

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**DAĞITIK SİSTEMLER İÇİN ADA UYDU GÖRÜNTÜLERİNİN
VEKTÖREL MODELLENMESİ VE ZAMAN-MEKÂNSAL
ANALİZİ**

SÜLEYMAN EKEN

KOCAELİ 2012

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ

YÜKSEK LİSANS TEZİ

**DAĞITIK SİSTEMLER İÇİN ADA UYDU GÖRÜNTÜLERİNİN
VEKTÖREL MODELLENMESİ VE ZAMAN-MEKÂNSAL
ANALİZİ**

SÜLEYMAN EKEN

**Yrd.Doç.Dr.Ahmet SAYAR
Danışman, Kocaeli Üniv.
Prof.Dr. Yaşar BECERİKLİ
Jüri Üyesi, Kocaeli Üniv.
Doç.Dr. Müfit ÇETİN
Jüri Üyesi, Yalova Üniv.**


.....
.....
.....

Tezin Savunulduğu Tarih: 27.06.2012

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada Landsat-7 uydu görüntüleri renk uzayı dönüşümü segmentasyonu, bölge gruplama-etiketleme ve sınır piksellerinin takibine dayalı vektörleştirme işlemine tabi tutularak poligon-vektör modellerinin oluşturulması sağlanmıştır. Daha sonra poligon-vektör modelinin nesne ilişkisel veritabanları üzerinden zaman-mekânsal (spatio-temporal), topolojik analizlerinin yapılması ve dağıtık sistemlerde farklı büyüklükteki görüntülerin raster ve vektörel olarak taşınmasındaki yanıt sürelerine göre performans analizleri yapılmıştır.

Yüksek lisans eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarına yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygıdeğer hocam Yard. Doç. Dr. Ahmet SAYAR'a, değerli görüşleri ile tez çalışmama katkıda bulunan saygıdeğer hocam Prof. Dr. Yaşar BECERİKLİ'ye, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümüne, tez çalışmasının başından sonuna kadar manevi desteklerini benden esirgemeyen tüm iş arkadaşlarıma teşekkürlerimi sunarım.

Ayrıca maddi ve manevi desteklerini tüm hayatı boyunca esirgemeyen merhume annem Hayriye EKEN'e, babam Nurettin EKEN'e ve kardeşime şükranlarımı sunarım.

Haziran – 2012

Süleyman Eken

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER.....	ii
ŞEKİLLER DİZİNİ	iv
TABLolar DİZİNİ	v
SİMGELER DİZİNİ ve KISALTMALAR	vi
ÖZET	vii
ABSTRACT	viii
GİRİŞ	1
1. GENEL BİLGİLER	3
1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri	3
1.2. Tez Çalışmasının Katkıları.....	4
1.3. Literatür Taraması	4
1.4. Tezin Yapısı	9
2. UZAKTAN ALGILAMA GÖRÜNTÜ İŞLEME.....	10
2.1. Görüntü İşleme	10
2.2. Gürültü Giderme ve Kenar İyileştirme	12
2.3. Uzaktan Algılama	14
2.4. Uydu Görüntüleri.....	21
2.5. Mekânsal Veritabanları.....	23
2.6. Raster ve Vektör Veri	26
3. MİMARİ: ADA UYDU GÖRÜNTÜLERİNİN VEKTÖRİZASYONU VE MEKÂNSAL VERİTABANLARINDA ANALİZİ	30
3.1. Veri Ön İşleme ve Yumuşatma	32
3.2. Renk Uzayı Dönüşümleri ile Ada Dışındaki Nesnelere Yok Etme	33
3.3. Bölge Gruplama ve Etiketleme	36
3.4. Morfolojik Kapanma İşlemi.....	37
3.5. Sınır Çıkarımı ve Vektörizasyon	38
3.6. Vektör Verileri Mekânsal Veritabanlarına Kaydetme	39
3.7. Geo-referanslama	40
4. ADA UYDU GÖRÜNTÜLERİ ÜZERİNE DURUM ÇALIŞMASI: MEKÂNSAL VERİTABANLARI ÜZERİNDE TOPOLOJİK SORGULAR VE DEFORMASYON ANALİZLERİ.....	43
4.1. Giriş	43
4.2. Ada Uydu Görüntüsünün Vektörel Modellenmesi.....	43
4.3. Vektör Model Üzerinde Topolojik Sorgular ve Deformasyon Analizleri	45
4.4. Sonuç	47
5. DAĞITIK SİSTEMLERDE GÖRÜNTÜNÜN TAŞINMASI RASTER VE VEKTÖR PERFORMANS ANALİZLERİ.....	48
5.1. Giriş	48
5.2. Raster ve Vektör Veri İletimi	48
5.3. Web Servisleri	49
5.4. Raster ve Vektör Veri İletimi İçin Dağıtık Bir Model.....	49

5.5. Önerilen Dağıtık Model ile Raster ve Vektör Veri Taşınması Performans Analizleri	53
5.6. Sonuç	56
6. SONUÇLAR VE ÖNERİLER.....	57
KAYNAKLAR.....	60
EKLER	60
KİŞİSEL YAYINLAR VE ESERLER	77
ÖZGEÇMİŞ.....	79

ŞEKİLLER DİZİNİ

Şekil 1.1. Kesişimlerdeki inceltme bozulmaları.....	6
Şekil 2.1. Uzaktan algılama sistemlerinin temelleri	15
Şekil 2.2. Gerçek dünyadaki verilerin vektörel temsilleri.....	27
Şekil 2.3. Coğrafi veri modellerinin gösterimi	27
Şekil 3.1. Örnek bir mozağın meta verileri	30
Şekil 3.2. Önerilen yöntemin akış şeması	31
Şekil 3.3. Uydu görüntülerindeki beş element	34
Şekil 3.4. Bölge gruplama ve etiketleme	37
Şekil 3.5. 8-yönlü komşulukve izinli yönler	39
Şekil 3.6. Sınır piksellerinin vektörleştirilmesi	39
Şekil 3.7. Dönüşüm işlemi	41
Şekil 4.1. Üzerindeçalışma yapılacak bölge.....	44
Şekil 4.2. Landsat-7 uydu görüntüsünün otomatik vektörizasyonu	44
Şekil 4.3. Alan ve çevre hesaplama örnek kodları.....	46
Şekil 4.4. Uydu görüntüsüne uygulanan bazı senaryolar	46
Şekil 5.1. Web servisleri aracılığıyla veri iletimi mimarisinin prototipi	51
Şekil 5.2. Gökçeada uydu görüntüsünün parçaları	53
Şekil 5.3. Vektörleştirme işlemi olmadan gerekli yanıt süreleri	53
Şekil 5.4. Vektörleştirme işlemi yapılırca gerekli yanıt süreleri	54
Şekil 5.5. Nokta iletimi için gerekli yanıt süresi	54
Şekil 5.6. Vektörizasyon sonrası harita oluşumu için geçen süre	55
Şekil 5.7. Vertex iletimi sonrası harita oluşumu için geçen süre	55

TABLULAR DİZİNİ

Tablo 2.1. Landsat TM bant özellikleri ve kullanım alanları.....	18
Tablo 2.2. Uydu sistemleri ve özellikleri	21
Tablo 2.3. Raster-vektör veri karşılaştırılması	29
Tablo 3.1. Beş element bileşenleri arasındaki ilişki	35
Tablo 3.2. Bazı mekânsal operatörler	40
Tablo 4.1. Gerçekleştirilen simülasyonların sonuçları	47

SİMGELER DİZİNİ VE KISALTMALAR

Kısaltmalar

ANN	: Artificial Neural Network (Yapay Sinir Ağı)
API	: Application Programming Interface (Uygulama Programı Arabirimi)
CAM	: Changes Accumulation Model (Değişim Birikim Modeli)
ÇOBAN	: Çok Bantlı Görüntüleyici
DBMS	: Database Management Systems (Veritabanı Yönetim Sistemleri)
EMR	: Electromagnetic Radiation (Elektromanyetik Radyasyon)
ETM+	: Enhanced Thematic Mapper Plus (Gelişmiş Tematik Haritalayıcı)
GEZGİN	: Gerçek Zamanlı Görüntü İşleyici
GIS	: Geographical Information System (Coğrafi Bilgi Sistemi)
GPS	: Global Positioning System (Küresel Konumlandırma Sistemi)
HRV	: High Resolution Visualization (Yüksek Çözünürlüklü Görüntüleme)
HSV	: Hue Saturation Value (Renk Doymuluk Gri)
HT	: Hough Transformation (Hough Dönüşüm)
IFOV	: Instantaneous Field of View (Anlık Görüş Alanı)
IMS	: Information Management System (Bilgi Yönetim Sistemi)
MSm	: Multi Spectral mode (Çok Bantlı mod)
MSS	: Multi Spectral Scanner (Çok Bantlı Spektral Tarayıcı)
OCR	: Optical Character Recognition (Optik Karakter Tanıma)
OZZ	: Orthogonal Zig Zag (Dikey Zig Zag)
PAN	: Panchromatic (Siyah-Beyaz)
RBV	: Return Beam Vidicon (Vidikon TV Kamerası)
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
RS	: Remote Sensing (Uzaktan Algılama)
SOA	: Service Oriented Architecture (Servis Tabanlı Mimari)
SPOT	: Système Pour l'Observation de la Terre (Dünya Gözlem Sistemi)
SQL	: Structured Query Language (Yapısal Sorgulama Dili)
SVG	: Scalable Vector Graphics (Ölçeklenebilir Vektör Grafikleri)
TIFF	: Tagged Image File Format (Taglı Resim Dosya Formatı)
TIN	: Triangular Irregular Network (Üçgenlenmiş Düzensiz Ağ)
TM	: Thematic Mapper (Tematik Haritalayıcı)
XML	: Extensible Markup Language (Genişletilebilir İşaret Dili)

DAĞITIK SİSTEMLER İÇİN ADA UYDU GÖRÜNTÜLERİNİN VEKTÖREL MODELLENMESİ VE ZAMAN-MEKÂNSAL ANALİZİ

ÖZET

Yerküre dinamik bir yapı olarak sürekli bir deęişim halindedir. Kıtlıklar, kuraklık, seller, şehirleşme, bu deęişimleri tetikleyen etmenlerden sadece birkaçıdır. Son yıllardaki uydu teknolojisindeki gelişmeler yerküre üzerindeki doğal ya da insan kaynaklı etmenlerin sebep olduęu deęişimlerin takip edilmesi ve incelenmesine olanak sağlamaktadır. Uydu görüntüleriyle ilgili çalışmalar uzaktan algılama olarak adlandırılmakta, gerek akademik ve ticari hayatta gerekse kamusal ve özel alanda birçok çalışmaya konu olmaktadır. Bu çalışmalar yerküre üzerindeki deęişimlerin zaman içindeki analizleri (şehirleşme, yeşil alanlar vb.) ve dięer birçok özellikler hakkında bilgi edinilmesini sağlamıştır. Ancak bu tür çalışmaların hemen hepsinde görüntüler raster olarak ele alınmış ve teknikler bunun üzerine geliştirilmiştir. Bizim önerdiğimiz vektör modellemede ise piksel-grid yapısından kurtulup bilgisayarın daha iyi anlayıp daha hızlı işlemlerini yapabileceęi gerçek veriler üzerinde vektör temsilin oluşturulmasıdır. Bu yaklaşımla veri üzerinde yapılabilecek analizler (topolojik ve zaman-mekânsal) kısmi olarak sınırlansa da birçok web tabanlı CBS için performans ve servis kalitesi açısından kazanımların büyük olacağı beklenmektedir. İnternet üzerindeki sınırlı bant genişlięi ve uydu görüntülerinin aşırı büyük olması ile gerçekleştirilemeyen bazı servis odaklı dağıtık sistem uygulamalarının gerçekleştirilebilmesinde kolaylık sağlanacaktır.

Anahtar Kelimeler: Dağıtık Sistemler için Veri Modelleme, Dağıtık Veri İletimi, Raster ve Vektör Veri, Uydu Görüntüleri, Vektörizasyon.

VECTOR MODELING OF ISLAND SATELLITE IMAGES FOR DISTRIBUTED SYSTEMS AND SPATIO-TEMPORAL ANALYSIS

ABSTRACT

The Earth is in constant change as a dynamic structure. Famines, droughts, floods, and urbanization are only a few of the factors that trigger these changes. Developments in satellite technology in recent years allow to be followed and examined changes caused by natural or human-induced on the Earth. Studies related to satellite images are called remote sensing and it has been the subject of both academic and commercial life or public and private areas. These studies provide information about analysis of changes on the Earth over time (urbanization, green areas, etc.) and many other features. However, images have been regarded as raster form in almost all of such studies and techniques have been developed over it. Vector modeling we suggested is getting rid of the structure of the pixel-grid modeling. So it results computer can process a better and faster on actual data. Although analysis (topological and spatio-temporal) done on data via suggested approach are constrained partially, it is expected to give better performance gain for many web-based GIS applications in terms of performance and service quality. Some service-oriented distributed system applications are not carried out because of limited bandwidth on the internet and satellite images with tremendous file size can be archived easily.

Keywords: Data Modeling for Distributed Systems, Distributed Data Transmission, Raster and Vector Data, Satellite Images, Vectorization.

GİRİŞ

Bilgisayarlarda görüntü; raster grafik denilen gri renk ton değerlerini içeren iki boyutlu bir dizi formatında veya vektör grafik denilen nokta, doğru, poligon gibi matematiksel olarak ifade edilen geometrik primitifler yolu ile olmak üzere iki şekilde temsil edilir. Raster görüntüde (örneğin TIFF görüntü) görüntüyü oluşturan bütün piksellerin lokasyon ve renk bilgileri tutulduğu için dosyanın boyutu çok büyük olmaktadır. Aynı görüntünün vektörel temsilinde (örneğin text dosyasındaki x,y dizileri) görüntünün matematiksel tanımlaması tutulduğu için dosya boyutu orijinalinden onlarca hatta binlerce kat daha küçük boyuta sahip olabilmektedir.

Rasterdan vektöre dönüşüm diğer bir deyişle vektörizasyon; Coğrafi Bilgi Sistemleri (CBS) ve Uzaktan Algılama görüntü işleme uygulamalarında, mühendislik çizimlerini analog biçimden sayısal ortama aktarmada, doküman analizi ve tanımada önemli bir işleve sahiptir. Geleneksel olarak harita ve diğer dokümanların vektörizasyonu manüel olarak bilgisayarda fare yardımıyla sayısallaştırma tablosu üzerinde yapıldı. Son zamanlarda çeşitli yazılımların geliştirilmesiyle birlikte vektörizasyon işlemi yarı otomatik veya tam otomatik hale gelmiştir.

Manüel işlemde tam otomatik sisteme geçişi sağlamak için görüntü işleme, görüntü analizi ve görüntü tanıma alanlarındaki birçok problemi çözmek için son 30 yıl boyunca yeni yöntemler geliştirilmeye çalışıldı; fakat günümüzde bütün problemlerin tam anlamıyla çözüldüğü söylenemez. Tombré [1]:

“Aslında metotlar iş yapar; fakat bunların hiçbiri mükemmel değildir. ... Bu metotlar iyi sonuçlar vermesine rağmen, hepsinin kendine özgü zayıflıkları vardır. Bundan dolayı mükemmel bir raster-vektör dönüşümünün mevcudiyeti söylenemez” demıştır.

İnternet, Coğrafi Bilgi Sistemleri kullanıcıları için mekânsal veri (raster veya vektör veri) transfer etmek, analizler gerçekleştirmek ve coğrafi sonuçları göstermek adına

günümüzde önemli bir içerik yayınlama ortamı haline gelmiştir. Web teknolojilerinin gelişmesiyle birlikte coğrafi bilgiye dayalı uygulamaların sayısı artmıştır. Diğer yandan, hızla artan mekânsal veri miktarı ile network bant genişliği arasında ters ilişki vardır. Cevap süresinin (istemcinin sunucudan mekânsal veri istemesinden veriyi almasına kadar geçen süre) kısa olması web-CBS uygulamalarında önemli bir unsurdur. Uydu görüntülerinin piksel ızgarası şeklinde (raster) tutulmasından ziyade vektörel olarak modellenmesi; aynı verinin daha az yer kaplaması, internet üzerinden daha hızlı taşınması ve daha hızlı işlenmesi ve yorumlanmasına olanak vermektedir.

Bu çalışmada Landsat-7 ada uydu görüntülerinin poligon-vektör modelleri oluşturulmuştur. Daha sonra poligon-vektör modelinin nesne ilişkisel veritabanları üzerinden zaman-mekânsal (spatio-temporal), topolojik analizlerinin yapılması ve dağıtık sistemlerde görüntünün raster ve vektörel olarak taşınmasının performans analizleri yapılmıştır. Bu çalışma diğer objelerin de vektör modellemesi çalışmalarına temel teşkil edecektir.

1. GENEL BİLGİLER

1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri

Bu çalışmada Landsat-7 uydu görüntüleri renk uzayı dönüşümü segmentasyonu, bölge gruplama-etiketleme ve sınır piksellerinin takibine dayalı vektörleştirme işlemine tabi tutularak poligon-vektör modelleri oluşturulmaya çalışılmıştır. Daha sonra poligon-vektör modelinin nesne ilişkisel veritabanları üzerinden zaman-mekânsal (spatio-temporal), topolojik analizlerinin yapılması ve dağıtık sistemlerde görüntünün raster ve vektörel olarak taşınmasının performans analizleri yapılmıştır. Ada görüntülerinin seçilmesinin nedeni, adaların diğer mekânsal objelerden (ülke, şehir, nehir, yol gibi) görsel olarak uydu görüntüleri üzerinden ayırt edilebilmeleridir. Bununla birlikte, bu çalışma diğer objelerin de vektör modellemesi çalışmalarına temel teşkil edecektir.

Uydu görüntüleri üzerinde yapılan zaman-mekânsal ya da topolojik analizlerde (komşuluk, kesişme vb.) piksel bazlı bilgiden ziyade nokta setleriyle tanımlanan şekil koordinat bilgilerini kullanmak hızlı ve pratik olması yönünden daha çok tercih edilmektedir. Bu çalışmada Landsat uydu görüntüleri öncelikle poligon nesnesi olarak modellenerek bilgisayarların ve bilgi sistemlerinin kolayca anlayabileceği ve işleyebileceği formata getirilmiştir. Adanın poligon temsili daha sonra nesne-ilişkisel veritabanlarında depolanabilir ve işlenebilir hale getirilmiştir. Nesne-ilişkisel veritabanlarından uzamsal (mekânsal) veritabanları; nokta, çizgi ve poligon gibi kompleks geometri objeleri ile tanımlanabilen mekânsal veriler için hızlı depolama ve sorgu avantajlarına sahiptir. Bu şekilde uydu görüntülerinin zaman-mekânsal ve topolojik analizlerinde, hızla gelişen nesne-ilişkisel veritabanlarındaki gelişmelerden faydalanılmıştır. Bu yaklaşım, uygulamada servis kalitesi yanında daha iyi bir performans sağlamaktadır. Geliştirmiş olduğumuz sistemin etkinliği ve kullanılabilirliği; deprem, toprak kayması, dolgu gibi çeşitli senaryo durumları ile

test edilmiştir. Senaryolar, doğa olaylarının belirlenen ada üzerindeki etkisinin uydu görüntüleri aracılığıyla ölçülmesi temeline dayanmaktadır.

