntüler, bandlarına bir takım matematiksel işlemler ile ayrıştırılabilir ve tezin ilk bölümlerinde bahsedilen spektral transformasyon gerçekleştirilebilir.

Yukarıda bahsedilen dijital görüntülere, orthofoto görüntüler ve tarayıcı vasıtasıyla dijital ortama aktarılan grafik kadastral pafta görüntüleri örnek olarak gösterilebilir.

### 3.1. Orthofoto Görüntüler

Orthofoto görüntüler, orthofotoğrafik fotoğraflardır. Normal hava fotoğrafları gibi ölçek, eğilme ve kabartma hataları içermezler. Kısaca orthofotolar, foto haritalardır. Haritalar gibi bir ölçeğe sahiptirler ve fotoğraflar gibi gerçek detaylarıyla araziye gösterirler. Fakat çizgiler ve semboller kullanmazlar. Bundan dolayı orthofotolar, kaynak araştırmacılar için bir fotoğraf gibi çabuk yorumlanan bir üründür. Fakat doğru mesafeler, açılar ve alanlar direk olarak ölçülemez.

Bu özelliklerden dolayı orthofotolar, G.I.S (Geographic Information System) veya Coğrafi Bilgi Sistemine (C.B.S) girilecek derleme veriler için bir altlık oluşturur veya Coğrafi Bilgi Sisteminde önceden yorumlanan tabakalama ve

editleme verilerini oluşturur. Ayrıca orthofotolar, konumsal (spatial) verinin iletimini artırır. Çünkü veriyi kullananlar; basmakalıp bir çizgi, sembol haritası veya görüntüsündense bir orthofoto ile daha iyi ilişki kurarlar.

Orthofotolar; analog, hardcopy ve dijital orthofotolar olmak üzere üçe ayrılırlar. Orthofotolar, artan bir şekilde softcopy formatında üretilmiş ve kullanılmıştır.

Dijital orthofotolar, Coğrafi Bilgi Sisteminde görüntü yorumlama ve özellik kodlama için ideal görüntü tabanlarını oluşturur. Dijital orthofotolar sadece Coğrafi Bilgi Sistemi verilerini bir arada tutmazlar, ayrıca böyle verileri görüntüleme ve güncelleme için bir taban oluştururlar.

Orthofotolar, diferansiyel rektifikasyon diye adlandırılan bir süreç içinde, çakışan veya üst üste binen basmakalıp fotoğraflardan üretilirler. Bu sürecin amacı, foto ölçek farklılığını gidermek, engebe ve eğilme ile sonuçlanan görüntü kaymalarını önlemektir.

Orthofotolar, stereoplotterlarda üretilen haritalarla aynı tarzda olacak şekilde orthofotoskop diye isimlendirilen aletlerle hazırlanırlar. Ancak; baz olarak alınan bir harita üzerinde stereomodeldeki seçilen çizim özellikleri yerine, stereomodelin tüm noktaları bir orthofotonegatif üzerine fotoğraflanır. Bu, daha sonra orthofotoğrafın çizimi için kullanılır.

Orthofotolar yalnız başına topoğrafik bilgi içermezler. Ancak ayrı bir stereoplotting işlemi ile hazırlanan kontur hat planları için bir baz harita olarak kullanılır. Orthofoto görüntü üzerine kontur bilgilerinin işlenmesi topoğrafik bir ortofotoğraf oluşturur [Lillesand Thomas M. and Kiefer Ralph W., 1994, s.335..340]. Şekil.3.1, iki adet orthofoto görüntüyü göstermektedir.

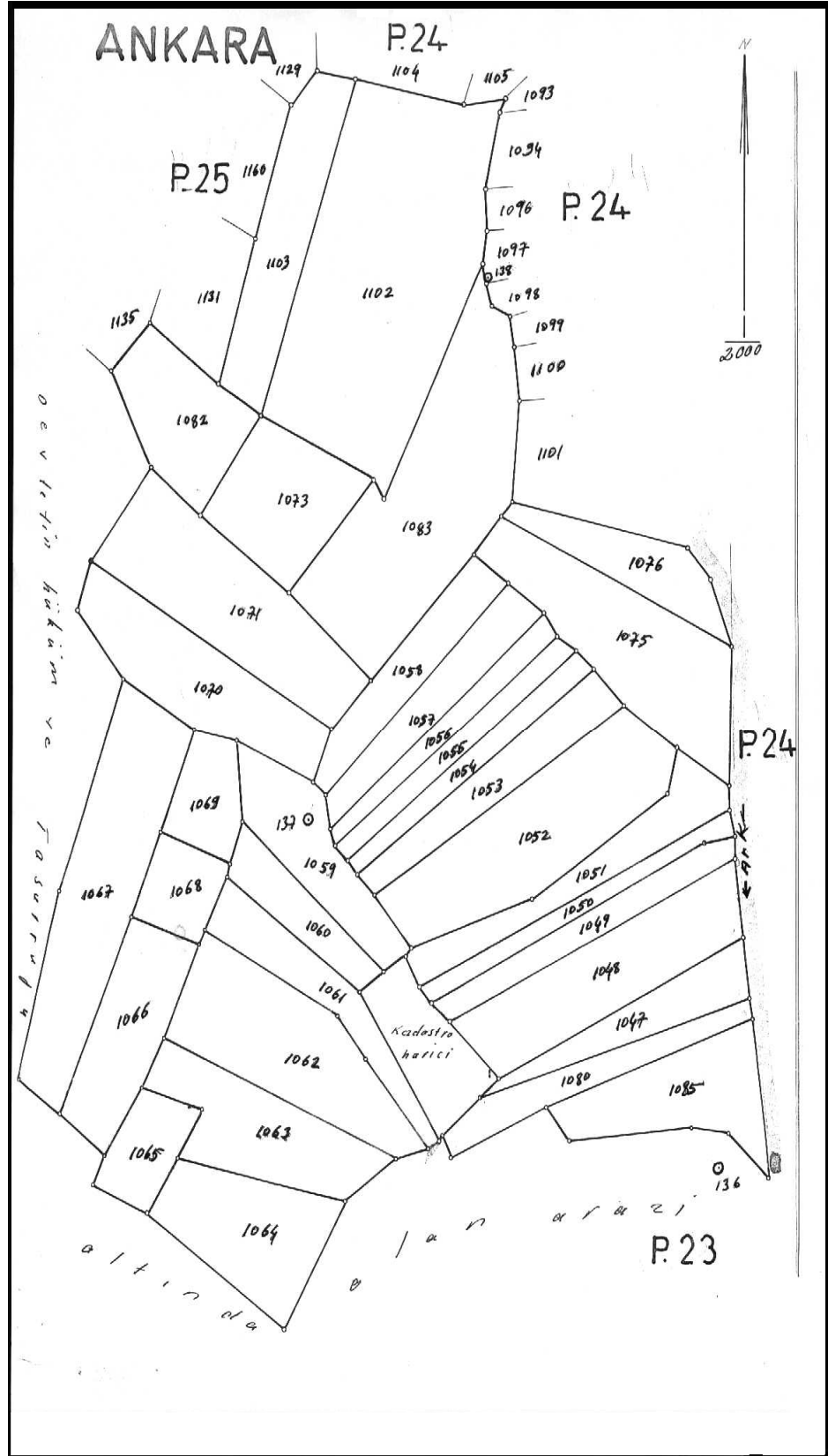


Şekil 3.1. 256x256 Piksellik (192 kilobayt) Orthofoto Görüntü Örnekleri

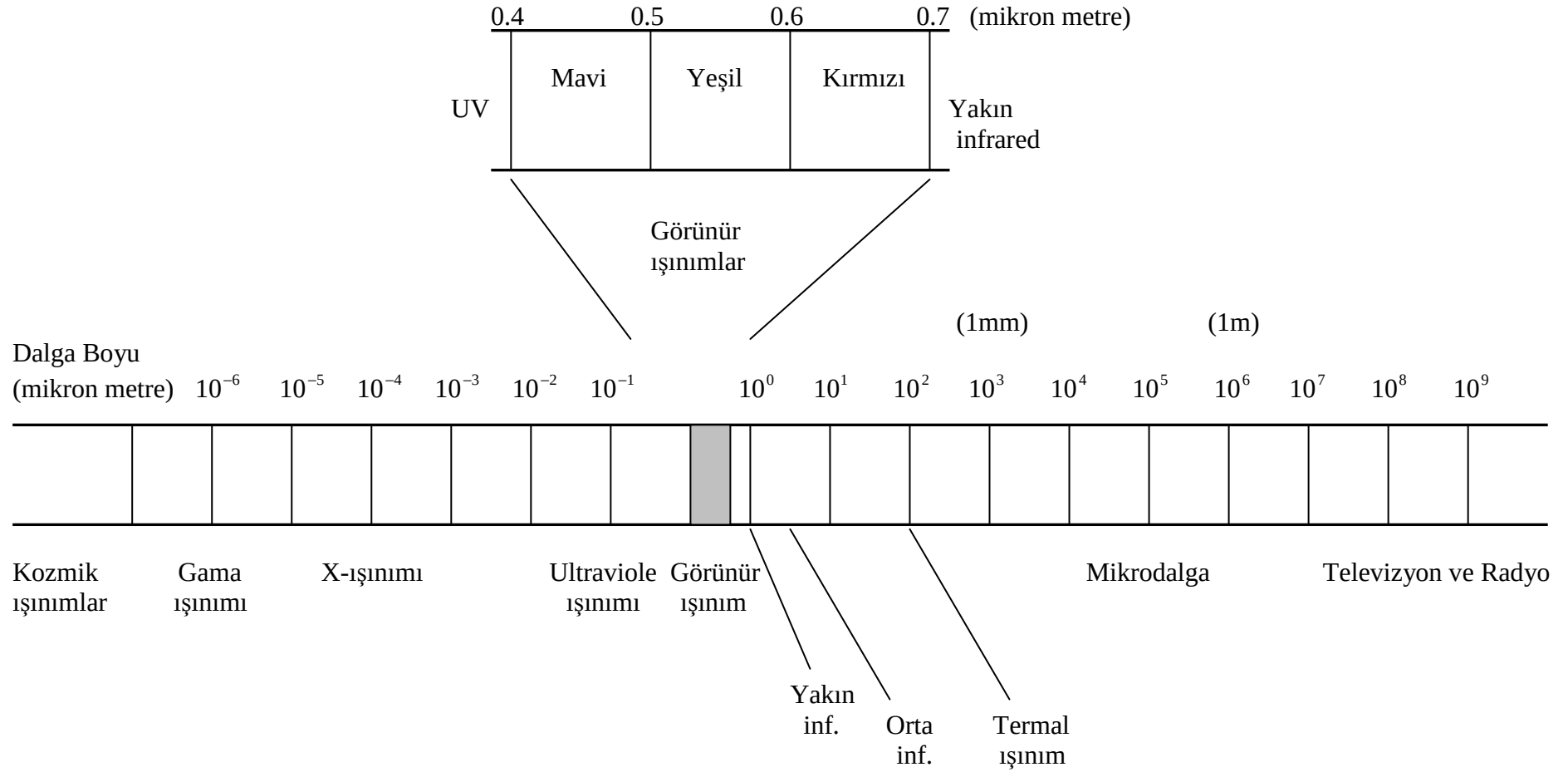
### 3.2. Grafik Kadastral Paftalar

Ülke Koordinat Sistemine dayanmayan uygulandığı zamanın teknik koşullarına göre hazırlanmış paftalardır. Genellikle köylerde yapılan kadastro çalışmalarında paftalar bu yöntemle oluşturulmuştur. Bu paftalar hazırlanmadan önce, kadastronu yapılacak olan alana gerekli poligonlar atılıp bu poligonlar vasıtası ile açı ve kenar ölçülmek koşuluyla grafik paftalar oluşturulmuştur. Paftalarda koordinat değerleri bulunmamaktadır. Her çalışma alanı mevzii olarak ölçülüp paftasına işlenmiştir.

Grafik paftalarda ölçek genellikle 1/1000, 1/2000, 1/2500 olmaktadır. Şekil 3.2, Ankara 'nın Nallıhan İlçesi'nin bir köyünde yapılan grafik kadastral bir paftayı göstermektedir. Bu pafta orijinal altlıktan tarayıcı (scanner) vasıtasıyla taranarak elde edilmiştir.



Şekil 3.2. 1790 x 1280 Piksellik (6721 kilobayt) Grafik Kadastral Pafta Örneği.