1.2. Tez Çalışmasının Katkıları

Tez çalışmalarının katkıları beş ana başlık altında ifade edilebilir:

- Ada nesnesinin renk uzay dönüşümü ile segmentasyonu ve bölge gruplama ve etiketlemeye dayalı yöntemlerle vektörel olarak modellenmesi yapılmıştır.
- Zaman-mekânsal ve topolojik sorguların ada üzerinde test edilmiştir.
- Doğal olayların (dolgu, aşınma ve deprem vb.) sistem üzerinde simülasyonun gerçekleştirilerek sistemin etkinliğinin ve doğruluğunun test edilmesi sağlanmıştır.
- Dağıtık sistemler üzerinde raster ve vektör formatındaki görüntünün sistem performansına etkisi araştırılmıştır.

1.3. Literatür Taraması

Vektörizasyon algoritmalarını kabaca altı sınıfa ayırabiliriz: Hough Dönüşüm tabanlı (Hough Transform based) metotlar, inceltme tabanlı (thinning based) metotlar, şekil tabanlı (contour based) metotlar, grafik yürütme tabanlı (run-graph based) metotlar, ağ desen tabanlı (mesh pattern based) metotlar ve ayırık piksel tabanlı (sparse pixel based) metotlar.

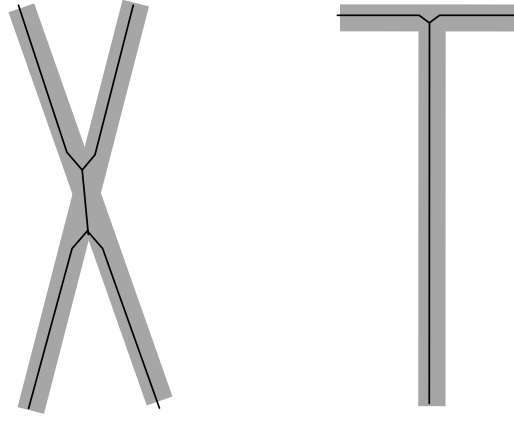
Hough Dönüşüm (HD) tabanlı metotlarda, ikili görüntüdeki (siyah-beyaz) uzaysal uzatılmış desenler daha kısa parametre uzayına dönüştürmek suretiyle vektörizasyon gerçekleştirilmektedir. Bu dönüşüm zor olan görüntü uzayındaki tespit problemini, parametre uzayında daha kolay çözülen yerel doruk tespit problemine dönüştürmektedir. Hough Dönüşüm tabanlı metotlar düz çizgileri, eğimi ve kesişim noktalarına göre parametrize ederek tespit ederler: (x,y) koordinat eksenindeki her çizgi, (m,c) ekseninde bir noktaya karşılık gelmektedir. (x,y) eksenindeki bir noktadan ise sonsuz sayıda doğru geçebilmektedir. Bu doğruların eğimleri ve

sonlandırılma noktaları (m,c) düzleminde bir çizgi oluşturmaktadır. Bu çizgi ise Eşitlik (1.1) ile tanımlanmaktadır.

$$c=-mx+y \quad (1.1)$$

m ve c değerleri orijinal görüntüde kırık ve gürültülü çizgilere ait olan noktalar için (m,c) düzleminde doruklar oluşması beklendiğinden, Hough Dönüşüm metodu gürültülü görüntülerdeki çizgileri de tespit edebilmektedir. Bu algoritmanın en basit versiyonu çizgileri tanımasına rağmen, daha karmaşık şekillere (eğriler, çemberler ve elipsler gibi) uyarlanabilir. Vektörleştirme işlemine tabii tutulacak ada görüntüsünün sınırları yalnızca düz çizgilerden oluşmadığı veya düzgün bir çember, elips olmadığı için HD direkt olarak kullanılamaz. Yöntem, her bir piksel üzerinde en az bir kez işlem yaptığı için dikkate değer bir hesaplama zamanını gerektirir. Ayrıca çok fazla bellek gereksinimi, kısa çizgiler için düşük dorukların oluşumu ve kenar piksel bağlantısının muhafazasındaki yetersizlik gibi bir takım kısıtlamalara sahiptir [2]. Bu kısıtlamalar HD'nin sadece daha küçük boyutlu görüntüler üzerinde kullanılmasına olanak veriyordu. HD'nin farklı alanlarda uygulanabilirliğinin sağlanması için farklı versiyonları geliştirildi. Bu kapsamda literatürde Li ve ark. [3] hızlı HD, Illingworth ve Kittler [4] adaptif HD, Ben-Tzvi ve Sandler [5] tümleşik HD, Princen ve ark. [6] hiyerarşik HD ve Atiquzzaman [7] çoklu çözünürlüklü HD'yi önerdiler.

Literatürde kullanılan yöntemlerden biri olan iskeletleştirme (çekirdek çizgi tespiti, orta eksen dönüşümü veya simetrik eksen dönüşümü) algoritmaları; yinelemeli sınır erozyonu (iterative boundary erosion), uzaklık dönüşümü (distance transform) ve uygun iskelet (adequate skeleton) olmak üzere üç ana grupta toplanabilir. Yinelemeli sınır erozyonu, orta eksen ya da iskelet kalana kadar, sınır piksellerini silme düşüncesine dayanmaktadır [8,9]. Bu tür algoritmalar, Şekil 1.1'de görüldüğü gibi, X ve T gibi kesişimleri bozabilmektedir. Yinelemeli sınır erozyonu için geliştirilen daha iyi yöntemler, görüntü üzerinde gezdirilen pencere boyutunu değiştirmeyi veya işaretleme kurallarını iyileştirmeyi içermektedir. Örneğin Deutsch [10]kare olmayan pencereler kullanırken, O'Gorman [11] k x k boyutunda pencereler kullanmaktadır. Fakat bu yöntemde yapılan düzenlemeler, doğruluk ve hız açısından çok az ilerleme sağlamışlardır.



Şekil 1.1. Kesişimlerdeki inceltme bozulmaları

Bir diğer yaklaşım uzaklık dönüşümüdür. Rosenfeld [12] ve Pfaltz [13], resimdeki her pikselin yerine, piksele en yakın beyaz pikselin uzaklığını gösterecek şekilde, bir sayı atamak suretiyle uzaklık dönüşümünü gerçekleştirmişlerdir. Bu algoritmanın en kötü yanıiskeletin, özellikle birleşim yerlerinde bağlantısının kopmasıdır.

Yukarıdaki iki farklı algoritmayı birleştiren bir üçüncü algoritma ise Davies ve Plummer[14]tarafındanuygun iskelet diye adlandırılan, birleşik inceltme algoritmasıdır. Genel olarak, inceltme algoritmalarının amacı, veri boyutunu azaltarak, sadece resimdeki şekilleri kullanmayı sağlamaktır. Birçok inceltme algoritması; yüksek zaman karmaşıklığı, çizgi kalınlığı gibi şekil bilgilerinin kaybolması, kesişimlerde bozulma, yanlış ve sürpriz dallanmalar gibi dezavantajları vardır. Bu algoritmalar, daha çok çizgisel görüntülerin vektörizasyonunda kullanılır. Temel uygulama alanları görüntü alanının küçük ve çizgi kalınlığının önemli olmadığı, optik karakter tanıma uygulamalarıdır. Performans bakımından çok iyi olan algoritmalar da mevcuttur [14-18].

Şekil tabanlı metotlarda, ilk önce raster görüntülerden şekiller çıkarılmakta ve daha sonra çizgi gibi alanları tanımlamak için eşlenebilir şekiller belirlenmektedir. Çoğunlukla nokta zincirleri ile temsil edilen orta eksenler, bu şekil çiftleri arasından oluşturulmaktadır [19]. Vektörizasyon algoritmasının, her türlü şekle karşı uygulanabilir olması, kesişimlerde yanlış karar vermemesi ve yanlış iskelet üretmemesi gerekliliği dikkate alındığında, çok kesişimin ve eğik çizgilerin olduğu çizimlerde kullanılması uygun değildir.

Grafik yürütme tabanlı metotlar, yürütme uzunluğunu hesaplamak için raster görüntüleri ya satır ya da sütun boyunca taramaktadır[20]. Daha sonra yürütmeler, grafik yapıları oluşturmak için analiz edilmektedir. Bu metot, gürültüye duyarlıdır ve kesişimlerde bozulmalara neden olabilmektedir. Ağ desen tabanlı metotların temel fikrini, Lin ve ark. [21] tüm görüntüyü belirli bir ağa bölerek ve ağın sınırları içerisindeki siyah piksellerin dağılımına bakarak karakteristik desenlerini belirlemişlerdir. Bu desenler kullanılarak görüntüye ait bir kontrol haritası oluşturulmakta ve bu haritalar yardımıyla da uzun, düz çizgilerin çıkarılması gerçekleştirilmektedir. Elde edilen ağın özellikleri, veritabanındaki ile karşılaştırılmakta ve eşdeğeri ile etiketlenmektedir. Daha sonra görüntüdeki her ağ biriminin yerine gerçek görüntüde etiketi kullanılarak kontrol haritası oluşturulur. Eğer bilinmeyen, karmaşık desenler var ise soru işareti ile etiketlenirler. Ağ çok büyük olursa daha fazla işlem zamanı harcanmaktadır. Küçük olursa erişim oranı artarak işlemi zorlaştırmaktadır. Ağ boyu, çizimdeki en büyük çizgi kalınlığından daha büyük iki çizgi arasındaki mesafeden de daha küçük olmalıdır. Çizgi tespiti gerçekleşebilmesi için, ağ boyu çizimdeki en küçük çizgi boyundan da küçük olmalıdır.

Ayrık piksel tabanlı metotlar, Dori tarafından geliştirilen dikey zig-zag (OZZ) metodundan esinlenerek yine Dori tarafından gerçekleştirilmiştir [22]. OZZ algoritmasının temel fikri, tek piksel kalınlığında bir ışık ışınının çizgileri takip ederek herhangi bir kenara çarpması durumunda dikey olarak dönmesi prensibi ile çalışır.

Literatürde vektörizasyonla ilgili var olan diğer çalışmalar da şunlardır: Zhang ve ark. [23] matematiksel morfolojiye dayalı bir yöntem sunmuşlardır. Bu yöntemde önce görüntüden gürültü ve kenarları ayırmak için dört yapısal element (sağ, sol ve diyagonaller) şablonuna dayalı kenar saptandıktan sonra sekiz yapısal elementle ana ve ara yönlerdeki pikseller kaldırılarak görüntü tek piksel kalınlığına indirgenir. Son olarak Freeman zincir kodlaması kullanılarak vektörleştirme yapılır. Mohamed ve ark. [24] vektörizasyonu kısıtlamalı Delaunay üçgenleştirme ile sağladıktan sonra elde edilen üçgenler vektör görüntü oluşturarak poligonlar şeklinde gruplanır.

Tam ve Heidrich [25] temel bileşen analizi yardımıyla kenar modelleme tekniğini sunmuşlardır. Bera [26] Canny kenar saptama yaptıktan sonra orijinal görüntüden rastgele renk çıkarımı kullanılarak her bir tayin edilen segmente gölge ataması yapar ve renk dalgacıklarına SVGşema ile haritalar. Lee ve ark. [27] uygun parametre seçimine izin veren bilgi tabanlı otomatik vektörizasyon metodu önermişlerdir. Kirsanov ve ark. [28] uydu görüntüsündeki nesnelere bulmak için, önce görüntünün gradyan geçişlerindeki kenarları saptayan eşikleme filtresi kullanılmıştır. Sonra elde ettikleri veriyi vektöre çevirmek için birleşik bileşen analizinden yararlanmışlardır.

Dunfrey ve ark. [29] mekânsal verileri modellemede SVG kullanımı önerisi, ya da Ballagh ve ark. [30]'nın Google'ın geliştirdiği daha çok küçük boyutlu verilerin harita üzerinde bindirilmesi için kullanılan KML standart veri modelini bilimsel verilerin temsili için uyarlama çalışması vektör veri modellemeye örnek olarak verilebilir. Ancak çalışmalarındaki gerek SVG gerekse KML yaklaşımının amacı, mekânsal ve mekânsal olmayan öz niteliklerinin bir arada tutulmasını XML tabanlı bir yapıda formülize etmektir. Bunu yapabilmeleri için gerekli olan lokasyon bilgileri nokta setleri olarak kendilerine verilmektedir.

Henüz uydu görüntüsünden ada çıkarımı ve vektörleştirme çalışması yapılmamasına rağmen, araştırmacılar uydu görüntülerinden nesne (yol, nehir ve kıyı şeridi gibi) çıkarmak için alternatif çözümler sunmuşlardır. Ahmed ve ark. [31] uydu görüntüsünün kenar segmentasyonu için 2-boyutlu faz uygunluğu modelininin yeni bir uygulamasını sunmuşlardır. Yaklaşımları; giriş resmini yumuşatmak için içeriksel doğrusal olmayan yumuşatma algoritması ve 2-boyutlu gerilmiş Gabor filtresi içermektedir. Önerilen yöntemin yetkinliği bilinen diğer metotlarla kıyaslanmıştır. Marina ve ark. [32] giriş resmi üzerinde minimum uzaklık segmentasyonu uygulayan daha sonra eşleşen karelerin süper çözünürlük iyileştirmesi uygulayarak kenar izleme işlemi yapan bir yöntem önermişlerdir. Geleneksel kenar ölçüm metotlarından daha iyi sonuçlar verdiği saptanmıştır. Jubai ve ark. [33] deniz/okyanus üzerine dökülmüş kirlilikleri saptamak için bulanık mantık teorisine dayalı geliştirilmiş Pal-King algoritması ve genetik algoritmayı birleştirmişlerdir. Shah ve ark. [34] sulu kısımları çıkarmak için kenar saptama, morfolojik erozyon ve özellik çıkarma gibi

algoritmaları kullanan; çıkarılan kısımlardaki sulu yerleri sınıflandırmak (normal, kurak, sel gibi) için K-ortalama kümeleme, Hill Climbing, renk histogramı ve görüntü eşikleme gibi algoritmaları kullanan bir yöntem önermişlerdir. Jodouin ve ark. [35] topolojik veritabanından alınan çok bantlı görüntüler üzerinde alan ve kontur saptamak için ful otomatik bir model sunmuşlardır. Veritabanında yer alan vektörleri güncellemek için istatistiksel ölçümleri ve gradyan tabanlı kontur algoritmalarını entegre etmişlerdir. Mangala ve Bhirud [36] uydu görüntüsünden yolları çıkarmak için ANN-tabanlı bir yöntem önermişlerdir. Bu çalışmada segmentasyon adımında görüntüler bölge ölçeklenebilir uydurma modeli kullanılarak segmente edilmiştir. Bir sonraki adımda baskın nesnelere iki aşamalı morfolojik işlemler yardımıyla çıkarılmış. Son aşamada ise iyi eğitilmiş bir sinir ağı yardımıyla tatmin edici sonuçlar alınmıştır.

1.4. Tezin Yapısı

Bu tez çalışması, 5 bölümden oluşmaktadır. Bölüm 1, giriş bölümü olup burada çalışmanın konusu, bu teze başlanmasındaki amaçlara açıklanmış ve konu ile ilgili geniş literatür taraması yapılmıştır. Bölüm 2'de görüntü işleme hakkında kısa bilgi verilmiş ve coğrafi veri kaynaklarından uydu görüntülerini kullanım gerekçelerinden bahsedilmiştir. Ayrıca raster ve vektör veri yapılarının birbirlerine göre üstünlükleri, mekânsal veritabanları ve uzaktan algılama hakkında kısa bilgiler verilmiştir. Bölüm 2'de ayrıca gürültü giderme ve kenar iyileştirme ile ilgili bilgiler verilmiştir. Bölüm 3'te Landsat-7 uydu görüntülerinin poligon nesnesi (vektörel) olarak modellenerek bilgisayarların ve bilgi sistemlerinin kolayca anlayabileceği ve işleyebileceği hale getirilmesi temeline dayalı mimari hakkında bilgi verilmiştir. Bölüm 4'te çalışma için oluşturulan zaman-mekânsal değişimlerin irdelenmesi ve raster-vektör veri transfer zamanı karşılaştırılması uygulamaları ele alınmıştır. Bölüm 5'te yöntemimiz ile alakalı sonuçlar verilmiş, bu sonuçlar değerlendirilerek ileriye dönük yapılabilecek çalışmalar önerilmiştir.

2. UZAKTAN ALGILAMA GÖRÜNTÜ İŞLEME

2.1. Görüntü İşleme

Günümüz teknolojisinde uzaktan algılama verisi sayısal olarak kaydedilmekte ve gerçekte tam görüntü yorumlama ve analizi bazı sayısal işlem elemanlarını içermektedir. Görüntü işleme, verinin bilgisayar prosedürleriyle düzeltilmesi, daha iyi görüntü sağlamak için sayısal zenginleştirme veya coğrafi özelliklerin otomatik olarak sınıflandırılmasıdır. Sayısal görüntü işleme için geniş bellek kapasitesine sahip hızlı bir bilgisayar ve veri işleme için uygun bir yazılımdır. Bunun içingünümüzde, ticari amaçlarla üretilen ERDAS, ERMapper, ENVI, Idrisi ve GRASS gibi birçok CBS ve görüntü işleme yazılımı gerçekleştirilmiştir. Bunlara örnek olarak bulunmaktadır.

Görüntü analiz sistemlerinde bulunan temel görüntü işleme fonksiyonları aşağıdaki şekilde sınıflandırılabilir:

- Ön işlem (preprocessing)
- Görüntü iyileştirme (image enhancement)
- Görüntü dönüşümleri (image transformation)
- Görüntü sınıflandırma ve analizi (image classification and analysis)

Sayısal görüntü analizinde ön işlem başlangıç ve temel veri analizidir. Bu aşamada verilere genellikle radyometrik ve geometrik düzeltmeler getirilmektedir. Bu işlemlere aynı zamanda görüntü restorasyonu ve doğrulama ya da rektifikasyon adı verilir. Amaç, algılayıcı ve platformdan kaynaklanan radyometrik ve geometrik bozulmaları ortadan kaldırmaktır. Radyometrik düzeltmeler; algılayıcı düzensizlikleri ve istenmeyen algı ve atmosferik gürültülerin düzeltilmesini ve sensör tarafından algılanan, elektromanyetik ışınımın en az hata ile ölçülmesini içerir.