Şekil 3.3. Elekteromanyetik Spektrum [Lillesand Thomas M. and Kiefer Ralph W., 1994, s.5]

## 4. GÖRÜNTÜ SIKIŞTIRMA PROGRAMI

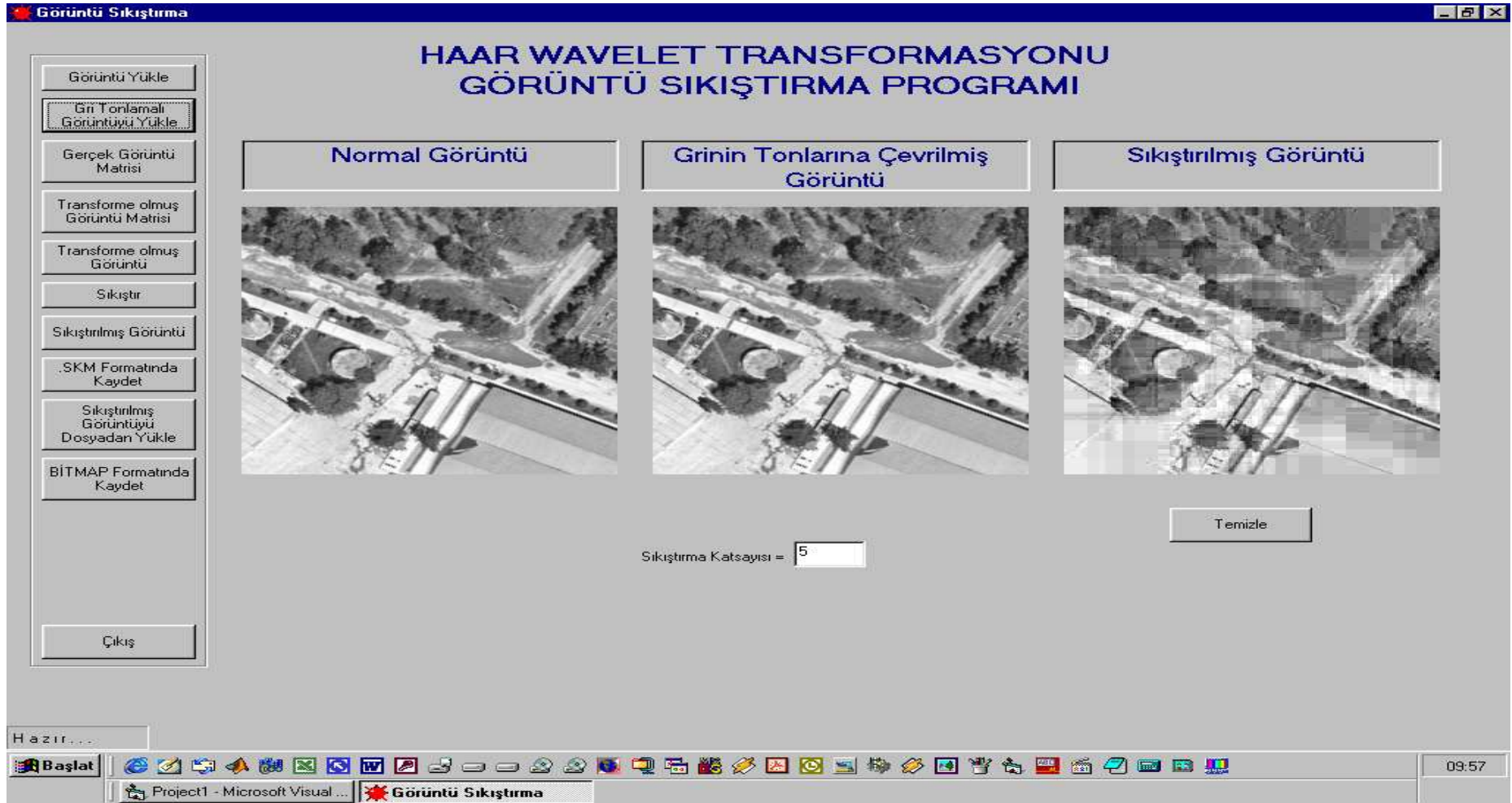
Haar Wavelet Görüntü Sıkıştırma Programı, Visual Basic 6.0 ile geliştirildi. Program, bitmap (.BMP) uzantılı  $256 \times 256$  piksel karelik herhangi bir kare boyutlu görüntüyü  $\delta$  sıkıştırma katsayısına bağlı olarak sıkıştırıp, .SKM formatında (programın ürettiği bir format) bilgisayar ortamında saklama özelliğine sahiptir. Ayrıca; sıkıştırılmış (.SKM uzantılı) bir dosyayı açıp, tekrar üzerinde çalışma yapılabilecek .BMP formatında kayıt etme özelliği de programda bulunmaktadır.

### 4.1. Programın Arayüzü

Program, Visual Basic programlama dilinde ‘Command Button’ ismi verilen 12 butona sahiptir. Bu butonlar Şekil 4.1 ‘de görüldüğü üzere yukarıdan aşağı doğru “Görüntü Yükle”, “Gri Tonlamalı Görüntüyü Yükle”, “Gerçek Görüntü Matrisi”, “Transforme Olmuş Görüntü Matrisi”, “Transforme Olmuş Görüntü”, “Sıkıştır”, “Sıkıştırılmış Görüntü”, “.SKM Formatında Kaydet”, “Sıkıştırılmış Görüntüyü Dosyadan Yükle”, “Bitmap Formatında Kaydet”, “Çıkış” ve en sağ alt köşedeki “Temizle” butonlarıdır. Programda ayrıca; yüklenen görüntülerin niteliğini kullanıcıya tanıtmak için Visual Basic ‘te ‘Label’ adı verilen 3 adet etiket, sıkıştırma katsayısını belirlemek için ‘Text Box’ adı verilen 1 adet text kutucuğu ve sıkıştırma oranını belirlemek, programın çalıştığı anda hangi işlevi yerine getirdiğini göstermek amacıyla ‘Status Bar’ adı verilen durum çubuğu kullanılmıştır.

Yukarıda bahsedilenler dışında, görüntüleri ekrana yansıtabilmek için 1 adet ‘PictureBox’ (Görüntü Kutusu) ve 2 adet ‘Picture Box’ (Resim Kutusu), Programın hangi aşamada işlediğini görmek için 1 adet ‘Status Bar’ (Durum Çubuğu) ve sıkıştırma sürecini izleyebilmek için 1 adet ‘Progress Bar’ (İlerleme Çubuğu) kullanılmıştır.

Ayrıca; dosya açıp kapama ve kaydetme işlevleri için Visual Basic’te ‘Common Dialog’ adı verilen 2 adet iletişim komponenti kullanılmıştır.



Şekil 4.1. Programın Arayüzü

Program, ana ve yardımcı olmak üzere iki forma sahiptir. Ana form, programın arayüzünü, diğer form ise üzerinde görüntü matrislerinin elemanlarını gösteren, Visual Basic'te 'msFlexGrid' adı verilen komponenti barındırır.

## **4.2. Programdaki Butonların Görevleri**

Daha önce bahsedildiği gibi programda 12 buton bulunmaktadır. Bu alt bölümde butonların programdaki işlevleri anlatılacaktır.

### **4.2.1. Görüntü Yükle Butonu**

Bu butona mouse ile tıklandığı zaman, dialog formu ekranda görülür. Formun en üstünde form başlığında “Bir Resim Dosyası Seçiniz” ibaresi yazar, formun en altındaki text kutusunda, dosya filitresi kısmında ise “Bitmap Dosyaları” yazmaktadır. Bu ibarenin anlamı, bu program vasıtası ile sıkıştırılmamış bir resim dosyası tipi olan bitmap formatındaki dosyaların sadece sıkıştırılacağıdır. Sıkıştırılması istenen dosya hangi dizin içinde ise bulunur, mouse ile seçilir ve ‘Aç’ butonuna basılırsa, bitmap uzantılı dosya, programdaki ‘imagebox’ vasıtası ile görüntülenir.

### **4.2.2. Gri Tonlamalı Görüntüyü Yükle Butonu**

Bu butona mouse ile tıklandığı zaman, daha önceki bölümlerde bahsedildiği gibi orijinal görüntü spektral bir transformasyona uğratarak, sıkıştırılması için ilk adım olan gri tonlamalı görüntü üretilir ve bir picturebox yardımı ile programda görüntülenir.

### **4.2.3. Gerçek Görüntü Matrisi Butonu**

Bu butona basıldığında, Form2 üzerine yerleştirilmiş msFlexGrid bileşeni sayesinde, orijinal görüntünün bilgisayar ortamında RGB renk sistemine göre

saklanan her bir pikselinin sayısal değeri ekranda matris şeklinde görüntülenir. msFlexGrid sayesinde görüntülenen matrisin görüntü boyutlarını yansıtan 256×256 piksel kare boyutlarında bir matris olduğu görülür. Matrisin her bir elemanı, görüntünün her bir elemanı ile eşleşir. Şekil 4.2 ‘de dönüşüme uğramış görüntü matrisi msFlexGrid sayesinde görüntülenmiştir.

Matris Gösterme Formu

Sayısal Olarak Basitleştirilmiş Görüntü Matrisi

|    | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   |  |
|----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| 1  | 152 | -14 | 0   | 11  | 0   | -6  | -12 | 10  |  |
| 2  | 0   | 12  | 0   | 0   | 0   | 7   | 0   | -7  |  |
| 3  | 13  | 0   | 0   | 14  | 0   | 0   | 14  | 0   |  |
| 4  | -10 | 0   | 15  | 0   | 0   | 12  | -6  | 0   |  |
| 5  | -13 | -12 | 16  | 0   | 0   | 7   | -23 | 0   |  |
| 6  | 8   | 0   | 0   | 0   | 18  | -19 | 0   | 0   |  |
| 7  | 0   | 11  | 7   | 0   | 8   | 0   | 0   | -9  |  |
| 8  | -6  | 0   | 11  | 0   | 0   | 7   | 0   | 8   |  |
| 9  | -12 | -11 | 16  | -13 | 10  | 0   | -12 | 10  |  |
| 10 | 0   | 0   | -9  | 0   | -10 | 0   | -7  | 0   |  |
| 11 | 10  | 0   | 0   | 9   | -6  | -20 | 0   | 0   |  |
| 12 | 0   | 0   | -13 | 0   | 22  | 16  | 0   | 0   |  |
| 13 | 0   | 0   | 0   | -6  | -19 | 0   | 0   | 0   |  |
| 14 | 0   | 0   | 0   | -7  | 0   | 0   | 0   | -15 |  |
| 15 | -7  | 7   | 8   | 10  | 0   | 0   | 0   | 8   |  |
| 16 | 6   | -10 | 0   | 0   | 0   | 0   | 0   | 16  |  |
| 17 | -7  | 0   | 7   | -9  | 16  | -9  | -9  | 0   |  |
| 18 | 0   | -6  | 10  | -6  | 0   | 0   | -8  | 0   |  |
| 19 | 0   | 0   | -6  | 0   | 0   | 0   | -8  | 0   |  |
| 20 | 0   | 0   | 0   | 0   | -10 | 0   | 6   | 0   |  |
| 21 | 0   | 0   | 0   | 0   | 0   | 14  | 0   | 0   |  |
| 22 | 0   | 0   | 0   | 0   | 0   | -22 | 0   | 0   |  |
| 23 | 0   | 0   | -13 | 0   | 14  | 14  | 0   | 0   |  |
| 24 | 0   | 0   | 0   | 0   | 0   | 8   | 0   | 0   |  |
| 25 | 0   | 0   | 0   | 0   | -31 | 0   | -11 | 0   |  |
| 26 | 0   | 0   | 0   | 0   | 6   | 0   | 0   | 0   |  |
| 27 | 0   | 0   | 0   | 0   | 6   | 0   | 0   | -16 |  |

Şekil 4.2. Programın msFlexGrid Bileşeni

#### 4.2.4. Sıkıştır Butonu

Bu butona basıldığında, orijinal görüntünün sıkıştırılma aşaması başlamış olur. Programın ara yüzünün alt kısmında bir progressbar görüntülenir. Bu bileşen yardımı ile görüntünün sıkıştırılma süreci takip edilebilir. Ayrıca statusbarda da “Sıkıştırılıyor...” ibaresi görülür. İşlem tamamlanınca progressbar kaybolur, status

barda sıkıştırılma oranı yazar ve en sağdaki picturebox 'ta sıkıştırılmış görüntü belirlir.

#### 4.2.5. Sıkıştırılmış Görüntü Butonu

Bu buton yardımı ile o anda üzerinde çalışılan görüntünün sıkıştırılması ile elde edilen sıkıştırılmış görüntünün en sağdaki picturebox yardımı ile yeniden çizilmesi sağlanır. Programa bu butonun konulmasının nedeni : program çalışırken, çalışmakta olan başka bir programın formunun, sıkıştırılmış görüntü üzerine gelmesidir. Bu butona basılarak sıkıştırılmış görüntünün programımızdaki picturebox 'a yeniden çizilmesi sağlanır.