Geometrik hatalar, algılayıcı optiğinin perspektifliđi, tarama sistemlerinin hareketi, platform hareketi ve yüksekliđi, arazi eğimi ve dünyanın eğim ve ekseni gibi birçok faktöre bađlıdır. Geometrik düzeltmelerin amacı görüntüyü geometrik olarak gerçek dünyaya en yakın duruma getirmektir.

Görüntü zenginleştirme, görsel yorumlama ve görüntünün daha iyi anlaşılabilmesi için yapılır. Görüntü zenginleştirme fonksiyonları ekran üzerindeki çeşitli detaylar arasındaki ton farklılıklarını daha da belirginleştirir ve bir görüntüdeki konumsal detayları artırarak veya eksilterek örüntü tanıma işlemlerine yardımcı olur.

Görüntü zenginleştirme tek bir bant üzerinden yapılırken görüntü dönüşümü birden fazla çok bantlı görüntü kullanılarak yapılır. Temel aritmetik işlemlerle mevcut orijinal bantlar üzerinde ilave, çıkarma, çoğaltma, parçalama yapılarak coğrafik detayların ekranda daha kaliteli bir görüntü oluşması sağlanır. Spektral oranlama (spectral rationing) görüntülerin dönüşümünde en çok kullanılan yöntemdir. Özellikle yüzey yansımalarının farklıolduđu durumlarda iki farklı banttaki spektral yansıma deđerleri kullanılarak elde edilen oran bantları önemli bilgiler sağlamaktadır. Çok kanallı görüntülerde kullanılan diđer bir yöntem temel bileşen analizi (principal component analysis) dönüşümüdür. Bu yöntemle veri boyutlarını (örneğin bant sayısı) azaltarak orijinal görüntüdeki bant sayısına eşit korelasyonsuz yeni veri seti elde edilmektedir.

Sınıflandırma ve analiz işlemleri benzer özellikteki pikselleri sayısal olarak tanımlamak ve bunları kümelemek olarak tanımlanabilir. Sayısal olarak görüntülerin sınıflandırılmasında bir veya birden çok spektral banttaki spektral piksel deđerleri kullanılır. Böylece görüntüden yeryüzüne ait mekânsal ve spektral detayları (mısır, arpa, buğday gibi) ifade eden bir tematik harita elde edilir. Sınıflandırma, aynı özelliđe sahip mekânsal detayların (benzer ağaç türleri gibi) gruplaşmasını yapan bilgi sınıflandırılması (information classification) ve piksellerin parlaklık deđerlerini kullanan spektral sınıflandırma (spectral classification) diye ikiye ayrılabilir. Sınıflandırma çeşitli yollarla gerçekleştirilebilir. Bunlardan en yaygın olanı denetimli (supervised) ve denetimsiz (unsupervised) sınıflandırmalardır. Denetimli sınıflandırmada görüntü yorumlayıcı tecrübesine dayalı olarak öncelikle görüntü

üzerinde farklı özellik gösteren alanlar belirlenir. Daha sonra bu alanlara isabet eden piksel özellikleri diğer bölgeler ile karşılaştırılarak görüntünün tamamındaki pikseller sınıflandırılır. Denetimsiz sınıflandırma da ise denetimlinin tersine öncelikle piksellerin değerine dayalı olarak spektral sınıflar oluşturulur. Daha sonra kümeleme algoritmalarıyla verinin istatistiksel olarak sınıflanması yapılır.

2.2. Gürültü Giderme ve Kenar İyileştirme

Genel olarak görüntü iyileştirme yöntemleri sorunların çözümüne sezgisel (heuristic), yaklaşır. Görüntü iyileştirmede karşıtlığı arttırma, gürültü giderme, kenarlara vurgu yapma, kenar bulma gibi işlemleri yerine getirmek için, çeşitli şekillerde uygulanan filtrelerden (süzgeç) faydalanılır. Filtrelerle ilgili çözümler de, uzamsal ve frekans düzlemlerinde olmak üzere farklı yaklaşımlar vardır. Uzamsal düzlemde, görüntü pikselleriyle doğrudan işlem yapılır. Frekans düzleminde, Fourier dönüşümlerinin kullanımıyla doğrusal filtreler, kuramsal olarak tasarlanır. Bir filtrenin sayısal olarak elde edilmesinde çeşitli hesaplamalar kullanılsa da, tasarım ilkelerinin çıkarımı Fourier dönüşümünden elde edilir.

Görüntüden istenen özelliklerin elde edilmesinde, karşıtlık ve parlaklığın ayarlanması yeterli değildir. Görüntünün iyileştirilmesi için birçok farklı filtreleme yönteminin de kullanılması gerekir. Bu işlemlerde kullanılan algoritmalar uygulamada, doğrusal ve doğrusal olmayan yaklaşımla ifade edilir. Aradaki temel fark, doğrusal sistemlerin Fourier düzleminde analizinin mümkün olmasına karşılık, doğrusal olmayan sistemlerin analizinin mümkün olmamasıdır. Doğrusal olmayan filtreleme yöntemlerinin en büyük avantajı hızlı ve basitleştirilmiş hesaplamalardır. Genellikle istatistik temelli filtrelerdir ve toplanır türdeki gürültülerde iyi sonuç verirler [37].

Filtreleme, bir görüntünün iyileştirilmesi ya da değiştirilmesi amacıyla kullanılan komşuluk bölgesi işlemlerine dayanan yöntemlerdir. Uzamsal düzlemde uygulanan filtreler bir görüntüdeki istenen nesnelere belirginleştirilmesi ya da bastırılması amacıyla kullanılır. Düzleştirme (smoothing), alçak geçiren filtre; keskinleştirme (sharpening), yüksek geçiren filtre; gürültülü görüntü üzerinde kenar iyileştirme işlemleri de bant geçiren filtre işlevleridir. Örneğin, alçak geçiren

bir filtre, benzer tonda homojen alanları düzleştirerek, küçük detayları azaltıp homojenleştirmeye yarar. Yüksek geçiren filtrelerse, ince detayları (kenar) belirginleştirir ve mümkün olduğu kadar çok detayı ortaya çıkarır. Görüntü düzleştirme ve özellik korumalı filtrelerin birçoğu doğrusal olmayan özellikte olduğundan, bu tür filtreler genelde Fourier analizi ile sağlıklı sonuçlar vermez. Görüntü üzerine filtrelerin uygulanması; karışıklığı artırma, gürültü kaldırma, kenarlara vurgu yapma, bulma, şekli değiştirme gibi amaçlar içerir. Bir görüntü üzerinde doğrusal filtreleme konvolüsyon operatörüyle uygulanır.

Görüntülerdeki gürültülü yerleri yok etmek için kenar olan yerlerin muhafazasını sağlayan bir operatör gerekmektedir. Örneğin Gauss filtresi, kenar özelliklerini belirsiz hale getirmeden optik görüntüleri filtreleyebilir. Lee filtresi [38], kenar keskinliğini bozmadan radar gürültü ve beneklerini azaltabilir. Ortalama filtre, Ortanca filtre, Wiener filtre [39], Frost filtre [40] ve Kuan filtre [41] gibi süzgeçler görüntüdeki gürültüleri azaltmak için kullanılabilir.

Bir görüntünün, kenarlarının algılanabilirliği, küçük detayları, yüksek uzamsal frekans bileşenlerine ait genliklerinin artırılmasıyla geliştirilebilir. Bu iş için keskin olmayan (unsharp) maskeleme tekniği en iyi ve en etkin yöntemdir. Keskin olmayan maskeleme formunun gösterimi Eşitlik (2.1)'deki gibi;

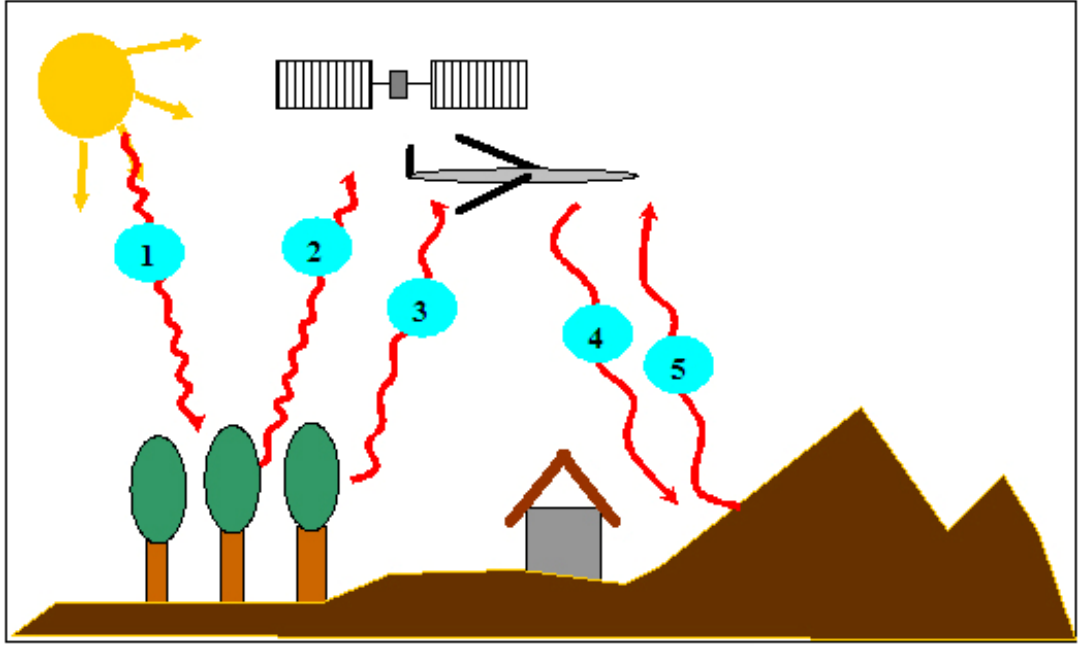
$$Y=X+\alpha(X-\bar{X}), \alpha>1 \quad (2.1)$$

ifade edilir. X orijinal görüntünün piksel değerleri, Y sonuçta elde edilen görüntü ve \bar{X} orijinal görüntüden elde edilen düzleştirilmiş versiyonudur. Fark görüntüsü ($X-\bar{X}$) versiyonudur. Fark görüntüsü görüntünün yüksek uzamsal frekans içeriğini gösterir ve iyileştirme faktörü α , bu bileşenlerin sonuçta elde edilen görüntüye ilave etme değerini saptar. Kenar iyileştirme için dinamik erim azaltma veya eşyönsüz difüzyon filtre [42] de kullanılabilir.

2.3. Uzaktan Algılama

Uzaktan algılama, yeryüzünün ve yer kaynaklarının incelenmesinde onlarla fiziksel bağlantı kurmadan kaydetme ve inceleme tekniğidir [43]. Uzaktan algılama; uzaktan algılama sistemleri ile elde edilen verilerden yararlı bilgiler elde etmek amacıyla yapılan bütün kayıt, işleme, analiz, tanımlama, yorumlama ve sonuç olarak bilgi üretme çalışmalarının tamamını kapsamaktadır. Bu çalışmaların tamamının yapılabilmesi enerjiye bağlıdır. Nasıl ki doğadaki tüm olaylar bir enerji vasıtasıyla gerçekleşiyorsa uzaktan algılamada bu şekilde enerji vasıtasıyla gerçekleşir. Uzaktan algılamanın daha iyi anlaşılabilmesi için kendi beş duyumuzun algı özelliklerini kısaca olsa da bilmekte yarar vardır. Gözlerimiz görüş alanına düşen tüm cisimleri alır ve değerlendirip tanımlaması için beyine gönderir, beyinde bunları değerlendirip algının tamamlanmasını sağlar. Aynı şekilde kulak da etraftan yayılan ses dalgalarını toplayarak anlamlandırması için beyne gönderir ve beyinde tanımlama işlevini tamamlayarak duyma dediğimiz olayın gerçekleşmesini sağlar. Tüm bu olayların gerçekleşmesi içinde ATP(Adenozin trifosfat) enerjisi kullanılır. Uzaktan algılama için kullanılan enerji kaynağında ya güneştir ya da yapay bir güç kaynağıdır.

Uzaktan algılamanın temelini oluşturan esas olay algılamadır. Algılayıcılar kullandıkları enerji kaynaklarına göre aktif ve pasif algılayıcılar olmak üzere ikiye ayrılırlar. Uzaktan algılamada bilgi araştırılan objeden (yeryüzü veya atmosfer) algılayıcıya elektro manyetik enerji yayılımı ile taşınır. Şekil 2.1 uzaktan algılama sistemlerinin temellerini göstermektedir [44].



Şekil 2.1. Uzaktan algılama sistemlerinin temelleri

Doğal bir kaynaktan (güneş) elektromanyetik enerji yayılır (1) ve yeryüzü tarafından yansıtılan elektromanyetik enerji yeryüzüne ait bilgiyi bir uzaktan algılama platformuna yerleştirilmiş algılayıcıya (uydu veya uçak) taşır (2). Ya da yer yüzeyi ve objeler elektromanyetik enerji yayarlar (3), böylelikle yeryüzüne ait bilgiler algılayıcıya taşınır. Aktif algılayıcı taşıyan bir uçak veya uydu, yeryüzüne elektromanyetik ışınlar gönderir (4), Bu ışınlaryeryüzü tarafından yansıtılır ve bilgiler algılayıcıya ulaşır (5). (1) ve (2)'deki temellere sahip algılayıcı sistemlere pasif sistemler, (3)'deki sisteme sahip algılayıcılara ise aktif sistemler adı verilir. Pasif algılayıcıların önemli iki özeliği vardır. İlk olarak yalnızca güneş varlığında algılama yaparlar, hava bulutlu ise algılama yapamazlar. İkinci önemli özelliği ise yeryüzünden yansıyan elektromanyetik enerjiyi kullanmaları ve bu sayede enerji tasarrufu sağlamalarıdır. Aktif algılamada korkunç bir enerjiye ihtiyaç duyulur. Aktif algılamada kullanılan cihazların ömürleri de kendilerine depolanan enerji maliyetindedir. Bu tür sistemler gece gündüz demeden sürekli görüntü alma kabiliyetinesahiptirler.

Bir uydu görüntüsünü oluşturan dört temel bileşen; yer, zaman, yansıma ve radyometrik özelliklerdir. Uzaktan algılamada bu temel bileşenlerden maksimum düzeyde yararlanmak, bilim adamlarının ve uzay araştırmalarının temel konusu

olmuştur. Bu bileşenler bir görüntüde çözünürlük olarak ifade edilmektedir. Çözünürlük, genel olarak ekranda görünen piksel sayısını veya görüntüdeki bir pikselin yeryüzündeki karşılığını ifade etmek için kullanılır. Uzaktan algılamada; spektral (spectral), zamansal (temporal), mekânsal(spatial) ve radyometrik (radiometric) olmak üzere dört farklı çözünürlük tipinden bahsedilebilir. Günümüzde yeni fırlatılan ya da planlanan uydularla, mekânsal(IKONOS, QuickBird vb.) ve radyometrik (MERIS, MODIS vb.) çözünürlüğü yüksek veri temini mümkün hale gelmiştir. Bunun yanı sıra uydularla zamansal çözünürlük artırılarak yeryüzündeki aynı noktayı tekrar gözlemleme süreleri birkaç güne kadar düşürülmüştür.

Spektral çözünürlük, bir algılayıcının elektromanyetik spektrumdaki ayırt edebildiği dalga boyu genişliklerini göstermektedir. Her bir spektral bant belirli bir dalga genişliğinde algılanmış görüntüden oluşur. Çoğu uzaktan algılama sistemleri enerjiyi birçok dalga boyu genişliğinde ve çeşitli spektral çözünürlükte gözlem yapar. Bu tür sistemler, çok bantlı algılayıcılar olarak adlandırılır. Çokbantlı uydu sistemlerinin daha gelişmiş teknolojisi ise hiperspektral (çoklu-ışınsal) algılayıcılardır. Hiperspektral sensörler elektromanyetik spektrumun görünür, yakın kızıl ötesi ve orta kızıl ötesi bölgelerinde yüzlerce spektral bant genişliğinde ayrı ayrı algılama yapabilirler.

Mekânsal çözünürlük, yeryüzünde görüntülenen en küçük birim alanın büyüklüğüdür. Pasif bir algılayıcının konumsal çözünürlüğü, IFOV (Instantaneous Field of View)'a bağlıdır. IFOV, bir detektör tarafından anlık görüntüleme açısidir. Görüntülenen alanın büyüklüğü IFOV ile algılayıcının yere olan uzaklığına bağlıdır. Uzaktan algılama uygulamalarında en fazla ilgi çeken veriler, günümüzde kullanımı yaygınlaşan ve ağırlıklı olarak ABD tarafından geliştirilen yüksek mekânsal çözünürlüğüne sahip uydu verileri olmuştur. Yüksek mekânsal çözünürlüğü, görüntüdeki piksel boyutlarının gerçek yer düzleminde 10x10 m veya daha küçük büyüklükte ifade edilmektedir [45].

Radyometrik çözünürlük, görüntünün EMR (Elektromanyetik radyasyon) şiddetine karşı hassasiyeti anlamına gelmektedir. Bir görüntüleme sisteminin radyometrik çözünürlüğü, enerjideki çok küçük değişimleri belirleyebilme kabiliyetidir ve

algılayıcının sinyal/bozulma oranına bağlıdır. Görüntülerde mümkün olan maksimum parlaklık sayısı, kaydedilen enerjiyi temsil eden bit sayısı ile ifade edilir. Örneğin 8 bitlik bir veride 0-255 aralığında 256 sayısal değer vardır. Bit sayısı azaldıkça radyometrik çözünürlük de azalır. Genellikle en düşük sinyal seviyesine 0, en yüksek sinyal seviyesine 255 değeri verilir [46]. Zamansal çözünürlük, belirli bir bölge için bir algılayıcının görüntüyü hangi sıklıkta elde ettiğini gösterir.

CBS açısından en etkin uydular yeryüzünün doğal ve yapay detayları üzerindendeğişik şekillerde bilgi toplayan uydulardır. İlk uzaktan algılama uydusu LANDSAT1972 yılındaABD tarafından uzaya gönderilmiştir. Ardından Fransız, Belçika ve İsveç ortak yapımı SPOT uydusu, 1986 yılında fırlatılmıştır. Daha sonra 1990-95 yılları arasında beş yeni uydu uzayagönderilmiştir. Bunlar; Avrupa ortak yapımı ERS, ilk Kanada uydusu RADARSAT, Rus yapımı RESURS, Hindistan yapımı IRS ve Japon uydusu JERS'dir. Başlıca uzaktan algılama uyduları ve özellikleri aşağıda belirtilmiştir.

LANDSAT: İlk fırlatıldığında ERTS (Dünya Kaynakları Teknoloji Uydusu) olarak isimlendirildi. Bu uyduyu takiben bu seri içinde iki uydu daha 1975 ve 1978 tarihlerinde yörüngeye oturtuldu. Bu uydular iki ayrı set algılayıcı taşımaktaydı: Yüksek çözünürlüklü görüntü elde edebilen geliştirilmiş vidikon televizyon kamerası (RBV) ve çok bantlı spektral tarayıcıdır (MSS). RBV üç banda (0.475-0.575mm; 0.58-0.68mm ve 0.69-0.83mm) sahipken; MSS 0.5-0.6 mm, 0.6-0.7mm, 0.7-0.8 mm, 0.8-0.11 mm dalga boylarını kapsayan dört banda sahipti. Landsat 1, 2 ve 3 dünya etrafındaki tam bir turu 103 dakikada almaktaydı. Tekrar devri ise 18 gündü (18 günde bir aynı yeri algılar). Tarama uzunluğu 185 km ve tam görüntü büyüklüğü ise 185x185 km²'dir. Landsat uydularının ikinci serisi 1982 ve 1984 yılında fırlatıldı. Önceki uydularla kıyaslandığında Landsat 4 ve 5 'in RBV kamerası taşımadığı fakat bunun yerine daha gelişmiş algılayıcı seti olan geliştirilmiş geometrik çözünürlüklü 30x30 m² ve termal sıcaklığı ölçen bir bant da dâhil olmak üzere 7 bantlı spektral çözünürlüklü tematik haritalayıcıya (TM) sahiptir. Öncekilerde 18 gün olan zamansal çözünürlük bunlarda 16 gündür. Çok bantlı spektral algılayıcı bu uydularda da bulundurulmuştur. 1993 yılında, LANDSAT-6 şansız bir şekilde düştükten sonra LANDSAT-7 geliştirilmiş ve Mart 1999 da fırlatılmıştır. LANDSAT 7

geliştirilmiş ETM+ adı verilen yeni spektral algılayıcı taşımaktadır. Standart 7 Band'a ek olarak 15 m çözünürlüğe sahip pankromatik (siyah-beyaz) bant eklenmiştir. Üzerinde bulunan kayıt ünitesi sayesinde alıcı istasyonun olmadığı bölgelerde de görüntü çekebilmektedir. LANDSAT TM'nin bantlara göre kullanım alanları Tablo 2.1'de verilmiştir.

Tablo 2.1. Landsat TM bant özellikleri ve kullanım alanları

Bant	Kullanım Alanları
1	Bitki ve toprak arasındaki farklılıkların, ormanlık alanların ve kıyı çizgisinin haritalanması
2	Canlı bitkilerin yeşil bölümleri
3	Farklı bitki türlerinin tespiti, litoloji ve toprak arasındaki sınırın saptanmasında
4	Bitkilerin miktarını saptamada, litolojilerin tanımlanmasında, toprak/litoloji ve kara/su arasındaki zıtlığı gösterir
5	Kurak alanlar, su miktarı, kar ve buz arasındaki farkın bulunmasında
6	Sıcaklık miktarı, bitkiler, termal kirliliğin ve jeotermal alanların belirlenmesinde
7	Litoloji ve toprak arasındaki sınırın belirlenmesinde, toprak ve bitkilerdeki su miktarının saptanmasında

SPOT: SPOT uydusu Fransız Uzay Merkezi (CNES) tarafından planlanarak Fransa, Belçika ve İsveç tarafından üretildi. İlk kez 22 Şubat 1986'da işlev kazandı. İki farklı modda çalışan yüksek çözünürlüklü iki görüntüleme cihazına (HRV) sahiptir. Çok bantlı spektral modu (MSm) 20x20 m² geometrik çözünürlüklü tayfin yeşil (0.50-0.59 mm), kırmızı (0.61-0.68 mm) ve yakın kızılötesi (0.79-0.89 mm) bölümünü kaplayan 3 banttan oluşur. Pankromatik modu sadece görünür bölgeyi örten tek bant (0.51-0.73 mm) ve 10x10 m² yüksek geometrik çözünürlüğe sahiptir. Landsat'la kıyaslandığında SPOT uydusunun geliştirilmiş daha yüksek geometrik çözünürlüğü olduğu kadar, düşey doğrultusunun her iki yanından 27 dereceye kadar dik olmayan görüntüleme olanağına da sahip olması bazı gelişmiş özellikleridir. Uydunun tekrar döngüsü 26 gündür.

IKONOS: 24 Eylül 1999 yılında, Kaliforniya Vanderberg Hava Üssü'nden Space Imaging şirketi tarafından uzaya fırlatılmıştır. IKONOS, yeryüzüne ticari görüntü gönderen uydular içerisinde QuickBird uydusundan sonra en hassas olanıdır. Yüksek

detaylara sahip, çok bantlı, renkli (spektral), siyah-beyaz (pankromatik), üç boyutlu görüntüler; harita bazlı bilgilerin birleştirilmesi için idealdir. Uydu görüntülerinin yer kontrol noktalarıyla birlikte kullanımı, kesin yer tayini ve haritalama çalışmaları için kusursuz bir alan oluşturur. Uydu tamamen hareketli olup, görüntü alımında esneklik sağlar. Bu uydu ± 85 derece paralelleri arasında algılama yapabilir. IKONOS görüntüleri, CBS ve haritalama formatları ile uyumludur. IKONOS sensörü, müşteri tarafından belirtilen koordinatlar doğrultusunda görüntü alma imkânı sağlayan hareketli bir uzay aracı ile ticari uydu görüntüleme sisteminde atılan önemli bir adımdır.

IRS-1C/D: Hint yer gözlem uydusu (IRS-1C) 28 Aralık 1995'te yörüngeye oturtulmuştur. Ocak 1996'da ilk görüntüler alınmaya başlanmıştır. Serinin ikinci uydusu olan IRS-1D ise 29 Aralık 1997'de yörüngeye başarılı bir şekilde oturtulmuş olup 1997'nin Ekim'inde görüntü vermeye başlamıştır. Yörünge yüksekliği 817 km'dir. Günde 14 dönüşüm yapmaktadır. Dünya etrafındaki bir dönümü 101.35 dakika sürmektedir. Yeryüzü üzerinde aynı noktadan 24 günde bir geçer. IRS-1C ve 1D'nin birbirinden biraz farklı yörüngeleri vardır. Uydu görüntüsünün kaplama alanı $23 \times 23 \text{ km}^2$ 'dir.

QUICKBIRD: 2001 tarihinde ABD'nin Kaliforniya eyaletinden fırlatılmıştır. QuickBird 2, Digital Globe isimli özel bir A.B.D. şirketi tarafından çalıştırılmaktadır. 0.61-0.73 m pankromatik, 2.5-2.9 m multispektral konumsal çözünürlüğe sahip görüntü vermektedir. Standart proses ürünler için, pankromatik 0.70 m çözünürlük ve multispektral bantlar ise 3.0 m çözünürlük sunmaktadırlar. Başlıca kullanım alanları şunlardır: arazi örtüsü ve arazi kullanımı analizleri, sulama alanlarının tespiti, fiziksel yapı, ekim yoğunluğu, ürün sağlığı ve hastalık alanlarının belirlenmesi ve izlenmesi, tarla sınırları ve dönüm miktarı tespiti, çeşitli toprak türlerinin belirlenmesi, gelecekteki ürün performansı, ürün gelişimi ve ürün tahminine ilişkin modellerin geliştirilmesi, tarımsal veya tarım dışı arazi kullanımına ait haritaların çizilmesi, kentsel planlama, altyapı, telekomünikasyon, emlak, sigortacılık, acil yardım, milli savunma. QuickBird 2 bir görüntüyü ($17 \times 17 \text{ km}^2$) kabaca 4 saniyelik bir sürede çekmektedir.

ORBVIEW:Yüksek çözünürlüklüticari bir uydudur. Bir metrelik yer çözünürlüğü ile evlerin ve arabaların kolaylıkla ayırt edilmesini sağlamaktadır. 2003 yılında yörüngesine oturtulmuştur. Çerçeve büyüklüğü 8 km'dir. Stereo çekim özelliği de mevcuttur. 1m çözünürlüklü pankromatik, 4 m çözünürlüklü multispektral görüntüleme özelliklerine ilaveten belkide dünyanın ilk çoklu-ışınsal ticari uydusudur.

BİLSAT: İlk Türk yer gözlem uydusu BİLSAT 27 Eylül 2003 tarihinde Cosmos-3 roketiyle fırlatıldı. Uydunun yörüngeye yerleştirme işlemi başarıyla gerçekleştirildi. Uydu'nun üretimi Surrey Satellite Technology Limited (SSTL) Şirketi'nde gerçekleştirildi. Uydunun üretimine, Türkiye'nin teknoloji transferi sağlamak amacıyla görevlendirdiği 12 mühendis ve 4 teknisyen de katıldı. 686 km irtifadaki bir yörüngeye yerleştirilen uydunun temel görevi uzaktan algılama olmakla birlikte, bir ölçüde haberleşme yeteneğine de sahiptir. Günde iki defa Türkiye üzerinden geçecek olan, 129 kg ağırlığındaki BİLSAT Uydusu, uluslararası bir girişim olan Afet İzleme Takımuydu Sistemi'nde (Disaster Monitoring Constellation) de yer alıyor. Takımuydu Sistemi İngiltere, Cezayir, Nijerya, Çin ve Tayland'ın uydularından oluşacak ve dünyanın herhangi bir yerini en az günde bir kere görüntüleme yeteneğine sahip olacak. Böylece dünyanın herhangi bir yerinde meydana gelen bir afete ait uydu görüntüleri en geç bir gün içinde temin edilerek ilgili ülkenin afet yönetim merkezine ulaştırılacak. Teknoloji transferi ile gerçekleşen BİLSAT uydusunun iki önemli parçası TÜBİTAK BİLTEN (Bilgi Teknolojileri ve Elektronik Araştırma Enstitüsü) mühendislerince tasarlandı ve aynı kurumun tesislerinde üretildi. TÜBİTAK BİLTEN'de üretilen parçalardan birincisi ÇOBAN (Çok Bantlı Görüntüleyici) adı verilen, uzaydan 120 metre çözünürlükte görüntü verebilen 8 kanallı bir kamera. Diğeri ise, JPEG 2000 algoritmasıyla görüntü sıkıştırma ve GEZGİN (Gerçek Zamanlı Görüntü İşleyeci) adı verilen bir veri işleme kartı. ÇOBAN'dan alınacak görüntüler dışında, 12 m. çözünürlükte pankromatik ve 26 m. çözünürlükte multispektral (kırmızı, yeşil, mavi ve yakın kızıl ötesi bantlarda) iki görüntüleyiciden fotoğraf çekilerek TÜBİTAK BİLTEN'de kurulan yer istasyonu aracılığı ile yere indirilecek. Böylece BİLSAT'tan elde edilen görüntülerden ürün rekoltesi, çevre kirliliği, taabi afetlerin neden olduğu hasarın değerlendirilmesi gibi amaçlarla yararlanılabilecek. TÜBİTAK BİLTEN bu amaçla hem kamu kuruluşları

ile hem de uluslararası kuruluşlarla muhtelif projeler üretecektir. Tablo 2.2’de uydu sistemleri ve özellikleri özetlenmiştir.

Tablo 2.2. Uydu sistemleri ve özellikleri

UYDU	TARİH	ALGILAYICI TİPİ	ÇÖZÜNÜRLÜK			ŞERİT GENİŞLİĞİ (KM)
			Konumsal (metre)	Radyometrik (bit)	Zamansal (gün)	
LANDSA T-7	1999-	Pankromatik	15	8 bit	16	185
		ETM	30	8 bit	16	185
		ETM	60	8 bit	16	185
SPOT-5	2002-	HRS-PAN	10	8 bit	1-4(26)	120
		HRG-PAN	2.5-5	8 bit	1-4(26)	60
		HRG	10	8 bit	1-4(26)	60
		HRG	20	8 bit	1-4(26)	60
		Vegetation	1000	4/8 bit	1	2250
IKONOS	1999-	SAR	18	3 bit	44	75
		Pankromatik	1	11 bit	3.5-5	11
		Multispektral	4	11 bit	3.5-5	11
IRS-1C/D	1995-/ 1997-	Pankromatik	5.8	6 bit	24/25	70.5
		LISS-III	23	7 bit	24/25	141
		LISS-III	70	7 bit	24/25	141
		WIFS	188	7 bit	24/25	812
QUICKBI RD-2	2001-	Pankromatik	0.61-0.73	11 bit	3.5	16.5
		Multispektral	2.5-2.9	11 bit	3.5	16.5
ORBVIE W-3	2003-	Pankromatik	1	11 bit	8	8
		Multispektral	4	11 bit	8	8
BILSAT	2003-	Pankromatik	12	8 bit	5 (116)	25
		Multispektral	26	8 bit	4 (52)	55

2.4. Uydu Görüntüleri

Hava fotoğrafı, arazi çalışmaları ve kâğıt haritalar gibi birçok farklı coğrafik veri kaynağı varken, uydu görüntülerini kullanmanın yararları uydu görüntülerinin hızlı, daha ucuz ve daha iyi olmasıdır. Uydu görüntüleri genellikle coğrafi bilgi etmenin en pratik yoludur.

Hemen hemen bütün uydu görüntüleri sayısal (dijital) olarak elde edilmektedirler. Bu demektir ki pahalı sayısala dönüştürme aletlerine (scanner) gerek yoktur. Çok az bir ön hazırlıkla görüntü; coğrafi bilgi sistemine, görüntü işleme yazılımına veya harita işleme yazılımına yüklemeye ve kullanmaya hazır hale gelir ve sayısal olması sayesinde işlenen, zenginleştirilen ve yönlendirilen uydu görüntüsü diğer kaynaklardan alınan verilerde kaçırılacak ince ayrıntıları ortaya çıkarabilir.

Bir arazi ekibi ekipmanlarını hazırlarken veya bir pilot uçuş öncesi hazırlıklarını yaparken, bir uzaktan algılama uydusu geniş bir ormanı veya büyük bir şehri

görüntüleyebilir. Uyduların belli bir yörüngeye sahip olmaları nedeniyle, uydular genellikle bir projenin ilgili olduğu alandan en fazla bir hafta uzaklıktadırlar. Çok az planlama gerektirirler. Örneğin bugün verilen bir sipariş yarın, gelecek hafta veya uydunun programına bağlı olarak üç ay içerisinde görüntülenebilir. Geniş alanlar için uydu görüntüleri, hava fotoğraflarından veya arazi çalışmalarından daha ucuza mal olur. Genellikle, ham bir uydu görüntüsünün ortalama maliyeti kilometre başına bir doların altındadır.

Uydular politik veya coğrafik sınırlarla kısıtlanamazlar. Kutup yörüngesindeki ticari uydular dünyanın her bölgesini görüntülerler. Bir uzaktan algılama uydusu, inceleme yapılan alanın bir dağın tepesinde mi yoksa okyanusun ortasında mı yer aldığına bakmaksızın o bölgenin görüntüsünü elde edebilir. Günümüzün hızla değişen dünyasında, kritik iş kararlarını alabilmek için güncel verilere gereksinim duyulur. Haritalar genellikle aylar hatta yıllarca eskidir. Fakat bir uydu görüntüsü elde edildikten bir kaç gün sonra incelenebilir haldedir. Bir uydu görüntüsü elde edilebilen en yeni haritadır.

Uzaktan algılama uydusunun elde ettiği bir görüntüde; yeryüzü örtüsü, taşıt yolları veya belli başlı yapılar gibi yüzlerce veya binlerce km'lik alana yayılmış detaylar yakalanabilir. Kameranın yalan söylemediği gibi, uydu algılayıcıları da yalan söylemez. Ham uydu görüntüsünün elde edilmesi sürecinde hiç bir insani katılım yer almamaktadır. Elde edilen bilgiler, objelerin ve yapıların tarafsız gösterimi olup objektif ve doğrudur. Bir uydu görüntüsünde çalışırken, harita mühendisinin veya arazi ekibinin hata yapıp yapmadığı konusunda endişe edilmez. Uydu görüntülerinin işlenmesi veya bunlardan bilgi elde edilmesi, yapmak istenilen çalışmanın ne kadar basit veya karmaşık olduğu ile ilgilidir. Bir uydu görüntüsündeki bir ev ve su drenajına bakarak ikisi arasındaki ilişkiyi anlamak için çok hızlı bir bilim adamı olmak gerekmez. Çok daha karışık bilgileri derlemek ve bu bilgileri diğer kaynaklardan elde edilen coğrafik bilgilerle birleştirmek günümüz teknolojisi ve yazılımları sayesinde bir kaç saatlik bile eğitimle mümkün olabilmektedir.

Çalışmamız için gerekli olan görüntüler Landsat-7 uydusundan elde edilmiştir. Landsat programı National Aeronautics and Space Administration (NASA) ve U.S.

Geological Survey (USGS) tarafından yönetilmektedir. Görüntüler USGS'in çevrimiçi web sitesinden (<http://glovis.usgs.gov/>) elde edilmiştir. Site, kayıtlı kullanıcılarına farklı lokasyonlar ve farklı zaman aralıklarında uydu görüntülerini sunmaktadır. Landsat uyduları 1972 den beri yerküre ile ilgili görüntüleri toplamaktadır. Aynı çerçeve için görüntüler 16 günde bir güncellenir. Yani diğer bir deyişle uydu 16 günde bir aynı noktadan geçmektedir.