#### 4.2.6. .SKM Formatında Kaydet Butonu

Bu butona basıldığı zaman, ekranda “Kaydet” dialog formu görünür. Bu formda istenilen dizine geçilerek en alttaki textbox yardımı ile dosya, hangi isimle kaydedilecekse o isim yazılır ve ‘Kaydet’ butonuna basılarak sıkıştırılmış görüntü .SKM uzantılı olarak seçilen dizin içine kaydedilir.

#### 4.2.7. Transforme Olmuş Görüntü Butonu

Bu buton yardımı ile segmentlerine ayrıştırılmış görüntü matrisi (transforme olmuş görüntü matrisi), programın ara yüzünün ortasındaki picturebox vasıtasıyla görüntülenir. Bu görüntüye, transforme olmuş veya segmentlerine ayrıştırılmış görüntü denir. Oluşan bu matris  $\delta$  'dan daha küçük değerler sıfır yapılmadan önceki matristir. Özetle; .SKM formatında kayıt edilen görüntünün bir basamak öncesidir. Eğer bu matrizen  $\delta$  'dan daha küçük değerler ihmal edilerek saklanırsa, görüntü sıkıştırılmış olur ve sıkıştırılmış görüntü elde edilir. Programda kullanılacak görüntüler gri rengin tonlarına çevrildiğinden dolayı, bu butona basmakla, elde edilen görüntülerdeki görüntü ayrıştırması çok silik bir şekilde görüntülenecek, hatta bu görüntü siyah renkte boş bir görüntü gibi gözükecektir. Şekil 4.3, tez projesinin geliştirilmesi aşamasında yapılan ilk programlardan birinin işletilmesi sonucunda

elde edilmiştir. Özetle; spektral transform yapılmadan önceki bir insan fotoğrafının segmentlerine ayrıştırılmış şeklini temsil eder.



Şekil 4.3. Segmentlerine Ayrıştırılmış Bir İnsan Görüntüsü

#### 4.2.8. Transforme Olmuş Görüntü Matrisi Butonu

Bu butona basıldığı zaman, .SKM formatında kaydedilen görüntünün matris temsili görüntülenir. Yani; bu matriste  $\delta$  'dan daha ufak değerler sıfır yapılmıştır. Elde edilen bu matris bilgisayar ortamında .SKM formatında saklanır. Bu şekilde görüntü sıkıştırılmış olur.

#### 4.2.9. Sıkıştırılmış Görüntüyü Dosyadan Yükle Butonu

Bu buton yardımı ile daha önceden .SKM formatında kaydedilen görüntülerin ekranda görüntülenmesi sağlanır. Bu butona basıldığı zaman “Bir .SKM Uzantılı Dosya Seçiniz” form başlığı ile dialog formu ekrana gelir. Bu form üzerinde istenilen directory içerisine geçilip istenilen .SKM uzantılı dosya seçilir ve ‘Aç’ butonuna basılırsa, ana formun en altında progressbar belirir, statusbarda “Yükleniyor...” ibaresi yazar ve .SKM uzantılı dosya sıkıştırma sürecinin tersine uğrayarak en sağdaki picturebox yardımı ile görüntülenir

Bu butona basılmadan önce ‘Temizle’ butonu ile daha önce ekrana yansıtılan görüntüler temizlenmelidir. Aksi takdirde daha önce üzerinde çalışılan ve ekran üzerinde görünen görüntüler ile ekrana yeni yansıyacak görüntü üst üste binebilir. Bundan dolayı; eğer programda daha önceden bir görüntü sıkıştırılmışsa ve programdan çıkılmadan bir .SKM uzantılı dosya açılmak isteniyorsa, ‘Temizle’ butonu ile görüntüler temizlenmelidir.

#### **4.2.10. Bitmap Formatında Kaydet Butonu**

Daha önce .SKM formatında kaydedilmiş bir görüntü üzerinde çalışılacaksa, Örneğin; sıkıştırılmış bir orthofoto görüntü açıldıktan sonra Netcad gibi bir paket programa aktarılmak istenebilir. Bu durumda tüm görüntü işlem programlarının tanıdığı bir resim formatı olan .BMP formatı, açılan görüntüyü saklamak için en uygun format olacaktır. Görüntü açıldıktan sonra bu butona basılırsa, “Kaydet” dialog formu ekrana gelir. Kayıt edilmek istenilen dizin seçilip, dosya ismi textbox içine yazıldıktan sonra, ‘Kaydet’ butonuna basılırsa, açılan görüntü bilgisayarda bitmap formatında kaydedilmiş olur. Böylece daha önceden sıkıştırılmış dosyalar üzerinde istenilen işlemler gerçekleştirilebilir.

#### **4.2.11. Çıkış Butonu**

Programdan çıkmak için kullanılır. Programdan çıkmak için diğer bir alternatif, ana form üzerindeki en sağ üst köşedeki “×” butonuna basmaktır.

#### **4.2.12. Temizle Butonu**

Eğer program vasıtasıyla herhangi bir görüntü sıkıştırılmış ve programdan çıkmadan .SKM uzantılı bir dosya, ‘Sıkıştırılmış Görüntüyü Dosyadan Yükle’ butonu ile görüntülenmek isteniyorsa, bu buton kullanılmalıdır. İlk önceki görüntü, bu buton vasıtasıyla temizlendikten sonra ‘Sıkıştırılmış Görüntüyü Dosyadan Yükle’ butonuna basılmalıdır. Aksi takdirde iki görüntü üst üste binecektir.

### 4.3. Programın Çalışma Prensibi

Görüntü Sıkıştırma Programı, bir ana ve bir de yardımcı form olmak üzere iki adet forma sahiptir. Ayrıca; programdaki değişkenlerin tanımlarını yapmak ve alt programları tanımlamak için, bir de ana modül içerir. Programın ikonuna (simgesine) basıldığı zaman, bilgisayar ekranına yansıyan, ana formun görüntüsüdür. Buna programın arayüzü de denir. Özetle; programın kullanıcısı programla ilgili tüm isteklerini programın arayüzünden gerçekleştirir.

Ana form üzerinde görüntü işlemleri için kullanılan üç adet Visual Basic Componenti yerleştirilmiştir. Bunlar, 'imagebox1', 'picturebox1' ve 'picturebox2' komponentleridir.

Programın diğer bir formu olan yardımcı form veya Form2 üzerine, görüntüleri matris formunda kullanıcıya sunmak için bir adet 'msFlexGrid' componenti yerleştirilmiştir.

Sıkıştırma Programının amacı, bitmap formatında bilgisayarda depolanmış herhangi bir görüntüyü 'imagebox' yardımı ile görüntüledikten sonra, bu görüntünün her bir pikselinin sayısal değerini matris formundaki bir değişkene aktarıp, bu değişken üzerinden sıkıştırma işlemlerini gerçekleştirip, 'picturebox' yardımı ile oluşan sonuç görüntüleri ekrana yansıtmak ve transforme olmuş görüntüyü uygun bir formatta bilgisayar ortamında saklamaktır. Bu aşamaların her birine ayrıntılı olarak aşağıdaki alt bölümlerde değinilecektir.

#### 4.3.1 Görüntünün Imagebox ' a Yüklenmesi

"Görüntü Yükle" butonuna basıldığı zaman, command1\_click alt yordamı (alt programı) otomatik olarak devreye girer ve

```
Image1.Picture = LoadPicture(CommonDialog1.FileName)
```

komut satırı ile görüntü, 'imagebox' bileşenine yüklenir. Burada 'image1' bileşeninin picture özelliğine, commanddialog1 diye isimlendirilen ve resim yükle butonuna basıldıktan sonra ekranda görülen dialog formunun altındaki textbox içinde yazılı olan dosya ismi yüklenir. Böylece 'imagebox' bileşenine bu görüntü yüklenmiş olur.

#### 4.3.2. Spektral Dönüşüm

Bu işlemi yapmak için “Gri Tonlamalı Görüntüyü Yükle” butonuna basılır. Aynı işlem, “Sıkıştır” butonuna basılarak ta gerçekleştirilebilir. Çünkü aynı program yordamları, bu iki alt yordam içinde de yazılmıştır. Aralarındaki tek fark, “Gri Tonlamalı Görüntüyü Yükle” butonunun, sadece grinin tonlarına çevrilmiş görüntüyü elde edip ekrana 'picturebox' bileşeni yardımı ile yansıtmasıdır. Sadece grinin tonlarına çevrilmiş görüntüyü göstermek için kullanılır. Her iki durumda da görüntünün her bir pikselindeki mavi ve yeşil bandlar ayıklanarak sadece kırmızı banttaki değerler elde edilir.

Bunun nedeni; kırmızı bandın, 24 bitlik bir piksel değerinde en düşük sekiz bitteki konumda olmasıdır. Bu durum programlama açısından, diğer bandların değerini elde etmekten daha kolaydır. Eğer 24 bitlik bir piksel değerinin ilk ve ortadaki baytlarını sıfır ile son baytını 255 veya ikilik tabana göre “11111111” ile “and” işlemine tabi tutulursa, sadece kırmızı banttaki değer elde edilir. Özetle; her bir piksel değeri ikilik tabanda “11111111” değerine karşı gelen onluk tabanda 255 değeri ile “and” işlemine tabi tutulur. Bu işlem aşağıda gösterilmiştir.

```

For i = 0 To 255 ' Görüntü A matrisine yükleniyor.
  For j = 0 To 255
    Gercek_A(i + 1, j + 1) = Point(Image1.Left + i, Image1.Top + j)
    kirmizi_band = Gercek_A(i + 1, j + 1) And 255
    A(i + 1, j + 1) = kirmizi_band
  Next j
Next i
For i = 0 To 255
  For j = 0 To 255
    Picture1.PSet (i, j), RGB(A(i + 1, j + 1), A(i + 1, j + 1), A(i + 1, j + 1))
  Next j
Next i

```



```

    A2(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A2(k, i + 64) = -0.5
    Else
        A2(k, i + 64) = 0.5
    End If
    k = k + 1
Next j
Next i

For i = 129 To 256
    A2(i, i) = 1
Next i

```

Yukarıdaki program yordamlarıyla 8 adet dönüşüm matrisi oluşturulduktan sonra, tek bir transformasyon matrisi olan  $W$  matrisini bulmak amacıyla  $A1, A2, A3, A4, A5, A6, A7, A8$  matrisleri birbirleriyle çarpılır. Bu işlem için programda bulunan ana modül içinde gösterilen “atama” ve “carpim” alt programlarından faydalanılmıştır. Bu alt programların avantajı; 8 adet çarpım işlemini defalarca yapmamak ve programın çalışma hızını artırmaktır. Bu şekilde 8 adet matris çarpımı için aynı yordamlar defalarca kullanılmamış, sadece bir kez kullanılmış olacaktır. Ana modül içindeki alt programlar aşağıda gösterilmiştir.

```

Public Sub carpim(iii, kkk, jjj) ‘ Matris çarpımlarını yapan alt program
    For ii = 1 To iii
        For kk = 1 To kkk
            C(ii, kk) = 0
            For jj = 1 To jjj
                C(ii, kk) = C(ii, kk) + AA(ii, jj) * BB(jj, kk)
            Next jj
        Next kk
    Next ii
End Sub

```

```

Public Sub atama() ‘ Çarpımdan sonra bazı ifadeleri birbirine atamak
    For ii = 1 To 256 ‘ için kullanılan alt program.
        For jj = 1 To 256
            W(ii, jj) = C(ii, jj)
        Next jj
    Next ii
End Sub

```

Yukarıda gösterilen alt programlara dallanmayı ve dönüşüm matrislerinin çarpımlarının nasıl yapıldığını göstermek amacıyla  $A1 \times A2$  ve  $A1 \times A2 \times A3$  işlemlerini gerçekleştiren program yordamları aşağıda gösterilmiştir.

```
For i = 1 To 256    'A1*A2 oluşturuluyor
  For j = 1 To 256
    AA(i, j) = A1(i, j)
    BB(i, j) = A2(i, j)
  Next j
Next i
carpim 256, 256, 256
atama
```

```
For i = 1 To 256    'W*A3 oluşturuluyor
  For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A3(i, j)
  Next j
Next i
carpim 256, 256, 256
atama
```

Yukarıdaki işlemlerden sonra,  $W$  dönüşüm matrisi elde edilir. Bundan sonraki adım, ortalama ve fark alma prosedürlerinin uygulandığı aşamadır. Bunun için  $W^T \times A$  ve  $W^T \times A \times W$  matris çarpımları yapılır. Bu çarpımlar ile ortalama ve fark alma prosedürleri gerçekleştirilmiş olur. Bu işlemler için “atama1”, “atama2” , “carpim” ve “transpoze\_al” alt programlarından faydalanılmıştır. Bu işlemlerin sonucunda,  $T1$  olarak isimlendirilen dönüşüme uğramış görüntü matrisi elde edilir. Programdaki bu işlemler aşağıda gösterilmiştir.

transpoze\_al        'W matrisinin transpozisini alır

```
For i = 1 To 256    'WT*A oluşturuluyor
  For j = 1 To 256
    AA(i, j) = WT(i, j)
    BB(i, j) = A(i, j)
  Next j
Next i
carpim 256, 256, 256
atama1
```

```
For i = 1 To 256    ' WT*A*W oluşturuluyor
  For j = 1 To 256
    AA(i, j) = WTA(i, j)
```

```

        BB(i, j) = W(i, j)
    Next j
Next i
carpim 256, 256, 256
atama2

```

Transforme olmuş görüntü matrisi  $T1$  bulunduktan sonra, bu matris içinde diğer değerlere göre çok ufak olan değerler, 'textbox' vasıtasıyla girilen bir  $\delta$  sıkıştırma katsayısı yardımıyla sıfır yapılır. Bu işlem sayesinde  $\delta$  sıkıştırma katsayısından daha küçük olan değerler yerine  $T1$  matrisinde sıfır koyulur. Bu süreçten sonra,  $T$  matrisi oluşturulur.  $T$  matrisinin oluşturulma aşaması aşağıda gösterilmiştir.

```

For i = 1 To 256      ' küçük değerler ihmal ediliyor
    For j = 1 To 256
        If Abs(T1(i, j)) <= Abs(Val(Text1.Text)) Then
            T(i, j) = 0
        Else
            T(i, j) = T1(i, j)
        End If
    Next j
Next i

```

$T$  matrisinin oluşturulması aşamasından sonra, sıkıştırılmış görüntü matrisi oluşturulmalıdır. Dolayısıyla  $W$  matrisinin tersi alınmalıdır. Bu işlem için,

```

For i = 1 To 256      ' W matrisinin tersi hesaplanıyor
    For j = 1 To 256
        If W(i, j) = 0 Then
            Q(j, i) = 0
        ElseIf W(i, j) > 0 Then
            Q(j, i) = 1
        ElseIf W(i, j) < 0 Then
            Q(j, i) = -1
        End If
    Next j
Next i

```

program satırları kullanılmıştır. Böylece  $W$  matrisinin inversi (tersi) olan  $Q$  matrisi üretilmiştir. Bu aşamalardan sonra sıkıştırılmış görüntü matrisini temsil eden  $SKM$  matrisini elde etmek için aşağıda gösterilen program satırları kullanılmıştır.

```

For i = 1 To 256
  For j = 1 To 256
    QT(i, j) = Q(j, i)
  Next j
Next i

```

```

For i = 1 To 256
  For j = 1 To 256
    AA(i, j) = QT(i, j)
    BB(i, j) = T(i, j)
  Next j
Next i
  carpim 256, 256, 256
  atama3

```

```

For i = 1 To 256
  For j = 1 To 256
    AA(i, j) = QTT(i, j)
    BB(i, j) = Q(i, j)
  Next j
Next i
  carpim 256, 256, 256
  atama4

```

Bu aşama için yine programın ana modülünde bulunan “carpim”, “atama3”, “atama4” ve “transpoze\_al” alt programlarından faydalanılmıştır. Ana modülde bulunan bu alt programlardan “atama3” ve “atama4” alt programları aşağıda gösterilmiştir.

```

Public Sub atama3()
  For ii = 1 To 256
    For jj = 1 To 256
      QTT(ii, jj) = C(ii, jj)
    Next jj
  Next ii
End Sub

```

```

Public Sub atama4()
  For ii = 1 To 256
    For jj = 1 To 256
      SKM(ii, jj) = C(ii, jj) ‘Sıkıştırılmış Görüntü Matrisi
    Next jj
  Next ii
End Sub

```

Sıkıştırılmış görüntü matrisi elde edildikten sonra, 'picturebox' componentinde onu görüntülemek için,

```

For i = 0 To 255
  For j = 0 To 255

Picture2.PSet (i, j), RGB(Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))),Int(Abs(SKM(i + 1, j + 1))))

  Next j
Next i

```

yordamları kullanılmıştır. Burada, pset(x,y) fonksiyonu *SKM* matrisindeki RGB değerlerini 'picture2' komponentine aktaracak ve sıkıştırılmış görüntü matrisi ekranda görünür hale gelecektir.

#### 4.3.4. Sıkıştırılmış Görüntünün Bilgisayara Kaydedilmesi

Görüntü Sıkıştırma Programı ile sıkıştırılan görüntü, bilgisayar ortamına ayrı bir formatta kayıt edilmelidir. Çünkü; bitmap formatında kayıt edilen hiçbir görüntünün bilgisayarda kapladığı hafıza miktarı küçülmez. Bitmap formatında  $256 \times 256$  piksel karelik dolu bir görüntü, aynı boyutta boş bir görüntü gibi yaklaşık 192 kilobayt yer kaplar.

Bitmap formatında aynı boyutlarda bir görüntünün hafızada kapladığı yeri azaltmak için, siyah beyaz bir görüntü olarak kayıt etmemiz gerekir. Bu durumda da görüntü kayıpları olacaktır ve sağlıklı bir sonuç ortaya çıkmayacaktır. Bu nedenlerden dolayı, görüntünün sıkıştırılması aşamalarının en önemlilerinden biri olan ve spatial transformun kapsamına giren, uygun bir sıkıştırma formatı geliştirilmiştir.

##### 4.3.4.1. Kayıt Formatı

Daha önce bahsedildiği gibi, görüntünün her bir pikseline spektral transform uygulanmış ve her bir pikseldeki mavi ve yeşil bandlar ayıklanarak, sadece kırmızı band değerleri elde edilmiştir. Görüntüyü ekranda çizmek içinse, Visual Basic dilinin

RGB fonksiyonunda yeşil ve mavi band değerleri yerine kırmızı band değerleri konularak resmin gri tonlarla çizilmesi sağlanmıştır. Örneğin; kırmızı band değeri 155 olan bir pikselin gri tondaki değerini elde etmek için RGB(155,155,155) yazılmalıdır. Bu durumda, bilgisayar ortamında üç bandın sayısal değerlerini kaydetmek yerine, tek bandın sayısal değerini kaydetmek görüntünün bilgisayarda kapladığı hafıza miktarını azaltacaktır.

Ayrıca; kaydedeceğimiz görüntüyü tekrar bilgisayar ortamında görüntüleyebilmek için bu görüntünün sıfırdan farklı piksellerinin matris içindeki konumlarının da kaydedilmesi gerekecektir. Yapılan denemeler sonucu, sıfırdan farklı her bir eleman için satır ve sütun kodlaması yapıldığı zaman boyutu 192 kilobayt olan  $256 \times 256$  piksel karelik bir bitmap görüntünün hafızada kapladığı yerin daha da artacağı görülmüştür. Bu sorunu gidermek için, *SKM* formatı geliştirilmiştir. Bu formatta bilgisayara kaydedilen bir dosyanın içeriğinin bir kısmı aşağıda gösterilmiştir.

? 1 1 153 2 -23 ? 3 15 -22 ? 6 6 23 ? 8 7 22 27 28 .....= (Dosya sonu)

Yukarıda verilen dosya içeriğine göre, bir boşluk karakterinden sonra ‘?’ karakteri kullanılmış ve ‘?’ karakterinden sonra yine bir boşluk karakteri kullanılmıştır. ‘?’ karakterinin sağındaki boşluk karakterinden sonra 1 sayısı ve yine ondan sonra bir boşluk ve 153 sayısı kullanılmıştır. Bu, birinci satırın birinci sütununda RGB renk sistemine göre kırmızı bandın karşılığı olan 153 değeri bulunmaktadır anlamına gelir. Aynı şekilde bunu takiben, birinci satırın ikinci sütununda -23 değeri bulunmaktadır. Aynı şekilde yorumlanmaya devam edilirse, ‘?’ karakterinden sonra gelen değer, satır numarası anlamına geldiğinden dolayı üçüncü satırın onbeşinci sütununda -22, altıncı satırın altıncı sütununda 23, sekizinci satırın yedinci sütununda 22 değeri bulunmaktadır.

Dosyanın sonunda ise ‘=’ karakteri yer almaktadır. Bu, dosyanın sonunun geldiği anlamındadır. Bu formatta, RGB değeri sıfırdan farklı pikseller için her satır sadece bir kez kodlanmıştır. Böylece sayısal değeri sıfır olmayan ve aynı satırda

bulunan değerler için satır numarası sadece bir kez kullanılarak görüntünün sıkışması sağlanmıştır.

‘?’ karakterinin sağındaki boşluktan sonra gelen karakter satır numarası anlamındadır. Satır karakterinin sağındaki boşluktan sonra gelen karakter pikselin RGB değeridir. Bir sonraki ‘?’ karakterine kadar satır değeri değişmez ve bu şekilde her piksel değeri için satır sayısı yeniden kodlanmayarak görüntünün sıkışması sağlanmıştır. Bu formatı kodlayan program yordamı aşağıda gösterilmiştir.

Private Sub Command10\_Click()

```

CommonDialog1.DialogTitle = "Bir Dosya İsmi Yazınız"
CommonDialog1.Filter = "Sıkıştırma Dosyaları|*.SKM"
CommonDialog1.Action = 2
CommonDialog1.FileName = CommonDialog1.FileName & ".skm"
If CommonDialog1.FileName = "" Then
    GoTo bos
End If
Open CommonDialog1.FileName For Output As #1
For i = 1 To 256
    For k = 1 To 256
        If T(i, k) <> 0 Then
            p = 0
            Print #1, " " & "?" & Str(i); Exit For
        ElseIf T(i, k) = 0 Then
            p = 1
        End If
    Next k
    For j = 1 To 256
        If p = 1 Then
            GoTo atla
        End If
        If T(i, j) > 0 Then
            Print #1, Str(j) & Str(T(i, j));
        ElseIf T(i, j) < 0 Then
            Print #1, Str(j) & " " & Str(T(i, j));
        End If
    Next j
    Print #1, " " & "="
    Close #1
atla:
bos:
End Sub

```

#### 4.3.5. Daha Önce . SKM Formatında Kaydedilmiş Bir Görüntünün Bilgisayarda Görüntülenmesi

Daha önceden Görüntü Sıkıştırma Programı vasıtası ile elde edilen .SKM uzantılı dosyalar, yine aynı program vasıtası ile görüntülenip, bitmap formatında bilgisayar ortamına kayıt edilebilirler. Bu işlemleri gerçekleştirmek için, ilk önce bu dosyanın içinde bulunan değerler belirli değişkenlere aktarılmalıdır. Bu işlemi gerçekleştiren program yordamı, “Sıkıştırılmış Görüntüyü Dosyadan Yükle” butonuna basıldıktan sonra işlemeye başlar. Bu yordam aşağıda gösterilmiştir.

```
Private Sub Command11_Click()
```

```
    CommonDialog1.DialogTitle = "Bir .skm Uzantılı Dosya Seçiniz"
    CommonDialog1.Filter = "Sıkıştırma Dosyaları|*.SKM"
    CommonDialog1.Action = 1
```

```
    If CommonDialog1.FileName = "" Then
        GoTo bitir
    End If
    Open CommonDialog1.FileName For Input As #1
```

```
    Line Input #1, tum
    i = 1
```

```
ugur:
```

```
    If Mid(tum, i + 1, 1) = "?" Then
        k = i + 2: satir = ""
        Do
            k = k + 1
            satir = satir + Mid(tum, k, 1)
            satir1 = Val(satir)
        Loop While Mid(tum, k, 1) <> " "
```

```
    ElseIf Mid(tum, i + 1, 1) = "=" Then
        GoTo kemal
```

```
    End If
    sutun = Empty
    Do
        k = k + 1
        sutun = sutun + Mid(tum, k, 1)
        sutun1 = Val(sutun)
    Loop While Mid(tum, k, 1) <> " "
    sayi = ""
```

```
    Do
        k = k + 1
```

```

        sayi = sayi + Mid(tum, k, 1)
        sayi1 = Val(sayi)
        Loop While Mid(tum, k, 1) <> " "
        T(satir1, sutun1) = sayi

        i = k
        GoTo ugur
kemal:
    Close #1
    For i = 1 To 256
    For j = 1 To 256
        If T(i, j) = Empty Then
            T(i, j) = 0
        End If
    Next j
    Next i
    .
    .
    .
    .
    .
    .
    .
    .
    .
    } Sıkıştırılmış görüntüyü açmaya yarayan yordamlar.
bitir:

End Sub

```

Bu işlemlerden sonra,  $T$  matrisi veya bilgisayar ortamına .SKM formatında kayıt edilen görüntü matrisi elde edilmiş olacaktır.  $T$  matrisi elde edildiğinden dolayı, aşağıdaki yordam yardımı ile görüntü 'picturebox' komponentine aktararak görüntülenir.

```

For i = 0 To 255
    For j = 0 To 255

        Picture2.PSet (i, j), RGB(Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))))

    Next j
Next i