Uydu görüntüleri birçok harita ve kaynak yönetimi projelerine çözüm olmaktadır; fakat coğrafi verilerin tek kaynağı değildir. Bazı özel uygulamalar için gerekebilecek diğer kaynaklar ise şu şekilde özetlenebilir: Hava fotoğrafları (aerial photography), küçük alanlarda ve 1 m'den küçük yapıları/objeleri haritalamak için düşük maliyetlidir. Birçok hava fotoğrafı siyah-beyaz veya standart renklerde veya yakın-kızılötesidir. Havadan hiperspektral ve Lidar tarayıcılar (Airborne Hyperspectral and Lidar Scanners), uydularda bulunan multi-spektral algılayıcıların uçaklara monte edilmesiyle, yüksek çözünürlüklü, çok bantlı görüntüler elde edilebilir ancak bunlarda hava fotoğrafları gibi geniş alanların görüntülenmesinde düşük maliyetli değildir. Hava Lidar Tarayıcılar, uçağa veya helikoptere monte edilebilen lazer tarayıcılar. Bu tip tarayıcılar yeryüzünün 3-boyutlu modellerinin oluşturulmasında kullanılırlar. Taranmış ve Sayısallaştırılmış Haritalar (Scanned and Digitized Maps), elde bulunan coğrafik verilerin CBS'ye yüklenmesini sağlayan daha ucuz bir yöntemdir. Elde edilen sayısal veriler genellikle eskidir ve hatalar içerebilir. GPS Arazi Çalışmaları (GPS Land Studies), küresel konum belirleme sistemleri kanıtlanmış, son derece doğru bilgileri içeren haritalama metodudur. Gerçekte, genellikle belli sayıda kontrol noktasının koordinatlarının belirlenmesiyle, uydu görüntüsünün koordinat doğruluğunu arttırmak için kullanılır. Ancak zaman alıcı ve pahalı bir sistemdir ve belli noktalardan veri sağladığı için alanın bütünü için bilgi aktaramaz [47].

2.5. Mekânsal Veritabanları

Veritabanı kavramı ilk olarak 1980'li yıllarda ortaya atılmış olmasına rağmen; günümüzde hemen hemen tüm veri kullanılan alanlarda Veritabanı Yönetim Sistemleri (VTYS) olmadan hiçbir şey yapılamaz hale gelmiştir. Basit bir web uygulamasından, devasa kuruluşların ağır verilerine kadar, günümüzde birçok alanda

veritabanı uygulamalarına ihtiyaç duyulmaktadır. İşletim sistemlerinden sonra en popüler ve en çok gelir getiren yazılımlar VTYS yazılımlarıdır. Kullanılan veri modeline göre yani verileri saklama ve onlara erişme bakımından veritabanı yönetim sistemleri hiyerarşik, ağ, ilişkisel, nesne ve nesne ilişkisel veritabanı modelleri olarak kategorilere ayrılırlar.

Hiyerarşik veritabanları, ana bilgisayar ortamlarında çalışan yazılımlar tarafından kullanılmakta olup bilgileri bir ağaç yapısında saklar. Kök (root) olarak bir kayıt ve bu köke bağlı dal (branch) kayıtlar bu tip veritabanının yapısını oluşturur. Ağ veritabanları hiyerarşik veritabanlarının yetersiz kalmasından dolayı ortaya konulmuş bir veritabanı türüdür. Ağ veritabanları verileri ağaçların daha da gelişmiş hali olan graflar (ağacın kendisi de özel bir graftır) şeklinde saklarlar. İlişkisel veritabanları, matematikteki ilişki teorisine (the relational theory) dayanır. Bu sistemde veriler tablo şeklinde saklanır. Bu veritabanı yönetim sisteminde; veri alış verişi için özel işlemler kullanılır. Bu işlemlerde tablolar operantlar olarak kullanılır. Tablolar arasında ilişkiler belirtilir. Bu ilişkiler matematiksel bağıntılarla (ilişkilerle) temsil edilir. Günümüzde hemen hemen tüm veritabanı yönetim sistemleri ilişkisel veri modelini kullanırlar.

Nesneye yönelik veritabanı da, C++ gibi nesneye dayalı bir dille (OOPL) yazılmış olan ve yine C++ gibi nesneye dayalı bir dille kullanılan veritabanı anlamına gelir. Günümüz teknolojisinde yüzde yüz nesneye yönelik bir veritabanı yaygın olarak kullanıma sunulmuş değildir. İlişkisel veritabanları ile karşılaştırıldığında; nesneye yönelik veritabanlarının sahip olduğu üstünlükler şunlardır:

- Nesnelere, bir tabloda yer alan bir kayıttan çok daha karmaşık yapıya sahiplerdir ve daha esnek bir yapıda çok daha kullanışlı düzenlenebilirler.
- Nesneye dayalı bir veritabanında, yapısı gereği arama işlemleri çok hızlı yapılabilir. Özellikle büyük tablolarla uğraşırken ilişkisel veritabanlarından çok daha hızlı sonuca ulaşırlar. Ancak çalışma mantığı tümüyle değişir.

Tüm bu özellikler tamamen nesneye yönelik olan veritabanları için geçerlidir. Bazı ilişkisel veritabanları ile çalışan yazılımlarda da nesnelere bazı özelliklerini

kullanılır, ama nesneye yönelik veritabanı bunu kendini ilişkisel veritabanı kurallarına uydurarak gerçekleştirebilir. Nesne ilişkisel veritabanı yönetim sistemleri ise soyut veri tipleri (abstract data types) ile obje tabanlı tasarım prensiplerini birleştiren veritabanı yönetim sistemleridir. Mekânsal nesnelere ait verileri saklayabilen ve sorgulayabilen mekânsal veri tabanları nesne ilişkisel veritabanı modelini kullanır.

Tipik veri tabanları nümerik ve karakter veri tiplerini anlayabilirken, mekânsal veri tabanlarında mekânsal veri tiplerini (nokta, çizgi ve poligon) analiz edebilir. Bu veri tipleri geometri ya da özellik olarak bilinir. Mekânsal veriler; nokta, çizgi ve alansal olarak ifade edilebilen coğrafi haritalar, nehirler, yollar, adalar veya piksel gruplarından oluşan uydu görüntüleri, sayısal yükseklik modelleri ve hava fotoğrafları olabilir. Mekânsal veri tabanları üzerinde; Kocaeli'nin nüfusu kaç, Kocaeli'de toplam kaç tane ilçe var, A harfi ile başlayan ilçelerini listeleyin gibi tipik SQL sorgularına ek olarak mühendislik fakültesine en yakın iki restoran nerededir (yakınlık/proximity), Kocaeli'de şehirlerarası otobüs terminali hangi ilçededir (içerme/containment), Türkiye'ye komşu ülkeler hangileridir (bitişiklik/adjacency) ve Kocaeli'nin hangi ilçeleri üzerinden tren yolu geçer (kesişim-örtüşme /intersection-overlap) gibi mekânsal sorgulamalar yapılabilir.

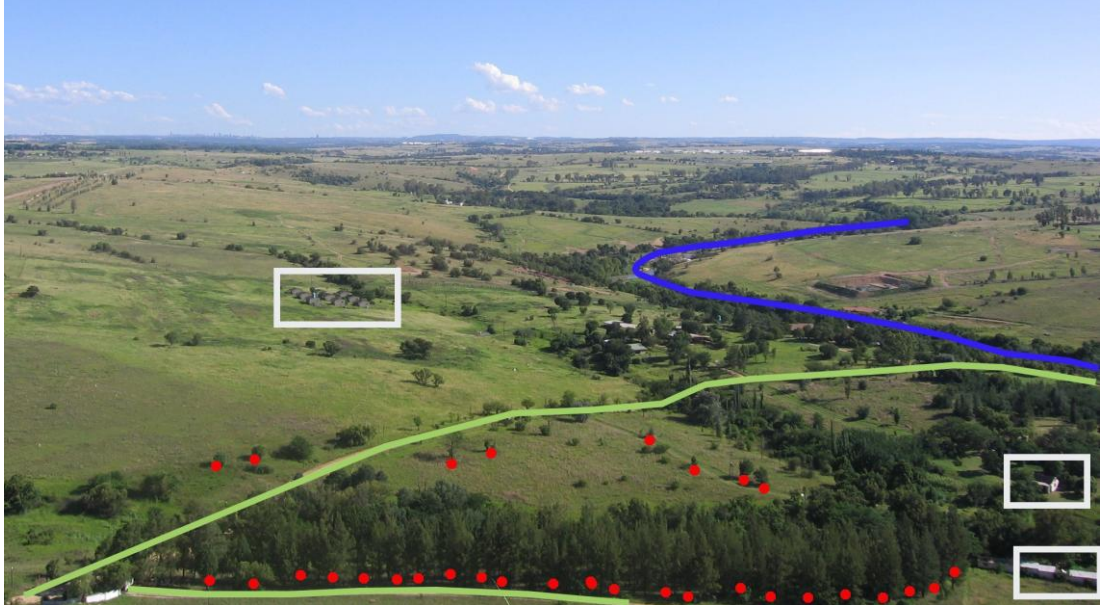
Diğer ticari ya da açık kodlu veritabanlarında bulunabilecek özelliklerin hemen hemen hepsini kapsayan ücretsiz ve açık kaynak kodlu veritabanı yönetim sistemlerinden en iyi performans veren ve güvenli olanlarından biri PostgreSQL'dir. Veritabanları için nesne ilişkisel veri modelini kullanan ve SQL standart sorgu dilini destekleyen veritabanı yönetim sistemi PostgreSQL, diğer birçok gelişmiş veritabanında olduğu gibi dışarıdan kendisine bağlanan istemciler ile belirli bir dilde konuşup anlaşabilmek için bir sunucu/istemci protokolüne sahiptir. PostgreSQL veritabanına ulaşmak için PostgreSQL'in soket bağlantısı üzerinden basit okuma/yazma sistem çağrılarında bulunmak gibi bir iç işleyişi ile ilgili hiçbir çalışma mekanizmasından haberdar olmak zorunda kalınmadan, programlama dilinin sunduğu API kütüphanesinden faydalanılarak kolaylıkla veritabanı ile etkileşime geçip, ilgili sorgulamalar gerçekleştirilebilir.

2.6. Raster ve Vektör Veri

CBS'ye ilişkin uygulama ve projelerin gerçekleştirilmesi ancak uygun yapıda verilerin mevcut olmasına bağlıdır. Bu nedenle CBS'de veri önemli bir unsurdur. Veriler grafik (graphic) ve grafik-olmayan (non-graphic) nitelikte olup farklı kaynaklardan değişik yöntemlerle toplanarak konumsal bilgi analizlerinde kullanılacak hale dönüştürülür.

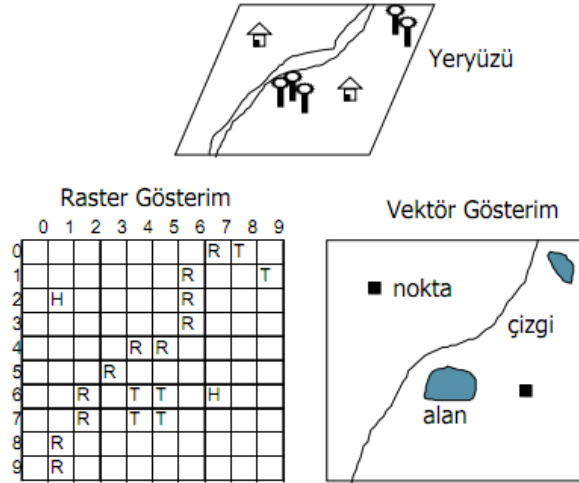
Yeryüzünde veya uzayda konumlanmış nesnelere ve olayların her biri coğrafi varlık (entity) olarak bilinir. CBS'nin verimli bir şekilde çalışabilmesi, yaşanan dünyadaki coğrafi varlıklar arasındaki doğal ve yapay ilişkilerin gerçekte olduğu gibi bir sistem dâhilinde modellenmesiyle mümkündür. Bunu sağlamak için gerçekte var olan tüm detayların coğrafi özellikleriyle ve aralarındaki ilişkilerle birlikte koordinat referanslı olarak tanımlanması gerekir. Böyle bir tanımlama için coğrafi varlığın konumu, büyüklüğü ve şekli hakkında bilgi verecek grafik veri yanında, özneliği (attribute) hakkında bilgi verecek grafik olmayan verilere de ihtiyaç vardır. Coğrafi varlıkları niteleyen unsurlar coğrafi veri olarak bilinirler. Coğrafi veriler doğal nitelikte olabildikleri gibi (örneğin nehirler, ormanlar, adalar, kıyılar vb.) yapay nitelikte (örneğin yollar, binalar, boru hatları vb.) de olabilirler.

Coğrafi veriler incelendiğinde bu verilerin üç temel unsurdan meydana geldiği görülmektedir. Bunlar; nokta (point), çizgi (line), poligon (polygon) şeklindeki geometrik yapılardır. Gerçek dünyadaki veriler irdelendiğinde bunların sadece bu üç temel geometrik yapıda olduğu görülmektedir. Örneğin ağaç, elektrik direği, istasyon, yerleşim merkezleri vb. coğrafi detaylar noktayla; akarsu, yol, su hattı, demir yolu vb. coğrafi detaylar çizgiyle; parsel, imar adası, orman, yerleşim alanları vb. coğrafi detaylar poligonla ifade edilirler. Dolayısıyla bir haritanın sadece nokta, çizgi ve poligonlardan meydana geldiğini söylemek mümkündür. Şekil 2.2'de de ağaçlar kırmızı noktalarla; nehirler mavi, yollar yeşil çizgilerle; evler beyaz poligonlarla ifade edilmiştir [48].



Şekil 2.2. Gerçek dünyadaki verilerin vektörel temsilleri

CBS’de konumsal veri modelleri Şekil 2.3’te görüldüğü gibi iki şekilde olmaktadır. Bunlar; vektörel veri modeli (vector data model) ve hüresel veri modeli (raster data model)’dir.



Şekil 2.3. Coğrafi veri modellerinin gösterimi

Coğrafi veriler, vektörel veri modelinde tıpkı bir çizgisel harita görünümüne sahiptir. Bu görünümde noktalar, sabit alanların çok küçük boyutlu şekillerini; çizgiler, süreklilik ve alan özelliği gösteren yine çok küçük boyutlu coğrafi varlıkları; poligonlar, homojen yapıya sahip bütünlük gösteren coğrafi varlıkları temsil eder. CBS vektör verisinin yardımıyla mekânsal analiz de denilen şu tip sorulara cevap

verebilir: Bütün insanların rahatça erişebileceği bir hastane nereye yapılabilir? A köyünü çevreleyen yolun adı nedir? X ve Y arazileri birbirine komşu mudur? Haydarpaşa-Adapazarı treni Kocaeli'nin hangi ilçelerinden kaçar metre uzunlukla geçer?

Coğrafi özelliklerin gösterimleri için kullanılan bir diğer veri modeli de hücreli (raster) veri modelidir. Raster gösterim, aynı coğrafi özelliklerin çekilmiş bir fotoğrafı gibidir. Raster veri daha çok vektörel özelliklere ayrılamayan bilgileri göstermek için kullanılır. Böyle bir fotoğrafın büyüteç altında incelenmesiyle görülecektir ki çok küçük boyutta, farklı renklere sahip kare biçimindeki kutucukların (gridlerden) bir araya gelmesiyle bütün bir görüntü oluşmaktadır. Raster bir haritada gösterilen coğrafi varlığın gerçeği yansıtırma gücü (konum hassasiyeti), harita ölçeğine ya da görüntünün elde edilme kalitesine bağlıdır. Raster gösterimde, bu hassasiyet piksel boyutuna göre değişen ayırma veya çözünürlük gücü ile ölçülür. Piksellerin boyutu, bilgisayar veya fotoğraf ortamında mikron biriminde ölçülürken gerçek boyutu (yersel çözünürlük) metre veya santimetre biriminde ölçülebilir.

Raster yapıdaki gösterimlerde piksel boyutunun küçük olduğu alanlar piksel boyutunun büyük olduğu alanlara oranla daha hassa yani ayırma gücü yüksek bir gösterime sahiptir. Piksel boyutunun büyümesiyle görüntü kalitesinde bozulma olmasına karşın görüntünün bilgisayar ortamında saklanması için gerekli bellek büyüklüğü de azalır. Örneğin 50x70 cm boyutundaki bir haritanın bilgisayarda 0.1x0.1 mm'lik piksellerle temsil edildiği kabul edilirse haritada 35 milyon eleman bulunacak ve her bir eleman için 1 bit kullanıldığında 4MB, 8 bit kullanıldığında 33 MB bellek gereksinimi olacaktır. Raster veri yapılarının gösterim gücünün artırılması için piksel boyutunun azaltılarak piksel sayısının çoğaltılması ve renk derinliğinin artırılması gerekir. Ancak raster gösterim, bilgisayarın belleğinde vektörel tekniğe göre daha fazla yer kaplar. Özellikle vektörel gösterimde haritalarda boş alan olarak gözüken kısımlar, raster gösterimde dolu alan olarak dikkate alındığından bu alanlar bellekte saklanmak durumundadır.

Raster verilerin uęraşmanın bazı dezavantajları vardır. Depolanması için çok büyük dosya büyüklüklerine ihtiyaç duyarlar. Piksel boyutlarının büyümesiyle veri

azalmasına karşın gerçek görünüm bozulmakta ve önemli derecede bilgi kaybı söz konusu olmaktadır. Raster ve vektör verilerin karşılaştırılması Tablo 2.3'te özet olarak verilmiştir.

Tablo 2.3. Raster-vektör veri karşılaştırılması

Vektör veri	Raster veri
Nokta, doğru ve poligon ile gösterilir.	Her şey piksel gridinden oluşur.
Dosya boyutu küçüktür.	Dosya boyutu büyüktür.
Networkler çok iyi temsil edilir.	Networklerin temsili iyi değildir.
Boyut büyütüldüğünde özellikler çok detaylı ve tam bir şekilde yansıtılır.	Boyut büyütüldüğünde görünüm bozulur, bilgi kaybı olur.
Topoloji tabanlı analizlere uygundur.	Simülasyon ve modellemeye uygundur.
Sürekli veriler (toprak kullanımı vb.) temsili için uygun değildir.	Sürekli veriler (yükseltmeler vb.) temsili için uygundur.
Koordinat sistem dönüşümleri çok az zaman alır.	Koordinat sistem dönüşümleri çok zaman alır.
Veri yapıları karmaşıktır.	Veri yapıları basittir.

3. MİMARİ: ADA UYDU GÖRÜNTÜLERİNİN VEKTÖRİZASYONU VE MEKÂNSAL VERİTABANLARINDA ANALİZİ

Çalışmamız için gerekli olan görüntüler Landsat-7 uydusundan elde edilmiştir. Landsat programı National Aeronautics and Space Administration (NASA) ve U.S. Geological Survey (USGS) tarafından yönetilmektedir. Landsat-7 görüntüler; USGS'in çevrimiçi web sitesinden (<http://glovis.usgs.gov/>) elde edilmiştir. Landsat uyduları 1972 den beri yerküre ile ilgili görüntüleri toplamaktadır. Aynı çerçeve-mozaik-için görüntüler 16 günde bir güncellenir. Yani diğer bir deyişle uydu 16 günde bir aynı noktadan geçmektedir. Mozaikler meta-veri (Şekil 3.1) olarak adlandırılan bilgi dosyaları ile sunulmaktadır. Meta veri, görüntü hakkında çerçeve koordinat değerleri (longitude (X) ve latitude (Y)), görüntünün çekildiği tarih ve bulutlanma cinsinden gürültü seviyesi vb. bilgiler içermektedir.