```

Elde edilen bu görüntü üzerinde herhangi bir çalışma yapılmak istenebilir. Bu durumda tüm görüntü işlem programlarının tanıdığı format şekli olan bitmap formatında görüntünün kayıt edilmesi gerekir. Bunun için aşağıdaki yordam, "BİTMAP Formatında Kaydet" butonuna basıldıktan sonra işletilir.

```
Private Sub Command12_Click()
```

```
    CommonDialog2.DialogTitle = "Uzantısız Bir Dosya İsmi Yazınız"
    CommonDialog2.Filter = "Bitmap Dosyaları|*.BMP"
```

```
    CommonDialog2.Action = 2
    CommonDialog2.FileName = CommonDialog2.FileName & ".bmp"
    If CommonDialog2.FileName = "" Then
        GoTo son
    End If
    Picture3.Picture = Picture2.Image
```

```
    SavePicture Picture3.Picture, CommonDialog2.FileName
```

```
son:
```

```
End Sub
```

#### 4.3.6. Görüntü Matrislerinin MSFlexGrid ile Görüntülenmesi

Form2 üzerine yerleştirilen MsFlexGrid yardımı ile Orijinal Görüntü Matrisi ve Transform Olmuş Görüntü Matrisi ekrana yansıtılabilir. Böylece, görüntülerin sıkıştırılma süreci hakkında yorumlar yapılabilir. “Gerçek Görüntü Matrisi” butonuna basılarak, aşağıdaki yordam işletilir.

```
Private Sub Command4_Click()
```

```
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11
```

```
    Form2.MSFlexGrid1.AllowUserResizing = flexResizeBoth
```

```
    Form2.Label1.Caption = "Gerçek Görüntü Matrisi"
```

```
    Form2.MSFlexGrid1.Cols = 257
```

```
    Form2.MSFlexGrid1.Rows = 257
```

```
    Form2.MSFlexGrid1.FixedCols = 1
```

```
    Form2.MSFlexGrid1.FixedRows = 1
```

```
    For i = 1 To 256
```

```
        Form2.MSFlexGrid1.Row = i
```

```
        Form2.MSFlexGrid1.Col = 0
```

```
        Form2.MSFlexGrid1.Text = i
```

```
        Form2.MSFlexGrid1.Row = 0
```

```
        Form2.MSFlexGrid1.Col = i
```

```
        Form2.MSFlexGrid1.Text = i
```

```
    Next i
```

```
    For i = 1 To 256
```

```
        For j = 1 To 256
```

```

Form2.MSFlexGrid1.Row = i
Form2.MSFlexGrid1.Col = j
Form2.MSFlexGrid1.Text = Gercek_A(i, j)
Next
Next
MousePointer = 1
Form2.Show
StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

```

Bu yordamın çalışması sonucu, orijinal görüntü matrisinin piksel değerleri msFlexGrid sayesinde görüntülenir. “Transforme Olmuş Görüntü Matrisi” butonuna basılarak aşağıdaki yordam çalıştırılıp, Transforme Olmuş Görüntü Matrisi ekrana yansıtılabilir.

```

Private Sub Command3_Click()
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11
    Form2.MSFlexGrid1.AllowUserResizing = flexResizeBoth

    Form2.Label1.Caption = "Sayısal Olarak Basitleştirilmiş Görüntü Matrisi"
    Form2.MSFlexGrid1.Cols = 257
    Form2.MSFlexGrid1.Rows = 257
    Form2.MSFlexGrid1.FixedCols = 1
    Form2.MSFlexGrid1.FixedRows = 1
    For i = 1 To 256
        Form2.MSFlexGrid1.Row = i
        Form2.MSFlexGrid1.Col = 0
        Form2.MSFlexGrid1.Text = i
        Form2.MSFlexGrid1.Row = 0
        Form2.MSFlexGrid1.Col = i
        Form2.MSFlexGrid1.Text = i
    Next i

    For i = 1 To 256
        For j = 1 To 256
            Form2.MSFlexGrid1.Row = i
            Form2.MSFlexGrid1.Col = j
            Form2.MSFlexGrid1.Text = T(i, j)
        Next
    Next
    MousePointer = 1
    Form2.Show
    StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

```

## 5. UYGULAMA

### 5.1. Orthofoto Görüntülerin Sıkıştırılması

Haar Wavelet Dönüşümünün,  $256 \times 256$  piksel karelik bir görüntüde farklı sıkıştırma katsayılarında hangi oranlarda sıkıştırma yapabildiğini görmek için, bu boyutlarda bir orthofoto görüntü üzerinde program vasıtasıyla uygulamalar yapıldı. Bu uygulamalar, uygulamaların sonuçları bu bölümde gösterilecektir. Şekil 5.1, sıkıştırılacak olan orijinal orthofoto görüntüyü göstermektedir.



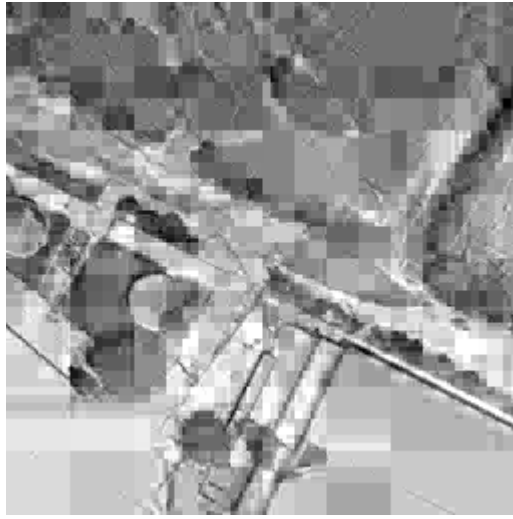
Şekil 5.1.  $256 \times 256$  piksellik (192 kilobayt) Orijinal Orthofoto Görüntü.

Şekil 5.2, sıkıştırma katsayısı  $\delta = 9$  alındığı zaman oluşan sıkıştırılmış görüntüyü göstermektedir. Buradaki sıkıştırma işlemi ile 65.536 pikselin ( $256 \times 256 = 65.536$  piksel) sıfırdan farklı 4.169 elemanının kaldığını göstermektedir. Buda sıkıştırma oranının yaklaşık 1'e 16 olduğunu göstermektedir.

Şekil 5.3, sıkıştırma katsayısı  $\delta = 7$  alındığı zaman oluşan sıkıştırılmış görüntüyü göstermektedir. Buradaki sıkıştırma işlemi ile 65.536 pikselin sıfırdan farklı 6.801 elemanının kaldığını göstermektedir. Buda sıkıştırma oranının yaklaşık 1'e 10 olduğunu göstermektedir.

Şekil 5.4, sıkıştırma katsayısı  $\delta = 5$  alındığı zaman oluşan sıkıştırılmış görüntüyü göstermektedir. Buradaki sıkıştırma işlemi ile 65.536 pikselin sıfırdan farklı 11.601 elemanının kaldığını göstermektedir. Buda sıkıştırma oranının yaklaşık 1'e 6 olduğunu göstermektedir.

Şekil 5.5, sıkıştırma katsayısı  $\delta = 3$  alındığı zaman oluşan sıkıştırılmış görüntüyü göstermektedir. Buradaki sıkıştırma işlemi ile 65.536 pikselin sıfırdan farklı 20.499 elemanının kaldığını göstermektedir. Buda sıkıştırma oranının yaklaşık 1'e 3 olduğunu göstermektedir.



Şekil 5.2.  $\delta=9$ , %94' lük sıkışma yüzdesi ile.(11.5 kilobayt)



Şekil 5.3.,  $\delta=7$ , %90 'lık sıkışma yüzdesi ile(19 kilobayt)



Şekil 5.4.  $\delta=5$ , %82 'lik sıkıştırma yüzdesi ile(34.5 kilobayt)



Şekil 5.5.  $\delta=3$ , %69'luk sıkıştırma yüzdesi ile(59.5 kilobayt)

Bu uygulamalar sonucunda, sıkıştırma katsayısı  $\delta = 7$  ve  $\delta = 5$  durumlarında oluşan görüntünün orijinal görüntüye hem çok yakın olduğu hem de sıkıştırma oranının istenilen derecede büyük bir oran olduğunu göstermiştir.  $\delta = 3$  durumunda orijinale en yakın görüntü elde edilmiştir. Fakat, bu durumda sıkıştırma oranı istenilen derecede değildir. Diğer durumlarda ise elde edilen görüntülerde, orijinal görüntünün önemli sayılacak ayrıntıları kaybolmuştur. Her ne kadar sıkıştırma

oranları istenen derecede olsa dahi, orijinal görüntüye oranla çok büyük farklılıklar olduğundan bu sıkıştırma derecelerinde sıkıştırılan görüntüler tercih edilmezler.

## 5.2. Grafik Kadastral Paftaların Sıkıştırılması

Haar Wavelet Dönüşümü bu bölümde tarayıcı vasıtasıyla raster görüntü haline getirilmiş Şekil 3.2 'de görülen Grafik Kadastral paftaya uygulanmıştır. Görüntünün boyutları 1792 x 1280 piksel karedir. Bu boyutlardaki bir görüntünün Görüntü Sıkıştırma Programı ile sıkıştırılması için 256 x 256 piksel karelik görüntülere parçalanması gerekmektedir. Bundan dolayı, görüntü 35 alt görüntüye ayrılmış ve bu alt görüntülerin her biri sıkıştırma katsayısı  $\delta = 5$  ve  $\delta = 7$  için ayrı ayrı sıkıştırılmıştır. Raster tipte tarayıcıyla bitmap formatında taranıp kaydedilmiş bu görüntünün bilgisayarda kapladığı alan 6721 kilobayttır.

$\delta = 5$  için sıkıştırma yapıldığında görüntünün bilgisayarda kapladığı alan yaklaşık 981 kilobayt olmaktadır. Bu sonuç orijinal görüntünün bilgisayarda kapladığı alanın yaklaşık %15 'idir. Dolayısıyla yaklaşık %85 'lik bir sıkıştırma yüzdesi elde edilmiştir. Sıkıştırılan görüntüler, Görüntü Sıkıştırma Programı ile yeniden Bitmap formatına getirilip birleştirildiği zaman görüntüde çok önemli bir kayıp olmadığı gözlenmiştir. Şekil 5.6,  $\delta = 5$  için sıkıştırılan görüntüyü göstermektedir.

$\delta = 7$  için sıkıştırma yapıldığında görüntünün bilgisayarda kapladığı alan yaklaşık 846 kilobayt olmaktadır. Bu sonuç orijinal görüntünün bilgisayarda kapladığı alanın yaklaşık %12 'sidir. Dolayısıyla yaklaşık %88 'lik bir sıkıştırma yüzdesi elde edilmiştir. Sıkıştırılan görüntüler Görüntü Sıkıştırma Programı ile yeniden Bitmap formatına getirilip birleştirildiği zaman görüntüde çok önemli bir kayıp olmadığı gözlenmiştir. Şekil 5.7,  $\delta = 7$  için sıkıştırılan görüntüyü göstermektedir.

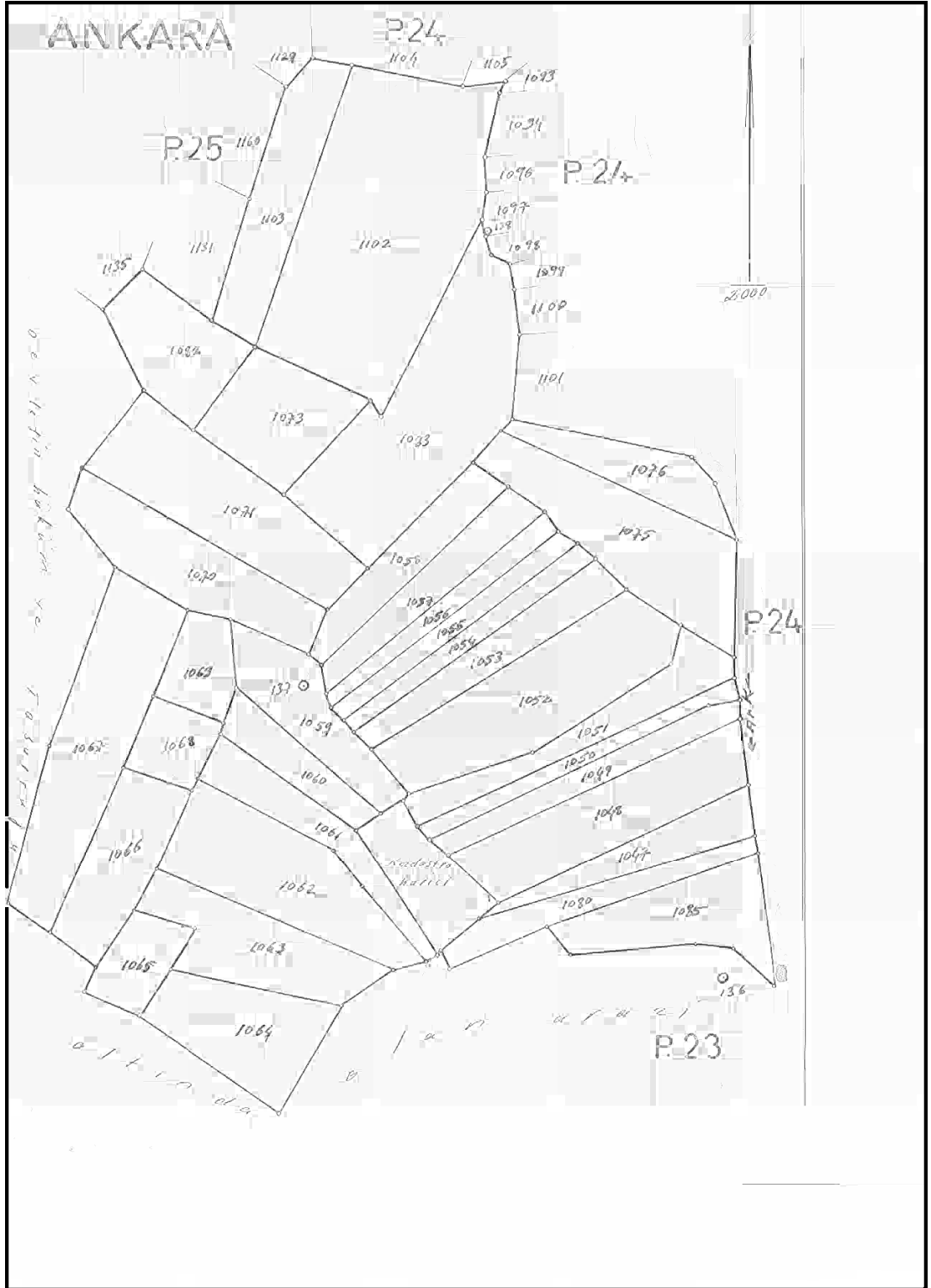
$\delta = 15$  için sıkıştırma yapıldığında görüntünün bilgisayarda kapladığı alan yaklaşık 466 kilobayt olmaktadır. Bu sonuç orijinal görüntünün bilgisayarda kapladığı alanın yaklaşık %7 'sidir. Dolayısıyla yaklaşık %93 'lük bir sıkıştırma

yüzdesi elde edilmiştir. Sıkıştırılan görüntüler Görüntü Sıkıştırma Programı ile yeniden Bitmap formatına getirilip birleştirildiği zaman görüntüde önemli kayıplar olduğu gözlenmiştir. Şekil 5.8,  $\delta = 15$  için sıkıştırılan görüntüyü göstermektedir.

$\delta = 22$  için sıkıştırma yapıldığında görüntünün bilgisayarda kapladığı alan yaklaşık 297 kilobayt olmaktadır. Bu sonuç orijinal görüntünün bilgisayarda kapladığı alanın yaklaşık %4 'üdür. Dolayısıyla yaklaşık %96 'lık bir sıkıştırma yüzdesi elde edilmiştir. Sıkıştırılan görüntüler Görüntü Sıkıştırma Programı ile yeniden Bitmap formatına getirilip birleştirildiği zaman görüntüde çok önemli kayıplar olduğu gözlenmiştir. Şekil 5.9,  $\delta = 22$  için sıkıştırılan görüntüyü göstermektedir.







Şekil 5.8.  $\delta = 15$  katsayısı ile sıkıştırılmış görüntü, %93'lük sıkışma(466 kb)



Şekil 5.9.  $\delta = 22$  katsayısı ile sıkıştırılmış görüntü, %96'lık sıkıştırma(297 kb)

## 6. SONUÇ VE ÖNERİLER

Bu tezde Dalgacık Dönüşümlerinin bir çeşidi olan Haar Dalgacık Dönüşümü anlatıldı. Tezin uygulamasında bu dönüşümü kullanan bir program geliştirildi. Program vasıtasıyla raster veri yapısında bilgisayar ortamına kaydedilen görüntüler farklı sıkıştırma katsayılarında sıkıştırıldı.

Raster görüntü örneği olan orthofoto bir görüntü program vasıtasıyla sıkıştırıldı. Sıkıştırma işlemi ayrı ayrı  $\delta = 3, 5, 7, 9$  katsayılarında denendi. Sıkıştırma katsayısı 3 için sıkıştırma işleminden sonra görüntüde 65536 pikselin sıfırdan farklı 20499 pikselinin kaldığı tespit edildi. Bu yaklaşık olarak %69'luk bir sıkıştırma yüzdesine karşılık gelmektedir. Bu kadarlık bir sıkıştırma yüzdesine karşılık sıkışmış görüntü orijinal görüntüye çok yakındır.

Sıkıştırma katsayısı 5 için yapılan uygulamada 65536 pikselin sıfırdan farklı 11601 elemanı kaldığı görülmüştür. Bu da yaklaşık olarak %83 'lük bir sıkışmaya karşılık gelmektedir. Bu kadar büyük bir yüzde ile sıkışan bir görüntüde bile görüntü kaybının çok az olduğu ve sıkışmış görüntünün normal görüntüye çok yakın olduğu görülmektedir.

Sıkıştırma katsayısı 7 için yapılan uygulamada 65536 pikselin sıfırdan farklı 6801 elemanının kaldığı gözlenmiştir. Bu da yaklaşık olarak %90 'lık bir sıkıştırma yüzdesine karşılık gelmektedir. Bu katsayıda sıkıştırılan görüntü ile orijinal görüntü arasında görüntü kaybının biraz arttığı gözlenmiştir. Fakat bu kaybın görüntünün işlevselliğini bozacak kadar büyük bir kayıp olmadığı görülmüştür.

Sıkıştırma katsayısı 9 için yapılan uygulamada 65536 pikselin sıfırdan farklı 4169 elemanının kaldığı tespit edilmiştir. Bu yaklaşık olarak %94 'lük bir sıkışma yüzdesidir. Bu katsayıda sıkıştırılan görüntüde çok büyük görüntü kayıplarının olduğu gözlenmiş ve sıkışmış görüntünün orijinal görüntüden çok farklı bir görüntü olduğu sonucuna varılmıştır.

Yukarıda elde edilen bilgiler ışığında orthofoto gibi çok yoğun bir görüntü için en uygun sıkıştırma katsayısının 3 ile 7 arasında olduğu görülmüştür. Bunun anlamı, geliştirilen program vasıtasıyla sıkıştırma katsayısı 3 ile 7 arasında sıkıştırılmış görüntülerden iyi sonuç alınabileceğidir. Sıkıştırma katsayısı 7 ‘den sonra çok büyük görüntü kayıplarının olduğu gözlenmiştir.

Diğer bir uygulama ise yine bir raster görüntü örneği olan Kadastral Grafik Paftaların sıkıştırılmasıdır. Bu uygulama için  $\delta = 5, 7, 15$  ve 22 katsayıları kullanılmıştır. Sıkıştırma katsayısı 5 için %85 ‘lik bir sıkıştırma yüzdesi, 7 için %88 ‘lik bir sıkıştırma yüzdesi, 15 için %93 ‘lük bir sıkıştırma yüzdesi, 22 için %96 ‘lık bir sıkıştırma yüzdesi elde edilmiştir. Sıkıştırma katsayısı 5 ve 7 için normal görüntüye çok yakın görüntüler elde edilmiştir. 15 için görüntü kayıpları olmaktadır. 22 için görüntü kayıpları daha fazla olmaktadır. Orthofoto görüntüler, Grafik Kadastral Paftalardan daha yoğun görüntüler olduğundan, Grafik Kadastral Pafta görüntüleri için daha iyi bir yüzdede sıkıştırılma yapılabilmektedir.

Haar Dalgacık Dönüşümü ayrıca her görüntüde uygulanabilir. Bu dönüşüm vasıtasıyla görüntüler dışında ayrıca ses dalgası sıkıştırılabilir, görüntülerde ve sinyallerde oluşan gürültüler (noise) giderilebilir.

Bu tezin devamında bir ses dalgası üzerinde sıkıştırma uygulamaları yapılabilir. Ayrıca, eski görüntülerde ve ses kayıtlarında bulunan gürültülerin, Haar Dalgacık Dönüşümü kullanılarak giderilmesi de bu tezin devamı olabilir.

## KAYNAKLAR

1. Bala Erdem, 2001.Applications of Multiwavelet Techniques to Image Denoising, Yüksek Lisans Tezi, Boğaziçi Üniversitesi,İstanbul.
2. Demir Hasan,1999. Wavelet Dönüşümü ve İşaret İşleme Uygulamaları , Yüksek Lisans Tezi, İstanbul Üniversitesi, İstanbul.
3. Ergün Hüseyin, 1997. Dalgacık Dönüşümündeki Yerel Tepe Değerlerinden Görüntü Geriçatma Ve Sıkıştırma, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.
4. G. Van Der Auwera, A. Munteanu and Cornelis J., 2003. Quincunx Wavelet Image Compression : 2D Filter Design By Transformations of Variables and Quadtree Coding Approach., Vrije Üniversitesi, Brüksel.  
<http://www.etro.vub.ac.be/Members/munteanu.adrian/private/Conferences/FilterDesignQuincunxWaveletQuadtreeCodingSPS2000.pdf>
5. Halvorson Michael, 2000, Visual Basic 6.0. Proffessional.
6. Karagülle İhsan & Pala Zeydin, 1999. Visual Basic 6.0 Professional, İstanbul.
7. Kızıl Ali & Uzunoğlu Mehmet & Onar Ömer Çağlar, 1999, Kolay Anlatımı İle İleri Düzeyde Matlab 6.0 – 6.5., İstanbul.
8. Krestyannikov Evgeny, 2002. Research on the Transform in Multispectralimage Compression, Intermediate Report, Lappeenranta Üviversitesi, Lappeenranta, Finlandiya. [http://www.lut.fi/~krestyan/Initial\\_Presentation.pdf](http://www.lut.fi/~krestyan/Initial_Presentation.pdf)
9. Manfred Kopp, Lossless Wavelet Based Image Compression with Adaptive 2D Decomposition, Viena Üniversitesi, Viena, Avusturya.  
[http://wscg.zcu.cz/wscg96/papers96/Kopp\\_96.pdf](http://wscg.zcu.cz/wscg96/papers96/Kopp_96.pdf)

10. Mert Funda Yıldırım, 1999. İmage Compression By Using Wavelet Transform, Yüksek Lisans Tezi, Ortadoğu Teknik Üniversitesi, Ankara.
11. Mladen Wictor Wickerhauser, 1994. Adapted Wavelet Analysis from Theory to Software, Wellesley,MA.
12. Peggy Morton and Arne Petersen, 1997. Image Compression Using Haar Wavelet,College of Redwoods.  
<http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall97/PMorton/pmorton.pdf>
13. Pujita Pinnemaneni, 1997. 3-D Haar Wavelet Transformation in Java,  
<http://www.erc.msstate.edu/vail/pubs/2001/saladi02.pdf>
14. Sandberg Kristian, 1998, The Haar Wavelet Transform, Colorado Üniversitesi.
15. Saral Tunç, 2000, Wavelet Transformasyonları İle Görüntü Sıkıştırma, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul.
16. Sönmez A. Gürkan, 1997, Analog VLSI Implementation of Wavelet Transforms Using Switched-Capacitor Filters, Yüksek Lisans Tezi, Boğaziçi Üniversitesi., İstanbul.
17. Thomas M. Lillesand and Ralph W. Kiefer, 1994, Remote Sensing and Image Interpretation.

## ÖZGEÇMİŞ

Uğur Kemal ÇELİK 1973 yılında Erzurum’da doğdu. İlk, orta ve liseyi Ankara’da bitirdi. 1994 yılında Selçuk Üniversitesi Mühendislik ve Mimarlık Fakültesi Jeodezi ve Fotogrametri Mühendisliği Bölümü’nde üniversiteye başladı. Buradan 1999 yılında mezun oldu. Bir müddet özel bir şirkette çalışdıktan sonra, 2001 yılında Gebze Yüksek Teknoloji Enstitüsü’nde yüksek lisansa başladı. 2002 yılında Midyat Kadastro Müdürlüğü’nde Kadastro Kontrol Mühendisi olarak göreve başladı. Halen bu görevi sürdürmektedir.