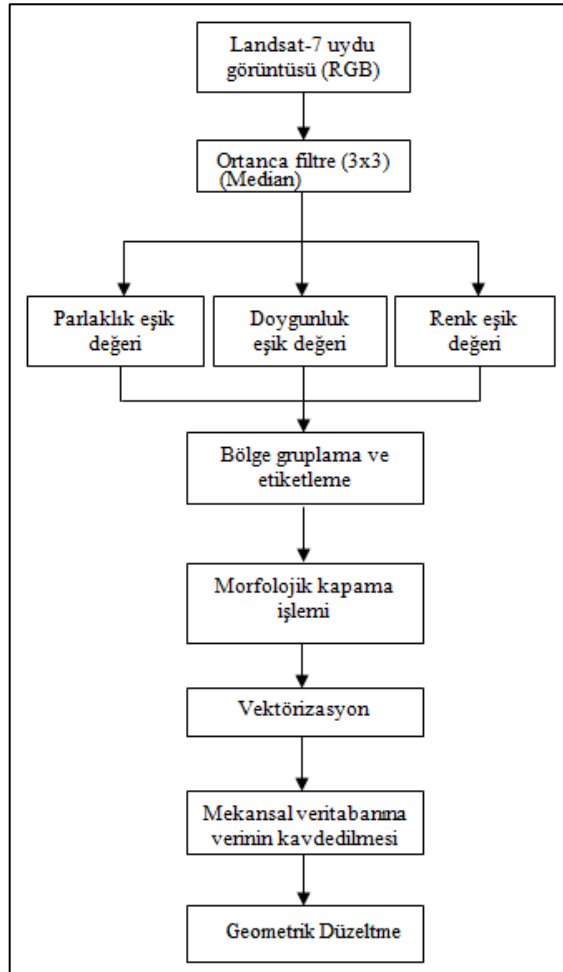
Stop Time	2011:122:08:16:30.2449375
Image Quality VCID 1	9
Image Quality VCID 2	9
Gap Phase Source	DE
Gap Phase Statistic	10.77906
Cloud Cover	3.58
Cloud Cover Quadrant Upper Lef	0.86
Cloud Cover Quadrant Upper Rig	7.97
Cloud Cover Quadrant Lower Lef	0.04
Cloud Cover Quadrant Lower Rig	5.44
Sun Elevation	62.31720352
Sun Azimuth	128.64250183
Full Aperture Calibration	N
Scene Center Latitude	34.60741 (34°36'26.68"N)
Scene Center Longitude	33.10603 (33°06'21.71"E)
Corner Upper Left Latitude	35.55730 (35°33'26.28"N)
Corner Upper Left Longitude	32.29578 (32°17'44.81"E)
Corner Upper Right Latitude	35.25615 (35°15'22.14"N)
Corner Upper Right Longitude	34.36942 (34°22'09.91"E)
Corner Lower Left Latitude	33.95039 (33°57'01.40"N)
Corner Lower Left Longitude	31.86244 (31°51'44.78"E)
Corner Lower Right Latitude	33.65514 (33°39'18.50"N)
Corner Lower Right Longitude	33.89702 (33°53'49.27"E)

Şekil 3.1. Örnek bir mozaığın meta verileri

Landsat-7 uydu ada görüntüsü üzerinde zaman-mekânsal değişimleri tespit etmek için renk uzayı dönüşümü segmentasyonu, bölge gruplama-etiketleme ve sınır piksellerinin takibinedayalı vektörizasyon işlemi toplam beş işlemden oluşmaktadır:

- 1.Ham görüntüye gürültü etkisini azaltmak için radyometrik düzeltme getirilmesi,
- 2.Ada olmayan objelerin elemine edilmesi,
- 3.Bölge gruplama ve etiketleme,
- 4.Poligon oluşturulması için sınırların izlenmesi,
- 5.Uydu görüntülerine geometrik düzeltme getirilmesi.

Yukarıda bahsedilen adımları içine alan metotlar dizisi Şekil 3.2’de gösterilmiştir.



Şekil 3.2. Önerilen yöntemin akış şeması

3.1. Veri Ön İşleme ve Yumuşatma

Görüntü ve sinyal işleme konularında, gürültü temizlemek için kullanılan filtrelerden birisi de ortanca filtredir (median filter). Amaç belirli bir pencere aralığındaki sayıların ortancasını alarak aşırı büyük atlamaları kaldırmaktır. Yani filtre uygulandıktan sonra resimde bulunan ve konumlarından belirgin şekilde ayrılan piksellerin tespit edilerek temizlenmesi sağlanır. Çoğu yumuşatma filtresi gürültüleri kaldırmasına rağmen kenarlarda blurlaştırarak yok eder. Ortanca filtre kenar noktalarını muhafaza eder. Sulu bölge (deniz, okyanus vb.) görüntüleri üzerindeki gürültüleri yok etmek için ortanca filtresinden yararlanılmıştır.

Basit bir ortanca filtresinin çalışma mantığını pencere genişliği (window size) 3 olacak şekilde aşağıdaki sayılar için şu şekilde gösterebiliriz:

$g = [2 \ 32 \ 2 \ 1 \ 2 \ 8 \ 9]$ //giriş renk değerleri

$\varphi [0] = [2 \ 2 \ 32] = [2 \ 2 \ 32] \Rightarrow 2$ // burada ilk sayıyı tekrar ettik çünkü katar 3 olmalı

$\varphi [1] = [2 \ 32 \ 2] = [2 \ 2 \ 32] \Rightarrow 2$ // ilk dizi giriş dizisinin ilk 3 sayısıdır. İkinci dizi ise sıralanmış halidir ve sonuç olarak ortanca değer 2 bulunur.

$\varphi [2] = [32 \ 2 \ 1] = [1 \ 2 \ 32] \Rightarrow 2$

$\varphi [3] = [2 \ 1 \ 2] = [1 \ 2 \ 2] \Rightarrow 2$

$\varphi [4] = [1 \ 2 \ 8] = [1 \ 2 \ 8] \Rightarrow 2$

$\varphi [5] = [2 \ 8 \ 9] = [2 \ 8 \ 9] \Rightarrow 8$

$\varphi [6] = [8 \ 9 \ 9] = [8 \ 8 \ 9] \Rightarrow 9$

$\varphi = [2 \ 2 \ 2 \ 2 \ 2 \ 8 \ 9]$ //çıkış renk değerleri

Yukarıdaki örnekte görüldüğü üzere ortanca filtresi uygulandıktan sonra çok büyük bir sayı olan ve yakınlarında başka benzer sayı bulunmayan 32 sayısı elenmiştir. Benzer şekilde yakınlarında benzeri bulunmayan 1 sayısında çok fazla 2 sayısı arasında kalıp bu sayılar tarafından boğulmuştur.

2-boyutlu resimlerde bu filtre kullanılırken 2-boyutlu pencere boyutu uygulanır. Örneğin 3×3 gibi. Bu durumda 9 hücreli dizinin içinde kalan sayılar sıralanarak ortalama değeri yeni değer olarak yazılır. Aşırı hesaplama yükünden kurtulmak için Eşitlik (3.1)'de verilen 3×3 'lük bir pencereden yararlanıldı[49].

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{vmatrix} \quad (3.1)$$

a_1 - a_9 ile gösterilen değerler görüntü üzerinde filtre penceresinin denk geldiği gri renk değerleridir.

3.2. Renk Uzayı Dönüşümleri ile Ada Dışındaki Nesnelere Yok Etme

Landsat-7 uydusundan alınan görüntüler RGB renk uzayı formundadır (renkli görüntüdür). Böylece uydu görüntüsünün karakteristik özelliklerini çıkarabilmek için RGB renk modelinden yararlanılabilir. Bu çalışmada YCbCr ve HSV olmak üzere iki tane daha renk modelinden yararlanıldı. YCbCr modeli daha çok dijital fotoğraf sistemleri ve videolarda kullanılan bir renk modelidir. Y görüntündeki parlaklığı, Cb mavi bileşenle referans değer arasındaki farkı, Cr kırmızı bileşenle referans değer arasındaki farkı temsil eder. Benzer şekilde HSV de o rengi diğer renklerden ayıran H değeri, parlaklığa bağlı renk oranını ifade eden S değeri ve bir yerdeki RGB değerlerinin ortalamasını gösteren V değerinden oluşmaktadır. Bütün bu bileşenler RGB renk modelinden Eşitlik (3.2)'den Eşitlik (3.8)'e kadar aşağıda verilen ifadelerden elde edilebilir,

$$Y=65.481R + 128.553G + 24.966B +16 \quad (3.2)$$

$$C_b=-37.797R -74.203G + 112.000B +128 \quad (3.3)$$

$$C_r=112.000R - 93.786G -18.214B + 128 \quad (3.4)$$

$$H=\begin{cases} \theta, & B \leq G \\ 360-\theta, & \text{otherwise} \end{cases} \quad (3.5)$$

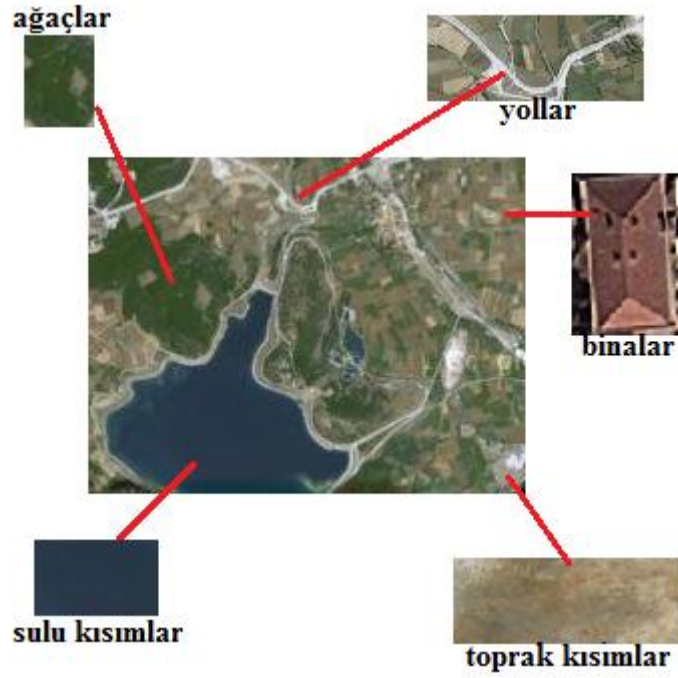
$$\theta=\cos^{-1} \left\{ \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2(R-B)(G-B)}} \right\} \quad (3.6)$$

$$S=1-\frac{3}{(R+G+B)} a \quad (3.7)$$

a değeri R, G ve B değerlerinden en küçük değere sahip olana eşittir.

$$V = \frac{(R+G+B)}{3} \quad (3.8)$$

Renkli uydu görüntüsünde bulunan çoğu yeryüzü objesini arařtırmacılar Őekil 3.3'te görüldüğü gibi beř gruba ayırmıřlardır: ağalar, yollar, binalar, sulu bölgeler ve kumlu-toprak bölgeler[50]. Bu elementlerin YCbCr ve HSV renk modellerindeki bileřenlerinin karakteristik özellikleri incelenmiřtir.



Őekil 3.3. Uydu görüntülerindeki beř element

Bir kiři ıplak bir gözle Őekil 3.3'teki beř farklı elementi renk ve parlaklıklarına göre birbirinden ayırt edebilir. Aynı ayırt etme özelliğini bilgisayarların da yapabilmesi için uydu görüntüsünün renk uzayı bileřenlerini (parlaklık, doygunluk ve renk deęerleri gibi) analiz edebilmesi gerekmektedir. Bu bileřenler görüntüyü YCbCr ve HSV modeline döndürmekle elde edilebilir. Bu beř elementin bileřenleri arasındaki iliřki Tablo 3.1'de gösterilmiřtir.

Tablo 3.1. Beş element bileşenleri arasındaki ilişki

	Ağaçlar	Yollar	Binalar	Toprak kısımlar	Sulu kısımlar
Parlaklık(Y)	25-100	110-160	50-150	100-200	50-80
Renk(H)	0.05-0.3	0.05-0.2	0-0.1	0.05-0.1	0.1-0.45
Doygunluk(S)	0.05-0.4	0.1-0.25	0.1-0.5	0.3-0.5	0.05-0.2
Gri ton(V)	0.05-0.5	0.15-0.8	0.2-0.6	0.4-1	0.2-0.3

Cb, Cr ve V gibi bileşenler işe yarar bilgiyi elde etmede zayıf kaldıkları görüldüğünden bu bileşenler kullanılmamıştır. Renk uzayı bileşenleri arasındaki ilişki aşağıdaki gibidir:

- Binalar renk değer aralığı 0-0.1 arasında olmakla en düşük seviyededir.
- Ağaçların parlaklık değeri 100'ün altındadır.
- Topraklı bölgeler ile yolların parlaklık aralığı birbirine yakındır.
- Yolların renk aralığı 0.05-0.2 arasındadır.

Uydu görüntülerinden adada var olan kısımları elde etmek için sulu bölgeleri elemine etmemiz yeterlidir. Sulu kısımları aşağıda verilen birkaç eşikleme ile yok edebiliriz:

- Parlaklık eşik değerini 50 ile 80 arasına setleyerek ($50 \leq Y \leq 80$)
- Renk eşik değerini 0.1 ile 0.45 arasına setleyerek ($0.1 \leq H \leq 0.45$)
- Gri ton eşik değerini 0.2 ile 0.3 arasına setleyerek ($0.2 \leq V \leq 0.3$ bazı ağaçlı bölgeleri elde etmek için)
- Doygunluk eşik değerini 0.05 ile 0.2 arasına setleyerek ($0.05 \leq S \leq 0.2$)

İlk olarak uydu görüntüsü YCbCr ve HSV modellerine dönüştürülür. Yok etme işlemi (sulu kısımları atma) bütün pikselleri Eşitlik (3.9) ile verilen testiyle geçirmekle yapılır.

$$(50 \leq Y \leq 80) \text{ VE } (0.1 \leq H \leq 0.45) \text{ VE } (0.2 \leq V \leq 0.3) \text{ VE } (0.05 \leq S \leq 0.2) \text{ (3.9)}$$

RGB renk uzayındaki uydu görüntüsünü YCbCr ve HSV renk uzaylarına dönüştürdükten sonra her bir piksel Eşitlik (3.9) ile verilen test işlemine tabii tutulur (Eşitlik (3.10)'da gösterildiği gibi). Test işlemiyle istenmeyen pikseller (sulu

bölgeler) elemine edilir. Test işlemi sonunda $b(i,j)$ ikili görüntüsü (Şekil 3.4a) oluşturulur:

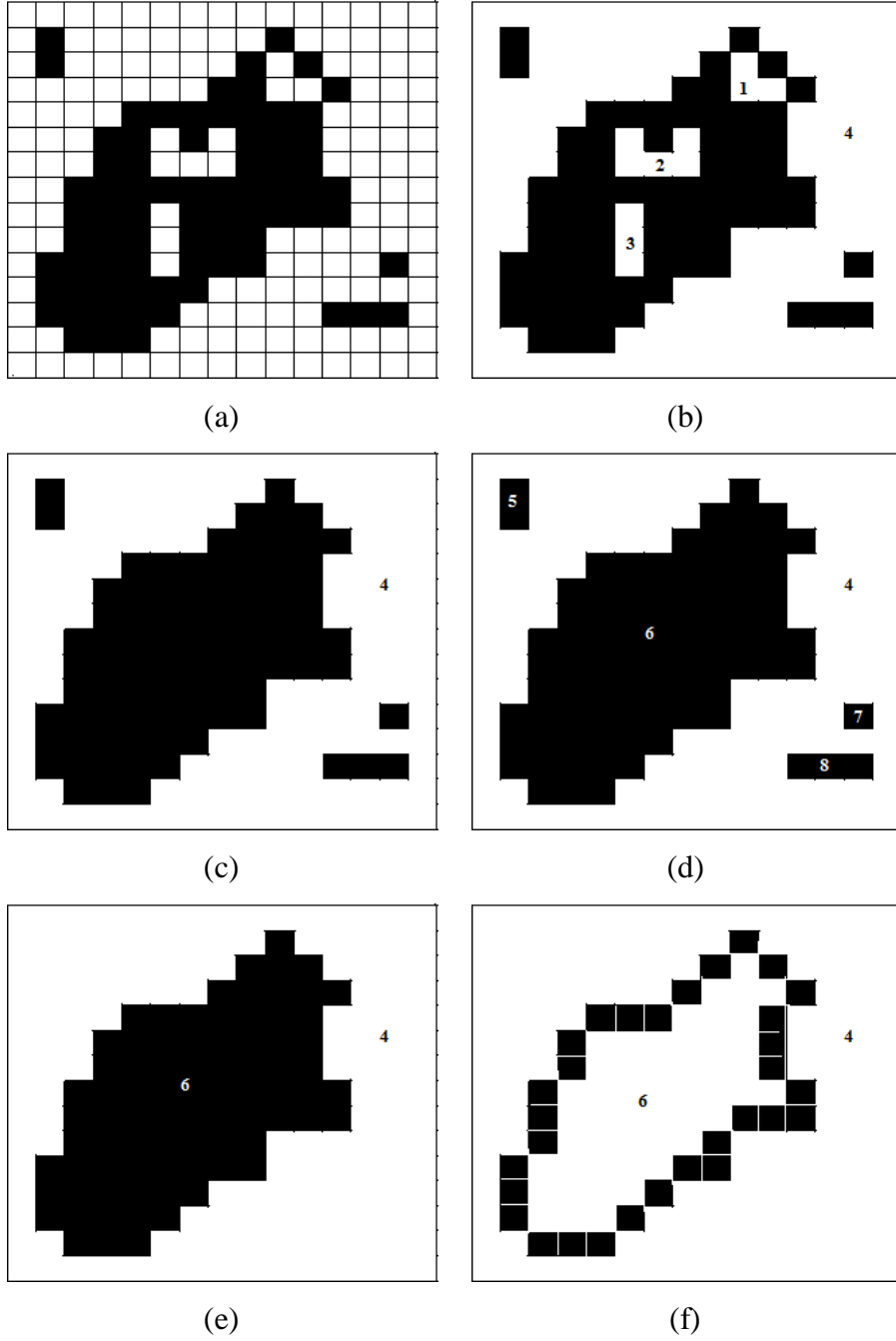
$$b(i,j)=\left\{ \begin{array}{l} 255, \text{ test işlemini geçerse} \\ 0, \text{ geçemezse} \end{array} \right\} \quad (3.10)$$

3.3. Bölge Gruplama ve Etiketleme

Bir önceki adımın sonunda ikili görüntüde temel olarak iki çeşit nesne kalır: sulu kısımlar (içinde adaya benzer gürültüler olan) ve ada(içinde suya benzer gürültüler olan) (Şekil 3.4). Ada veya sulu kısımlar içindeki bitişik komşu pikseller daha yüksek seviye görüntü objelerine gruplandırılır. Görüntüdeki objeler şekil ve büyüklüklerine göre karakterize edilir ve her biri numarandırılarak etiketlenir. Nesneleri görüntü objesi olarak tanımlamak diğer objelerden kolaylıkla ayrılmasını sağlar.