## EKLER

### EK 1. Görüntü Sıkıştırma Programı

#### Ana Program

```

Private Sub Command1_Click()
    CommonDialog1.DialogTitle = "Bir Resim Dosyası Seçiniz"
    CommonDialog1.Filter = "Bitmap Dosyaları|*.BMP"
    CommonDialog1.Action = 1
    MousePointer = 11

    Image1.Picture = LoadPicture(CommonDialog1.FileName)
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    If Image1.Picture <> none Then
        Label2.Caption = "Normal Görüntü"
    ElseIf Image1.Picture = none Then
        Label2.Caption = ""
    End If

    MousePointer = 1
    StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

Private Sub Command10_Click()

    CommonDialog1.DialogTitle = "Uzantısız Bir Dosya İsmi Yazınız"
    CommonDialog1.Filter = "Sıkıştırma Dosyaları|*.SKM"

    CommonDialog1.Action = 2
    CommonDialog1.FileName = CommonDialog1.FileName & ".skm"
    If CommonDialog1.FileName = "" Then
        GoTo bos
    End If
    MousePointer = 11
    Open CommonDialog1.FileName For Output As #1

    For i = 1 To 256
        For k = 1 To 256
            If T(i, k) <> 0 Then
                p = 0
                Print #1, " " & "?" & Str(i); Exit For
            ElseIf T(i, k) = 0 Then
                p = 1
            End If
        Next k
    Next i
End Sub

```

```

        End If
    Next k
    For j = 1 To 256
        If p = 1 Then
            GoTo atla
        End If
        If T(i, j) > 0 Then
            Print #1, Str(j) & Str(T(i, j));
        ElseIf T(i, j) < 0 Then
            Print #1, Str(j) & " " & Str(T(i, j));
        End If
    atla:
        Next j
        Next i
        Print #1, " " & "="
        Close #1
        MousePointer = 1
    bos:
End Sub

Private Sub Command11_Click()

    CommonDialog1.DialogTitle = "Bir .skm Uzantılı Dosya Seçiniz"
    CommonDialog1.Filter = "Sıkıştırma Dosyaları|*.SKM"
    CommonDialog1.Action = 1

    If CommonDialog1.FileName = "" Then
        GoTo bitir
    End If
    Open CommonDialog1.FileName For Input As #1

    MousePointer = 11

    ProgressBar1.Visible = True
    ProgressBar1.Value = 0
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    Line Input #1, tum
    i = 1
    ugr:
        If Mid(tum, i + 1, 1) = "?" Then
            k = i + 2: satir = ""

            Do
                k = k + 1
                satir = satir + Mid(tum, k, 1)
                satir1 = Val(satir)
            Loop While Mid(tum, k, 1) <> " "

            ElseIf Mid(tum, i + 1, 1) = "=" Then
                GoTo kemal

```

```

End If
sutun = Empty
Do
    k = k + 1
    sutun = sutun + Mid(tum, k, 1)
    sutun1 = Val(sutun)
Loop While Mid(tum, k, 1) <> " "
sayi = ""

Do
    k = k + 1
    sayi = sayi + Mid(tum, k, 1)
    sayi1 = Val(sayi)
Loop While Mid(tum, k, 1) <> " "
T(satir1, sutun1) = sayi

i = k
GoTo ugur

```

kemal:

```

Close #1
For i = 1 To 256
For j = 1 To 256
    If T(i, j) = Empty Then
        T(i, j) = 0
    End If
Next j
Next i

```

ProgressBar1.Value = 2

```

k = 1          'A1 matrisi oluşturuluyor
For i = 1 To 128
For j = 1 To 2
    A1(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A1(k, i + 128) = -0.5
    Else
        A1(k, i + 128) = 0.5
    End If
    k = k + 1
Next j
Next i

```

ProgressBar1.Value = 4

```

k = 1          'A2 matrisi oluşturuluyor
For i = 1 To 64
For j = 1 To 2

```

```

A2(k, i) = 0.5
M = k / 2
N = Int(M)
If M = N Then
    A2(k, i + 64) = -0.5
Else
    A2(k, i + 64) = 0.5
End If
k = k + 1
Next j
Next i

```

```

For i = 129 To 256
    A2(i, i) = 1
Next i

```

```

k = 1                'A3 matrisi oluşturuluyor
For i = 1 To 32
For j = 1 To 2
    A3(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A3(k, i + 32) = -0.5
    Else
        A3(k, i + 32) = 0.5
    End If
    k = k + 1
Next j
Next i

```

```

ProgressBar1.Value = 5

```

```

For i = 65 To 256
    A3(i, i) = 1
Next i

```

```

k = 1                'A4 matrisi oluşturuluyor
For i = 1 To 16
For j = 1 To 2
    A4(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A4(k, i + 16) = -0.5
    Else
        A4(k, i + 16) = 0.5
    End If

```

```

    k = k + 1
Next j
Next i

```

```

For i = 33 To 256
    A4(i, i) = 1
Next i

```

```

k = 1                'A5 matrisi oluşturuluyor
For i = 1 To 8
For j = 1 To 2
    A5(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A5(k, i + 8) = -0.5
    Else
        A5(k, i + 8) = 0.5
    End If
    k = k + 1
Next j
Next i

```

```

ProgressBar1.Value = 6

```

```

For i = 17 To 256
    A5(i, i) = 1
Next i

```

```

k = 1                'A6 matrisi oluşturuluyor
For i = 1 To 4
For j = 1 To 2
    A6(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A6(k, i + 4) = -0.5
    Else
        A6(k, i + 4) = 0.5
    End If
    k = k + 1
Next j
Next i

```

```

ProgressBar1.Value = 7

```

```

For i = 9 To 256
    A6(i, i) = 1
Next i

```

```

k = 1                'A7 matrisi oluşturuluyor
For i = 1 To 2
For j = 1 To 2
    A7(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A7(k, i + 2) = -0.5
    Else
        A7(k, i + 2) = 0.5
    End If
    k = k + 1
Next j
Next i

ProgressBar1.Value = 10

For i = 5 To 256
    A7(i, i) = 1
Next i

ProgressBar1.Value = 11
k = 1                'A8 matrisi oluşturuluyor
For i = 1 To 1
For j = 1 To 2
    A8(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A8(k, i + 1) = -0.5
    Else
        A8(k, i + 1) = 0.5
    End If
    k = k + 1
Next j
Next i

For i = 3 To 256
    A8(i, i) = 1
Next i

ProgressBar1.Value = 15

For i = 1 To 256    'A1*A2 oluşturuluyor
For j = 1 To 256
    AA(i, j) = A1(i, j)
    BB(i, j) = A2(i, j)
Next j
Next i
carpim 256, 256, 256

```

atama

ProgressBar1.Value = 16

```
For i = 1 To 256    'W*A3 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A3(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 17

```
For i = 1 To 256    'W*A4 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A4(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 18

```
For i = 1 To 256    'W*A5 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A5(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 19

```
For i = 1 To 256    'W*A6 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A6(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 20

```
For i = 1 To 256    'W*A7 oluşturuluyor
For j = 1 To 256
```

```

    AA(i, j) = W(i, j)
    BB(i, j) = A7(i, j)
Next j
Next i
carpim 256, 256, 256
atama

```

ProgressBar1.Value = 21

```

For i = 1 To 256    'W*A8 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A8(i, j)
Next j
Next i
carpim 256, 256, 256
atama

```

transpoze\_al 'W 'nin transpozmesini alır

ProgressBar1.Value = 22

```

For i = 1 To 256    'W*A8 oluşturuluyor
For j = 1 To 256
    AA(i, j) = WT(i, j)
    BB(i, j) = A(i, j)
Next j
Next i
carpim 256, 256, 256
atama1

```

```

For i = 1 To 256    'W*A8 oluşturuluyor
For j = 1 To 256
    AA(i, j) = WTA(i, j)
    BB(i, j) = W(i, j)
Next j
Next i
carpim 256, 256, 256
atama2

```

ProgressBar1.Value = 23

```

For i = 1 To 256    'W 'nin tersi hesaplanıyor
For j = 1 To 256
    If W(i, j) = 0 Then
        Q(j, i) = 0
    ElseIf W(i, j) > 0 Then
        Q(j, i) = 1

```

```

    ElseIf W(i, j) < 0 Then
        Q(j, i) = -1
    End If

Next j
Next i

ProgressBar1.Value = 24

For i = 1 To 256
    For j = 1 To 256
        QT(i, j) = Q(j, i)
    Next j
Next i

For i = 1 To 256
    For j = 1 To 256
        AA(i, j) = QT(i, j)
        BB(i, j) = T(i, j)
    Next j
Next i
carpim 256, 256, 256
atama3

For i = 1 To 256
    For j = 1 To 256
        AA(i, j) = QTT(i, j)
        BB(i, j) = Q(i, j)
    Next j
Next i
carpim 256, 256, 256
atama4

For i = 0 To 255
    For j = 0 To 255
        Picture2.PSet (i, j), RGB(Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))))
    Next j
Next i
MousePointer = 1
ProgressBar1.Value = 25
ProgressBar1.Visible = False
StatusBar1.Panels(1).Text = "H a z ı r . . ."
Label4.Caption = "Sıkıştırılmış Görüntü"
bitir:
End Sub

Private Sub Command12_Click()
    CommonDialog2.DialogTitle = "Uzantısız Bir Dosya İsmi Yazınız"
    CommonDialog2.Filter = "Bitmap Dosyaları|*.BMP"

```

```

CommonDialog2.Action = 2
CommonDialog2.FileName = CommonDialog2.FileName & ".bmp"
If CommonDialog2.FileName = "" Then
    GoTo son
End If
Picture3.Picture = Picture2.Image

SavePicture Picture3.Picture, CommonDialog2.FileName
son:
End Sub

Private Sub Command2_Click()
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11

    For i = 0 To 255 'Görüntü A matrisine yükleniyor.
        For j = 0 To 255
            Gercek_A(i + 1, j + 1) = Point(Image1.Left + i, Image1.Top + j)

            kirmizi_band = Gercek_A(i + 1, j + 1) And 255
            A(i + 1, j + 1) = kirmizi_band
        Next j
    Next i

    For i = 0 To 255
        For j = 0 To 255
            Picture1.PSet (i, j), RGB(A(i + 1, j + 1), A(i + 1, j + 1), A(i + 1, j + 1))
        Next j
    Next i
    Label3.Caption = "Grinin Tonlarına Çevrilmiş Görüntü"
    MousePointer = 1
    StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

Private Sub Command3_Click()
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11
    Form2.MSFlexGrid1.AllowUserResizing = flexResizeBoth

    Form2.Label1.Caption = "Sayısal Olarak Basitleştirilmiş Görüntü Matrisi"
    Form2.MSFlexGrid1.Cols = 257
    Form2.MSFlexGrid1.Rows = 257
    Form2.MSFlexGrid1.FixedCols = 1
    Form2.MSFlexGrid1.FixedRows = 1
    For i = 1 To 256
        Form2.MSFlexGrid1.Row = i
        Form2.MSFlexGrid1.Col = 0
        Form2.MSFlexGrid1.Text = i
        Form2.MSFlexGrid1.Row = 0
        Form2.MSFlexGrid1.Col = i
    
```

```

        Form2.MSFlexGrid1.Text = i
    Next i

    For i = 1 To 256
    For j = 1 To 256
        Form2.MSFlexGrid1.Row = i
        Form2.MSFlexGrid1.Col = j
        Form2.MSFlexGrid1.Text = T(i, j)
    Next
    Next
    MousePointer = 1
    Form2.Show
    StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

Private Sub Command4_Click()
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11

    Form2.MSFlexGrid1.AllowUserResizing = flexResizeBoth

    Form2.Label1.Caption = "Gerçek Görüntü Matrisi"
    Form2.MSFlexGrid1.Cols = 257
    Form2.MSFlexGrid1.Rows = 257
    Form2.MSFlexGrid1.FixedCols = 1
    Form2.MSFlexGrid1.FixedRows = 1
    For i = 1 To 256

        Form2.MSFlexGrid1.Row = i
        Form2.MSFlexGrid1.Col = 0
        Form2.MSFlexGrid1.Text = i
        Form2.MSFlexGrid1.Row = 0
        Form2.MSFlexGrid1.Col = i
        Form2.MSFlexGrid1.Text = i

    Next i

    For i = 1 To 256
    For j = 1 To 256
        Form2.MSFlexGrid1.Row = i
        Form2.MSFlexGrid1.Col = j
        Form2.MSFlexGrid1.Text = Gercek_A(i, j)
    Next
    Next
    MousePointer = 1
    Form2.Show
    StatusBar1.Panels(1).Text = "H a z ı r . . ."
End Sub