Bölge gruplama ve etiketleme uygulaması taşma doldurma (flood-fill) algoritmasına dayanmaktadır. Flood-fill algoritması, sınırlanmış bir alanın belirlenip istenen bir renkte doldurulmasını (boyanmasını) sağlayan bir algoritmadır. Algoritma bütün pikseller rekürsif olarak boyanıp etiketlenene kadar çalışmaktadır. Bu algoritma ikili $b(i,j)$ görüntüsüne uygulanır.

Ada görüntüsü içinde akarsu, göl, sulak alanlar ve diğer suya benzer görüntü gürültüleri gibi küçük sulu bölgeler olabilir. Bu gerçek gözüne alınarak, bu tür bölgeler yok edilerek ada görüntüsüne dönüştürülmelidir. Şekil 3.4b ve Şekil3.4c karşılaştırıldığında 1, 2 ve 3 ile etiketlenen bu tür bölgeler yok edilmiştir. Benzer şekilde sulu bölgeler içinde de küçük gürültü görüntü bölgeleri olabilmektedir. Bu tür küçük görüntülere buz dağı, gemiler ve diğer küçük nesnelere örnek olarak verilebilir. Bunlar da yine gerçek nesnelere olmadığı için yok edilerek sulu bölge görüntüsüne dönüştürülmelidir. Şekil 3.4d ve Şekil 3.4e karşılaştırıldığında 5, 7 ve 8 ile etiketlenen bu tür bölgeler yok edilmiştir. Bölge gruplama ve etiketleme sonunda iki tür bütün bir nesne kalır: tek ada görüntüsü ve tek sulu bölge.



Şekil 3.4. Bölge gruplama ve etiketleme

3.4. Morfolojik Kapanma İşlemi

Matematiksel morfoloji, iki değerlikli ve gri seviyeli görüntülerden geometrik bilgi çıkartan bir araçtır. Bir görüntü işleci elde etmek için yapıtaşı elemanı (structuring element) olarak bilinen bir şekil göstergesi kullanılır. Görüntü işlecinin çıktısı, bu göstergenin verilen görüntüyle örtüşüp örtüşmediğine bağlıdır. Çıkarılan

bilginin kullanılan göstergenin şekline ve büyüklüğüne bağlı olacaktır. Aşınma (erosion) ve genişleme (dilation), matematiksel morfolojinin en temel işlemleridir.

Aşınma operatörünün temel etkisi görüntüyü kenarlarından erozyona uğratmaktadır. Sonuçta piksel grupları ve aralarındaki boşluklar küçülür seçilen yapıtaşı elemanına göre gürültülü şekiller düzelir. Genişleme operatörünün temel etkisi görüntüyü kenarlarından genişletmektir. Sonuçta piksel grupları büyür ve aralarındaki boşluklar küçülür. Kapanma işlemi ise genişleme işlemi takip eden aşınma işlemidir. Bir önceki adımda elde edilen görüntü üzerinde kapama işlemi uygulayarak köşeli çentikli kenarlar yumuşatılır.

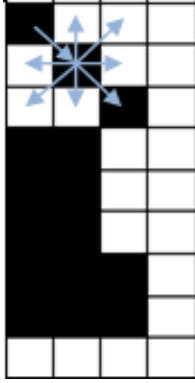
3.5. Sınır Çıkarımı ve Vektörizasyon

Sınırlar, sayısal görüntüdeki şeklin çevresini gösteren çizgilerdir. Raster bir ada görüntüsünde sınır çizgileri takip edilerek kapalı bir poligon elde edilebilir. Sınır değerleri elde edilirken $b(i,j)$ görüntüsünde sulu bölgeler arka plan olarak kabul edilir ve adayı temsil eden piksellerden arka planla komşu olan piksel sınır pikseli olarak işaretlenir. Bu sınır pikselleri 3×3 boyutlu bir maskenin $b(i,j)$ görüntüsü üzerinde gezdirilmesi ile elde edilir. Eşitlik (3.11)'de belirtildiği gibi sınır pikseller 255, sınır olmayanlar ise 0 olarak kodlanarak raster formatta $c(i,j)$ görüntüsü elde edilir (Şekil3.4f).

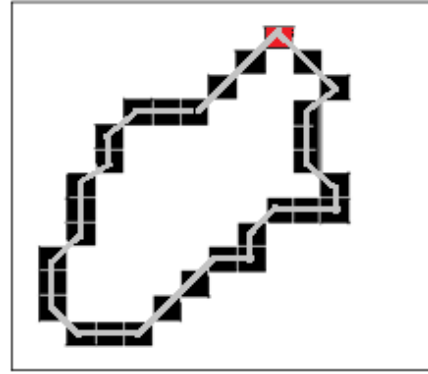
$$c(i, j) = \begin{cases} 255, & \text{if } b(i,j)=255 \text{ ve } (b(i-1,j) \vee b(i+1,j) \vee b(i,j-1) \vee b(i,j+1)=0) \\ 0, & \text{aksi durumda} \end{cases} \quad (3.11)$$

Bu aşamadan sonra sınır pikselleri takip edilerek oluşturulacak poligonun koordinat değerleri kaydedilir. Oluşturulacak poligon nesnesinin başlangıç ve bitiş noktaları aynı olmalıdır. Bunun için öncelikle adanın sınırı üzerinden bir nokta başlangıç noktası olarak seçilir. Amaç sınırdaki bir sonraki pikselin bulunmasıdır. Sonraki piksel, ilgilenilen piksele komşu olan bir pikseldir. Sınır çıkarma algoritması geldiği yön hariç yedi izleme yönüne izin verir (Şekil 3.5). Aslında sonraki piksel, önceki piksel ve ilgilenilen piksel tarafından tanımlı yönde bulunan pikseldir. Eğer sonraki piksel bu yönde yoksa bu yönle (referans yönü) arasında minimum açığa sahip olan piksel sonraki pikseldir. Bu pikselin koordinatı şeklin sınırını ifade eden

koordinat deęerleri arasınadâhil edilmelidir. Őeklin sınırında bulunan bir sonraki pikseli bulma iŐlemi baŐlangıŐ noktası (Őekil 3.6'daki kırmızı piksel) olarak seŐilen piksele rastlanana kadar devam ettirilir (Őekil 3.6).



Őekil 3.5. 8-yönlü komŐulukve izinli yönler



Őekil 3.6. Sınır piksellerinin vektörleŐtirilmesi

3.6. Vektör Verileri Mekânsal Veritabanlarına Kaydetme

Tipik veri tabanları nümerik ve karakter veri tiplerini anlayabilirken, mekânsal veri tabanlarında mekânsal veri tiplerini (nokta, çizgi ve poligon) analiz edebilir. Bu veri tipleri geometri ya da özellik olarak bilinir. Mekânsal veriler; nokta, çizgi ve alansal olarak ifade edilebilen coęrafi haritalar, nehirler, yollar, adalar veya piksel gruplarından oluŐan uydu görüntüleri, sayısal yükseklik modelleri ve hava fotoęrafları olabilir.

Veritabanları için nesne ilişkisel veri modelini kullanan ve SQL standart sorgu dilini destekleyen veritabanı yönetim sistemi PostgreSQL, dięer birçok geliŐmiŐ veritabanından olduęu gibi dıŐarıdan kendisine baęlanan istemciler ile belirli bir dilde konuŐup anlaŐabilmek için bir sunucu/istemci protokolüne sahiptir. PostgreSQL veritabanına ulaŐmak için PostgreSQL'in soket baęlantısı üzerinden basit okuma/yazma sistem çağrılarında bulunmak gibi bir iç iŐleyiŐi ile ilgili hiŐbir ھاalışma mekanizmasından haberdar olmak zorunda kalınmadan, programlama dilinin sunduęu API kütüphanesinden faydalanılarak kolaylıkla veritabanı ile etkileŐime geŐip, ilgili sorgulamalar gerŐekleŐtirilebilir. Biz de ھاalışmamızda Java programlama dilinin sunduęu "pgjdbc" API kütüphanesinden yararlandık.

Vektörel biçime çevirdiğimiz ada görüntüsü geometrik olmayan (adanın ismi, alanı ve çevresi) ve geometrik olan (poligon) özellikleri ile PostgreSQL veritabanına kaydedildi. Geometri veri tipi ve bu tipi kullanan fonksiyonlarla işlem yapabilmek için PostgreSQL'e mekânsalPostGIS eklentisi eklendi.

Örneğin 5 tane köşe noktasına sahip bir poligonu mekânsalveritabanına kaydedebilmek için aşağıdaki SQL ifadesi kullanılabilir:

```
CREATE TABLE polygon_region (isim varchar, koordinat geometry);
```

```
INSERT INTO polygon_region VALUES ('island', 'POLYGON((0 10,15 10,15 -5,10 -5,10 0,0 10))');
```

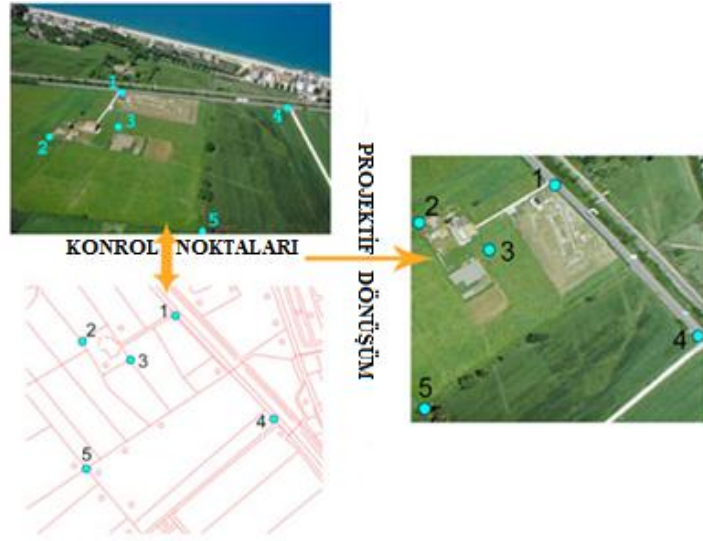
Veritabanına ada görüntülerini poligon olarak kaydettikten sonra geometri sınıfı üzerinde Tablo 3.2'de verilen mekânsal operatörleri gerçekleyebiliriz.

Tablo 3.2. Bazı mekânsal operatörler

Operatörler	Operatörlerin gerçeklediği fonksiyonlar
Envelope	Verilen geometriyi içine alan minimum alana sahip dikdörtgeni geriye döner
Closed	Verilen noktaların başlangıç ve bitiş noktaları aynı ise geriye true değer döner
Boundary	Verilen geometrinin sınırını döner
Intersection	Verilen geometriler ayrık değilse geriye true döner
Contains	Bir geometri diğerinin içinde ise geriye true döner
Overlap	İki geometri üst üste binmiş ise geriye true döner
Area	Poligon veya çoklu-poigonun alanını geriye döner
Length	Çizgi veya çoklu-çizginin uzunluğunu geri döner

3.7. Geo-referanslama

Geo-referanslama, konum veri setini mekânsal referans bilgisine (konuma) bağlama işlemidir. Normalde ham raster verinin koordinat sistemiyle bir bağlantısı yoktur. Bir harita görüntüsü veya uydu görüntüsünden alınan ham bir veri veya benzer bir veri setini koordinat ile ilişkilendirmek için en az üç tane kontrol noktasına ihtiyaç vardır. Eğer üçten fazla nokta olursa transformasyon işleminde hata oranını minimuma indirgenmiş olur. Şekil 3.7'de Beş tane kontrol noktasıyla bir yükseklik görüntüsü referanslanmış bir görüntüye dönüştürülmüştür.



Şekil 3.7. Dönüşüm işlemi

Dönüşümü yapılacak görüntünün koordinat sisteminde (ekran koordinat sistemi) sol alt köşesi (a,b), sağ alt köşesi (c,d), sol üst köşesi (f,g), sağ üst köşesi (k,l) ve dönüştürülecek dünya koordinat sisteminde enlem-boylam ikilisinin sol alt köşesi (u,v), sağ alt köşesi (r,s), sol üst köşesi (p,q), sağ üst köşesi (m,n) olduğu kabul edilirse, koordinat dönüşümü içinaşağıdaki lineer eşitlikleri kullanabiliriz:

$$X=Ax+By+C$$

$$Y=Dx+Ey+F \quad (3.12)$$

Eşitlik (3.12)'deki (x, y) yerine pikselin koordinat değerlerini koyarak (X, Y) enlem-boylam değerlerini bulabiliriz. Mesela dünya koordinat düzlemindeki sol alt köşe değerleri Eşitlik (3.12) yardımıyla şu şekilde hesaplanır:

$$u=Aa+Bb+C \quad (3.13a)$$

$$v=Da+Eb+F \quad (3.13b)$$

Sağ alt ve sol üst köşe noktaları için de Eşitlik (3.13)'e benzer denklemler elde ederek altı bilinmeyeni (A, B, C, D, E, F) bu denklemleri aşağıdaki gibi çözerek bulabiliriz:

$$A = (-gu+du-dp-rb+rg+pb) / (fb-cb+cg-ag+ad-fd) \quad (3.14a)$$

$$B = -(-fu+ap+cu-cp-ar+fr) / (fb-cb+cg-ag+ad-fd) \quad (3.14b)$$

$$C = (-pcb+dap-dfu+rfb+gcu-gar)/(fb-cb+cg-ag+ad-fd) \quad (3.14c)$$

$$D = (-gv+dv-dq-sb+sg+qb)/(fb-cb+cg-ag+ad-fd) \quad (3.14d)$$

$$E = -(-fv+aq+cv-cq-as+fs)/(fb-cb+cg-ag+ad-fd) \quad (3.14e)$$

$$F = (-qcb+daq-dfv+sfv+gcv-gas)/(fb-cb+cg-ag+ad-fd) \quad (3.14f)$$

Enlem-boylam deęerleri desimal derece veya dakika formatında olmak koşuluyla Eşitlik (3.14) ile belirtilen denklemlerdeki bilinmeyenler bulunur.

4. ADA UYDU GÖRÜNTÜLERİ ÜZERİNE DURUM ÇALIŞMASI: MEKÂNSAL VERİTABANLARI ÜZERİNDE TOPOLOJİK SORGULAR VE DEFORMASYON ANALİZLERİ

4.1. Giriş

Bu bölümde, 3. bölümde önerilen vektörleştirme modeline göre örnek bir uydu ada görüntüsü üzerinde durum çalışması yapılmaktadır. Söz konusu görüntü vektörel olarak modellenip mekânsal veritabanları üzerinde topolojik sorgularla deformasyon analizleri yorumlanmıştır.

Üzerinde çalışma yapılacak olan Kıbrıs adasının coğrafik özellikleri şu şekildedir: 240 km uzunluğu ve 100 km genişliği ile Akdeniz'in Sicilya ve Sardinya'dan sonra üçüncü büyük adasıdır. Kıbrıs adası 34.33 ve 35.41 Kuzey enlemleri ve 32.23 ve 34.55 Doğu boylamları arasında yer almaktadır. Üzerinde çalışılacak bölge beyaz kutu içinde Şekil 4.1'de gösterilmiştir.

4.2. Ada Uydu Görüntüsünün Vektörel Modellenmesi

Şekil 4.2'de önerilen tekniğin Landsat-7 uydu görüntüsüne uygulanmasının adımlarını göstermektedir. Gürültüleri azaltmak için öncelikle ortanca filtresi ham görüntüye uygulanır. Daha sonra uydu görüntüsünün karakteristik özelliklerini elde etmek için filtrelenmiş RGB renk uzayındaki görüntü HSV ve YCbCr renk uzaylarına dönüştürülür. HSV ve YCbCr bileşen değerlerine göre tüm görüntü (3.9) eşitliği ile verilen testten geçirilir ve (3.10) eşitliği ile testi geçen adayı ifade eden pikseller 255, geçemeyen sulu bölgeleri ifade eden pikseller 0 değeri ile kodlanır (Şekil 4.2b).



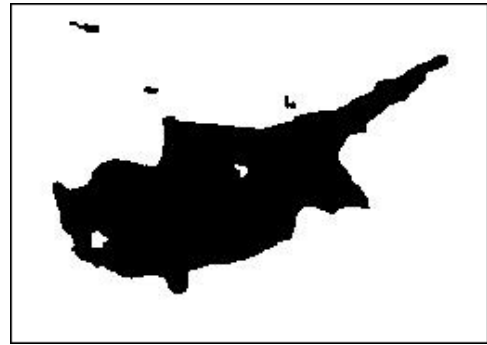
Şekil 4.1. Üzerinde çalışma yapılacak bölge

Segmente edilen görüntü bölge gruplama ve etiketleme işleminden geçirilir. Bu işlemle sulu bölgeler bireysel objeler olarak gruplanır, etiketlenir ve ufak gürültülü kısımlar atılır (Şekil 4.2c). Benzer şekilde adayı temsil eden bölgeler bireysel objeler olarak gruplanır, etiketlenir ve ufak gürültülü kısımlar atılır (Şekil 4.2d). Bu işlemler sonucunda ufak gürültülü kısımlardan soyutlanmış uygun ada ve sulu kısım kalır (Şekil 4.2e).

Bir sonraki adımda sınır pikselleri elde edilir. Sulu kısımlarla komşuluğu olan ada pikselleri Eşitlik (3.11) ile sınır pikselleri olarak belirlenir. Sınır pikselleri 255 (siyah), diğer pikseller ise 0 (beyaz) ile kodlanır. Bu şekilde sınır pikselleri çıkarılır (Şekil 4.2f). Raster formdaki sınır pikselleri 8-yönlü tarama ile vektör formata çevrilip PostgreSQL mekânsal veritabanına kaydedilir.