```

```
Private Sub Command5_Click()
```

```
    ProgressBar1.Visible = True
    ProgressBar1.Value = 0
```

```
    For i = 0 To 255 'Görüntü A matrisine yükleniyor.
        For j = 0 To 255
            Gercek_A(i + 1, j + 1) = Point(Image1.Left + i, Image1.Top + j)

            kirmizi_band = Gercek_A(i + 1, j + 1) And 255
            A(i + 1, j + 1) = kirmizi_band
        Next j
    Next i
```

```
    StatusBar1.Panels(1).Text = "S ı k ı ş t ı r ı l ı y o r . . ."
    MousePointer = 11
    For i = 1 To 256 'Tüm matrisler ilk kullanım için sıfırlanıyor
        For j = 1 To 256
```

```
            S(i, j) = 0
```

```
            W(i, j) = 0
            T(i, j) = 0
            A1(i, j) = 0
            A2(i, j) = 0
            A3(i, j) = 0
            A4(i, j) = 0
            A5(i, j) = 0
            A6(i, j) = 0
            A7(i, j) = 0
            A8(i, j) = 0
            AA(i, j) = 0
            BB(i, j) = 0
            C(i, j) = 0
            WT(i, j) = 0
            WTA(i, j) = 0
            Q(i, j) = 0
            QT(i, j) = 0
            QTT(i, j) = 0
            SKM(i, j) = 0
```

```
        Next j
    Next i
```

```
    ProgressBar1.Value = 1
```

```
    k = 1 'A1 matrisi oluşturuluyor
    For i = 1 To 128
```

```

For j = 1 To 2
  A1(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A1(k, i + 128) = -0.5
  Else
    A1(k, i + 128) = 0.5
  End If
  k = k + 1
Next j
Next i

```

```

k = 1          'A2 matrisi oluşturuluyor
For i = 1 To 64
For j = 1 To 2
  A2(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A2(k, i + 64) = -0.5
  Else
    A2(k, i + 64) = 0.5
  End If
  k = k + 1
Next j
Next i

```

```

For i = 129 To 256
  A2(i, i) = 1
Next i

```

```

k = 1          'A3 matrisi oluşturuluyor
For i = 1 To 32
For j = 1 To 2
  A3(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A3(k, i + 32) = -0.5
  Else
    A3(k, i + 32) = 0.5
  End If
  k = k + 1
Next j
Next i

```

```

For i = 65 To 256
  A3(i, i) = 1

```

Next i

```

k = 1                'A4 matrisi oluşturuluyor
For i = 1 To 16
For j = 1 To 2
  A4(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A4(k, i + 16) = -0.5
  Else
    A4(k, i + 16) = 0.5
  End If
  k = k + 1
Next j
Next i

```

```

For i = 33 To 256
  A4(i, i) = 1
Next i

```

```

k = 1                'A5 matrisi oluşturuluyor
For i = 1 To 8
For j = 1 To 2
  A5(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A5(k, i + 8) = -0.5
  Else
    A5(k, i + 8) = 0.5
  End If
  k = k + 1
Next j
Next i

```

```

For i = 17 To 256
  A5(i, i) = 1
Next i

```

```

k = 1                'A6 matrisi oluşturuluyor
For i = 1 To 4
For j = 1 To 2
  A6(k, i) = 0.5
  M = k / 2
  N = Int(M)
  If M = N Then
    A6(k, i + 4) = -0.5

```

```

        Else
            A6(k, i + 4) = 0.5
        End If
        k = k + 1
    Next j
Next i

```

```

For i = 9 To 256
    A6(i, i) = 1
Next i

```

```

k = 1                'A7 matrisi oluşturuluyor
For i = 1 To 2
For j = 1 To 2
    A7(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A7(k, i + 2) = -0.5
    Else
        A7(k, i + 2) = 0.5
    End If
    k = k + 1
Next j
Next i

```

```

For i = 5 To 256
    A7(i, i) = 1
Next i

```

```

k = 1                'A8 matrisi oluşturuluyor
For i = 1 To 1
For j = 1 To 2
    A8(k, i) = 0.5
    M = k / 2
    N = Int(M)
    If M = N Then
        A8(k, i + 1) = -0.5
    Else
        A8(k, i + 1) = 0.5
    End If
    k = k + 1
Next j
Next i

```

```

For i = 3 To 256
    A8(i, i) = 1
Next i

```

ProgressBar1.Value = 4

```
For i = 1 To 256    'A1*A2 oluşturuluyor
For j = 1 To 256
    AA(i, j) = A1(i, j)
    BB(i, j) = A2(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 5

```
For i = 1 To 256    'W*A3 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A3(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 6

```
For i = 1 To 256    'W*A4 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A4(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 8

```
For i = 1 To 256    'W*A5 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A5(i, j)
Next j
Next i
carpim 256, 256, 256
atama
```

ProgressBar1.Value = 9

```
For i = 1 To 256    'W*A6 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A6(i, j)
```

```

Next j
Next i
carpim 256, 256, 256
atama

```

```

ProgressBar1.Value = 11

```

```

For i = 1 To 256    'W*A7 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A7(i, j)
Next j
Next i
carpim 256, 256, 256
atama

```

```

ProgressBar1.Value = 13

```

```

For i = 1 To 256    'W*A8 oluşturuluyor
For j = 1 To 256
    AA(i, j) = W(i, j)
    BB(i, j) = A8(i, j)
Next j
Next i
carpim 256, 256, 256
atama

```

```

transpoze_al        'W 'nin transpozesini alır

```

```

ProgressBar1.Value = 15

```

```

For i = 1 To 256    'WT*A oluşturuluyor
For j = 1 To 256
    AA(i, j) = WT(i, j)
    BB(i, j) = A(i, j)
Next j
Next i
carpim 256, 256, 256
atama1

```

```

ProgressBar1.Value = 18

```

```

For i = 1 To 256    'WTA*W oluşturuluyor
For j = 1 To 256
    AA(i, j) = WTA(i, j)
    BB(i, j) = W(i, j)
Next j
Next i
carpim 256, 256, 256
atama2

```

```
ProgressBar1.Value = 20
```

```
For i = 1 To 256
For j = 1 To 256
    T1(i, j) = Int(T1(i, j))
Next j
Next i
```

```
For i = 1 To 256      ' küçük değerler ihmal ediliyor
For j = 1 To 256
    If Abs(T1(i, j)) <= Abs(Val(Text1.Text)) Then
        T(i, j) = 0
    Else
        T(i, j) = T1(i, j)
    End If
```

```
Next j
Next i
```

```
sikisirme_orani = 0
```

```
For i = 1 To 256
For j = 1 To 256
    If T(i, j) <> 0 Then
        Sikistirme_Orani = Sikistirme_Orani + 1
    End If
Next j
Next i
```

```
For i = 1 To 256      'W 'nin tersi hesaplanıyor
For j = 1 To 256
    If W(i, j) = 0 Then
        Q(j, i) = 0
    ElseIf W(i, j) > 0 Then
        Q(j, i) = 1
    ElseIf W(i, j) < 0 Then
        Q(j, i) = -1
    End If
```

```
Next j
Next i
```

```
ProgressBar1.Value = 21      'W'nin transpozesi hesaplanıyor.
```

```
For i = 1 To 256
For j = 1 To 256
    QT(i, j) = Q(j, i)
Next j
Next i
```

```

For i = 1 To 256
For j = 1 To 256
    AA(i, j) = QT(i, j)
    BB(i, j) = T(i, j)
Next j
Next i
carpim 256, 256, 256
atama3

ProgressBar1.Value = 23

For i = 1 To 256
For j = 1 To 256
    AA(i, j) = QTT(i, j)
    BB(i, j) = Q(i, j)
Next j
Next i
carpim 256, 256, 256
atama4

ProgressBar1.Value = 24

For i = 0 To 255
For j = 0 To 255
    Picture2.PSet (i, j), RGB(Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j + 1))))
Next j
Next i

MousePointer = 1

ProgressBar1.Value = 25
ProgressBar1.Visible = False
StatusBar1.Panels(1).Text = "Sıkıştırma Oranı Yaklaşık 1 'e " &
Int(Sıkıştırma_Oranı)
Label4.Caption = "Sıkıştırılmış Görüntü"
End Sub

Private Sub Command6_Click()
    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."
    MousePointer = 11

    For i = 0 To 255
    For j = 0 To 255

```

```

        Picture1.PSet (i, j), RGB(Int(Abs(T(i + 1, j + 1))), Int(Abs(T(i + 1, j + 1))),
Int(Abs(T(i + 1, j + 1))))

```

```

    Next j

```

```

    Next i

```

```

    Label3.Caption = "Transforme olmuş Görüntü"

```

```

    MousePointer = 1

```

```

    StatusBar1.Panels(1).Text = "H a z ı r . . ."

```

```

End Sub

```

```

Private Sub Command7_Click()

```

```

    Picture1.Picture = LoadPicture(Empty)

```

```

    Picture2.Picture = LoadPicture(Empty)

```

```

    Label3.Caption = ""

```

```

    Label4.Caption = ""

```

```

End Sub

```

```

Private Sub Command8_Click()

```

```

    StatusBar1.Panels(1).Text = "Y ü k l e n i y o r . . ."

```

```

    MousePointer = 11

```

```

    For i = 0 To 255

```

```

        For j = 0 To 255

```

```

            Picture2.PSet (i, j), RGB(Int(Abs(SKM(i + 1, j + 1))), Int(Abs(SKM(i + 1, j +
1))), Int(Abs(SKM(i + 1, j + 1))))

```

```

        Next j

```

```

    Next i

```

```

    Label4.Caption = "Sıkıştırılmış Görüntü"

```

```

    MousePointer = 1

```

```

    StatusBar1.Panels(1).Text = "H a z ı r . . ."

```

```

End Sub

```

```

Private Sub Command9_Click()

```

```

    Unload Me

```

```

End Sub

```

```

Private Sub Form_Load()

```

```

    ProgressBar1.Visible = False

```

```

    ProgressBar1.Min = 0

```

```

    ProgressBar1.Max = 25

```

```

    Image1.Stretch = True

```

```

    Image1.Height = 256

```

```

    Image1.Width = 256

```

```

    Picture1.Height = 256

```

```

    Picture1.Width = 256

```

```

    Picture1.ScaleHeight = 256

```

```
Picture1.ScaleWidth = 256
```

```
Picture2.Height = 256
Picture2.Width = 256
Picture2.ScaleHeight = 256
Picture2.ScaleWidth = 256
```

```
Picture3.Height = 256
Picture3.Width = 256
Picture3.ScaleHeight = 256
Picture3.ScaleWidth = 256
```

```
StatusBar1.Style = sbrNormal
StatusBar1.Panels(1).AutoSize = sbrContents
StatusBar1.Panels(1).Text = "H a z ı r . . ."
```

```
Form1.ScaleMode = 3
Picture1.ScaleMode = 3
Picture2.ScaleMode = 3
Picture3.ScaleMode = 3
```

```
CommonDialog1.Flags = cdlOFNHideReadOnly
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Unload Form2
End Sub
```

## **Programın Ana Modülü**

```
Option Explicit
Option Base 1
Public A(256, 256) As Variant
Public Gercek_A(256, 256) As Variant
Public T(256, 256) As Variant
Public T1(256, 256) As Variant
Public S(256, 256) As Variant
Public W(256, 256) As Variant    "Transformasyon Matrisi"
Public A1(256, 256) As Variant
Public A2(256, 256) As Variant
Public A3(256, 256) As Variant
Public A4(256, 256) As Variant
Public A5(256, 256) As Variant
Public A6(256, 256) As Variant
```

```

Public A7(256, 256) As Variant
Public A8(256, 256) As Variant
Public ii As Integer
Public jj As Integer
Public kk As Integer
Public AA(256, 256) As Variant
Public BB(256, 256) As Variant
Public C(256, 256) As Variant
Public WT(256, 256) As Variant
Public WTA(256, 256) As Variant
Public Q(256, 256) As Variant
Public QT(256, 256) As Variant
Public QTT(256, 256) As Variant
Public SKM(256, 256) As Variant
Public x1 As Object
Public Sub carpin(iii, kkk, jjj)

    For ii = 1 To iii
        For kk = 1 To kkk
            C(ii, kk) = 0
            For jj = 1 To jjj

                C(ii, kk) = C(ii, kk) + AA(ii, jj) * BB(jj, kk)

            Next jj

        Next kk

    Next ii

End Sub
Public Sub atama()
    For ii = 1 To 256
        For jj = 1 To 256
            W(ii, jj) = C(ii, jj)
        Next
    Next
End Sub
Public Sub transpose_al()
    For ii = 1 To 256
        For jj = 1 To 256
            WT(ii, jj) = W(jj, ii)
        Next
    Next
End Sub
Public Sub atama1()
    For ii = 1 To 256
        For jj = 1 To 256
            WTA(ii, jj) = C(ii, jj)
        Next
    Next

```

```
Next
End Sub
Public Sub atama2()
    For ii = 1 To 256
        For jj = 1 To 256
            T1(ii, jj) = C(ii, jj)
        Next
    Next
End Sub
Public Sub atama3()
    For ii = 1 To 256
        For jj = 1 To 256
            QTT(ii, jj) = C(ii, jj)
        Next
    Next
End Sub
Public Sub atama4()
    For ii = 1 To 256
        For jj = 1 To 256
            SKM(ii, jj) = C(ii, jj)
        Next
    Next
End Sub
```