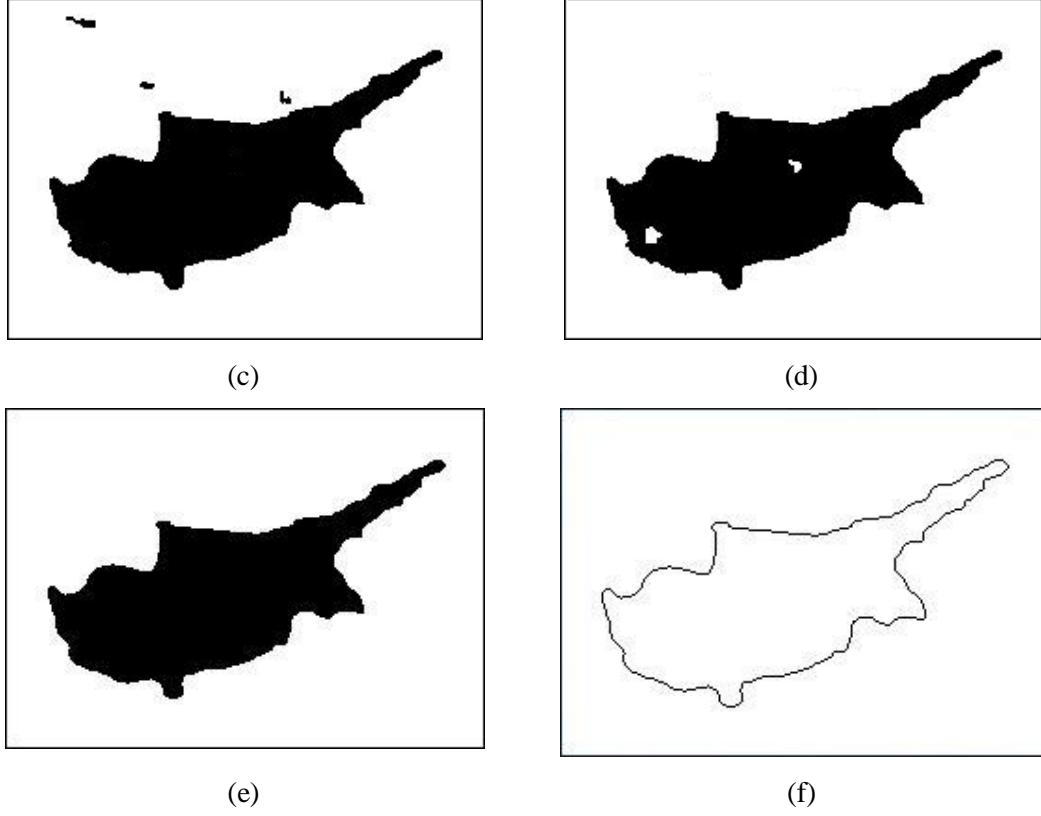


(a)



(b)

Şekil 4.2. Landsat-7 uydu görüntüsünün otomatik vektörizasyonu



Şekil 4.2. (Devam) Landsat-7 uydu görüntüsünün otomatik vektörizasyonu

4.3. Vektör Model Üzerinde Topolojik Sorgular ve Deformasyon Analizleri

Önerilen teknik aracılığıyla vektörleştirilen uydu görüntü üzerinde depremde dolayı kayma derecesi tespiti (Senaryo 1), erozyon (Senaryo 2) ve dolgu (Senaryo 3) gibi bazı senaryolar test edildi. Bu senaryoların sonuçları simülasyon öncesi ve sonrası görüntüler üzerinde çevre ve alan gibi değerlerin değişimleri incelenerek bulundu. Uygulamamızda daha çok alan ve çevre hesapları yaptığımızdan PostGIS fonksiyonları yardımıyla veritabanına kaydedilen ada görüntüsünün alanı ve çevresi hesaplandı. Hesaplanan değerlerden ve verilen harita ölçek değerine göre adanın gerçek alan ve çevre dönüşümü;

$$\text{Gerçek Uzunluk} = \text{Ölçek} \times \text{VT'deki Poligondan Hesaplanan Harita Uzunluğu} \quad (4.1a)$$

$$\text{Gerçek Alan} = \text{Ölçek} (\text{Payda})^2 \times \text{VT'deki Poligondan Hesaplanan Harita Alanı} \quad (4.1b)$$

Eşitlik (4.1) ile bulundu. Veritabanına kaydedilen vektörel yapıdaki adanın alan ve çevre uzunluklarını hesaplayan mekânsal sorgular da Şekil 4.3'teki gibidir:

```

SELECT Round(ST_Area(poligononce)) AS Alan, Round(ST_Area(poligonsonra)) AS
AlanSon,area AS Gercek FROM dbpoligon Where ad= 'kibris'

SELECT Round(ST_length(ST_Boundary(poligononce))) AS CevreOnce,Round(ST_length
(ST_Boundary(poligonsonra))) AS CevreSon, cevre AS Cevre FROM dbpoligon WHERE
ad='kibris'

```

Şekil 4.3. Alan ve çevre hesaplama örnek kodları

Senaryo 1’de Kıbrıs Ada’sı üzerinde öteleme işlemi ile yapay bir deprem varlığı elde edildi. Daha sonra ötelemeden önceki ve sonraki resimlerden elde edilen poligonların ağırlık merkezleri karşılaştırılarak ne kadar km’lik bir kayma olduğuna bakıldı (Şekil 4.4b’deki kırmızı oka bakınız). Kapalı bir poligonun ağırlık merkezi (c_x, c_y) aşağıdaki gibi hesaplanabilir,

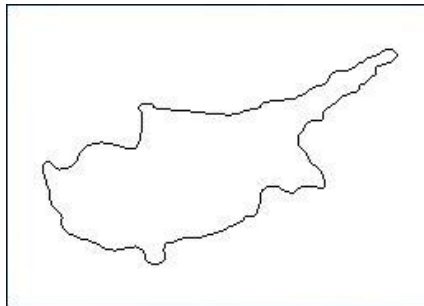
$$c_x = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \quad (4.2)$$

$$c_y = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \quad (4.3)$$

Eşitlik (4.2) ve (4.3)’teki A’lar, poligonun alanını göstermektedir ve şu şekilde hesaplanır,

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i) \quad (4.4)$$

Eşitlik (4.4)’teki N, poligonu oluşturan noktaların sayısıdır ve noktalar (x_i, y_i) şeklinde temsil edilir. Poligonu oluşturan ilk ve son nokta aynı olduğu için burada son nokta göz ardı edilmiştir.



(a)

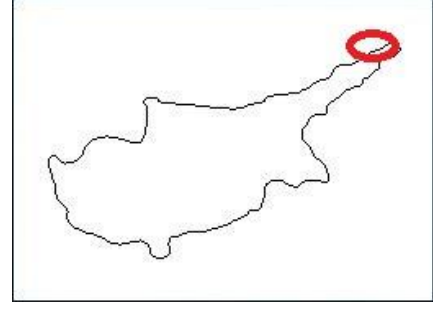


(b)

Şekil 4.4. Uydü görüntüsüne uygulanan bazı senaryolar



(c)



(d)

Şekil 4.4. (Devam) Uydu görüntüsüne uygulanan bazı senaryolar

Senaryo 2’de Kıbrıs Ada’sının bir kısmı silinerek yapay bir erozyon elde edildi. Daha sonra alan ve çevre sorgulamaları yapılarak ne kadar azalma olduğu araştırıldı (Şekil 4.4c’deki kırmızı ile gösterilen yere bakınız). Senaryo 3’te Kıbrıs Ada’sına bir kısım toprak eklenerek yapay bir doldurma/dolgu elde edildi. Daha sonra alan ve çevre sorgulamaları yapılarak ne kadar artma olduğu araştırıldı (Şekil 4.4d’deki kırmızı ile gösterilen yere bakınız).

4.4. Sonuç

Kıbrıs Ada’sının gerçek alanı 9251 km² çevresi ise 648 km’dir. Tablo 4.1’de gösterilen testlerden Senaryo1’de sadece öteleme yapıldığından alan ve çevre değişikliği gözükmemektedir. Kayma miktarı kaymadan önceki ve sonraki ada görüntülerinin ağırlık merkezlerinin karşılaştırılmasıyla 25 km olarak saptandı. Senaryo2’deki yapay erozyondan dolayı alanda 962 km² çevrede 23 km’lik bir azalma, Senaryo3’te ise yapay doldurmadan kaynaklanan 161 km²’lik bir alan ve 13 km’lik bir çevre artması görülmektedir. Senaryo2’de erozyon ve Senaryo3’te doldurma yapıldığından herhangi bir kayma olmamıştır.

Tablo 4.1. Gerçekleştirilen simülasyonların sonuçları

SENARYOLAR	Alan (km ²)		Çevre (km)	
	Önce	Sonra	Önce	Sonra
Öteleme	9251	9251	648	648
Erozyon	9251	8289	648	625
Doldurma	9251	9412	648	663

5. DAĞITIK SİSTEMLERDE GÖRÜNTÜNÜN TAŞINMASI RASTER VE VEKTÖR PERFORMANS ANALİZLERİ

5.1. Giriş

İnternet, CBS kullanıcıları için mekânsal veri (raster veya vektör veri) transfer etmek, analizler gerçekleştirmek ve coğrafik sonuçları göstermek adına önemli bir içerik yayınlama ortamı haline gelmiştir. Son yıllarda, web teknolojilerinin gelişmesiyle coğrafik bilgiye dayalı uygulamalar artmıştır.

Hızla artan mekânsal veri miktarı ile network bant genişliği arasında ters ilişki vardır. Cevap süresinin, istemcinin sunucudan mekânsal veri istemesinden veriyi almasına kadar geçen süre, kısa olması web-CBS uygulamalarında önemlidir. 2000 yıllarında web üzerinden mekânsal veri transferinde kullanıcılar istek yaptıktan sonra veri indirilene kadar hiçbir işlem yapmadan beklemek zorundaydı. Bertolotto tarafından getirilen çözümde kullanıcı veriyi istedikten sonra almasına kadar geçen sürede başka işler de yapabilmektedir [51]. Bu çözüm beraberinde etkili transfer tekniği eksikliğinden mimaride bir takım problemler getirdi.

5.2. Raster ve Vektör Veri İletimi

Şimdiye kadar raster veri(örneğin yüksek çözünürlüklü uydu görüntüsü) iletimiyle ilgili kapsamlı birçok çalışma yapıldı. Bu çalışmalar sıkıştırma [52], wavelet dönüşümü [53], fraktal geometriye dayalı yöntemleri [54,55] ve hibrit sıkıştırma [56]metotlarını içine alır. Ayrıca web üzerinden raster verinin daha az detaylı versiyonundan daha detaylı versiyonuna doğru artırımlı olarak veri transferini sağlayan artırımlı (progressive) veri iletim tekniği geliştirilmiştir [57]. Bu yöntemin avantajları etkin veri transferi, hızlı yanıt süresi ve istenilen detayda veri göndererek disk ve zaman kazanımı şeklindedir.

Mekânsal vektör verisini transfer etmek için de birçok algoritma ve model geliştirilmiştir. Bu algoritmalara üçgenlenmiş düzensiz ağa (ÜDA) dayalı [58,59] , ızgara tabanlı [60]metotlar örnek olarak verilebilir. İlk olarak Bertolotto ve Egenhofer[61] istemci-sunucu mimarisine dayalı artırılmış vektör verisi iletimini ortaya koymuşlardır. Buttenfield[62] artırılmış çözünürlük seviyelerinde geometri ve topolojiyi bozmadan çizgi inceltme metodu ile vektör koordinatlarını iletmiştir. Han ve Tao[63] vektör veri iletmek için istemci-sunucu modeli geliştirmiştir. Yang ve ark. [64] poligonların kesişim noktalarını azaltarak artırılmış poligon verisi transferini gerçekleştirmişlerdir. Ai ve ark. [65] poligonların artırılmış iletimi için bir poligondaki topolojiyi dikkate alan, bir poligon ile çizgi ve başka poligon gibi diğer objeler arasındaki ilişkiyi önemsemeyen değişiklikler birikim modelini (DBM) ortaya koymuşlardır.

5.3. Web Servisleri

Servis Odaklı Mimari (SOA), birçok uygulamanın "servis" denilen kendi modüllerini veya fonksiyonlarını başka uygulamalar tarafından kullanılacak şekilde tasarlandığı bir mimari kavramdır. Web servisleri, SOA altyapısında kullanılan bir yöntemdir. Basitçe uygulamanın özellikleri ya da servisleri bir web sayfası gibi URL adresi olmakta ve diğer uygulama bu web adresini çağırarak uygulamanın özelliklerini kullanabilmektedir. Böylece bir uygulamanın özellikleri başka uygulamaların da kullanacağı şekilde açılmış oluyor. Web servis mantığında; kullanıcı, diğer makineye bir takım veriler gönderir ve geriye yine işlenmiş birtakım verileri alır. Veriyi alan tarafın diğer makinede gerçekleştirilen uygulamalar, kullanılan algoritmalar ve fonksiyonlar hakkında hiçbir bilgisi yoktur.

5.4. Raster ve Vektör Veri İletimi İçin Dağıtık Bir Model

Yukarıda da bahsedildiği üzere artırılmış olarak vektör verisini transfer etmek hala önemli bir problemdir. Bizim çalışmamız, vektör harita verisini transfer etmek için web servis tabanlı bir mimari sunmaktadır. Sunucudan istemciye iletilecek vektör verisinin büyüklüğü sunucu tarafında kenarlaştırma yöntemiyle azaltılarak veri web servis teknolojisi ile iletilmektedir. Landsat-7 Gökçeada uydu görüntüsünün hem

raster hem de vektör hali ileilmek suretiyle yöntemin etkinliği deneysel sonuçlarla gösterilmiştir.

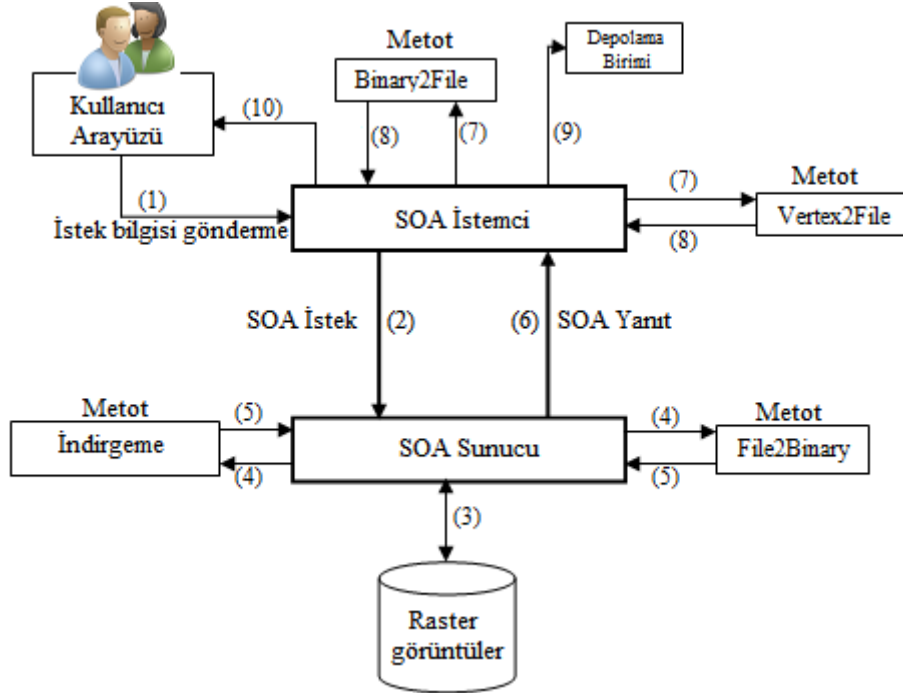
Raster görüntülerde çözünürlük ve renk derinliği arttıkça dosya boyutu artmaktadır. Örneğin 2"x3" büyüklüğünde çözünürlüğü 300 dpi ve renk derinliği 24-bit olan bir görüntünün boyutu 12Mb'in üzerinde olurken bu görüntün vektör formatında boyutu 96-bit kadar olmaktadır. Büyük boyutlu (1Mb'tan fazla) dosyalar internet üzerinden transfer edilirken her iki tarafta da yüksek hızlı internet bağlantısının olmasını gerektirir. Vektör formatında hem işlemciyi yormamış hem de sınırlı bant genişliğinde veri transferini sağlanmış olur. Vektör harita verisinin de internet üzerinden iletim zamanını kısaltmak için veri ilk önce sunucu tarafında vektörleştirme metoduyla basitleştirilip web servisler yardımıyla istemciye gönderilir. İstemci de aldığı veriden tekrar haritayı oluşturabilir.

Oluşturulan mimarinin yapısı şu şekildedir: İletilecek ham ada görüntüsü, resmi daha sonra işleyecek algoritmaların işini kolaylaştırmak için gri seviyeye çevrilmiştir. Görüntüdeki önemli yapısal kısımların muhafaza edilip daha az anlamlı bilgilerin filtrelenmesi yoluyla işlenecek veri sayısını azaltmak sebebiyle kenar bulma tekniklerinden Canny kenar saptama algoritması ile gri görüntü kenar görüntüsüne çevrilmiştir. Görüntü kirlilikleri kenar bulma algoritmalarının başarısını etkilemektedir. Uydu görüntülerinde görüntü kirlilikleri genelde havanın kapalı, bulutlu olmasından veya düşük olasılıkla da olsa farklı cisimlerden ortaya çıkar. Görüntülerin de kenar gibi algılanmaması için seçilen görüntülerin temiz olmasına dikkat ettik.

Uygulamamızda kenar saptama işleminden geçmiş görüntüdeki adayı resmin geri kalanından ayırmak için piksellerin gri seviye değerlerinin tüm imgedeki dağılımlarının yanı sıra yerel özelliklerini de göz önüne alan Otsu eşikleme yöntemi kullanılmıştır. Otsu eşikleme filtresinden geçen resmin adayı temsil eden her bir pikseli 1, geri kalan kısmı 0 olarak kodlanmıştır. İmge üzerinde yataydan başlayarak satır satır soldan sağa tarandığında her satırda ilk rastlanan 1 değerli pikselleri adanın sol sınır değerlerini sağdan sola doğru tarandığında her satırda ilk rastlanan 1 değerli pikselleri de adanın sağ sınır değerlerini verir. Yine imge üzerinde dikey taramalar

yaparak imgenin her sütun için rastlanan yukarıdan aşağıya tarandığında ilk 1 değerli pikselleri adanın üst sınır, aşağıdan yukarıya tarandığında ilk rastlanan 1 değerli pikselleri de alt sınır değerlerini verir [66].

Web servis aracılığıyla vektör harita verisi transferi sağlayan mimarinin (Şekil 5.1) temel iki bileşeni vardır: SOA İstemci ve SOA sunucu. Veri akışı adım numaraları, transfer çizgileri üzerinde parantez içinde belirtilmiştir.



Şekil 5.1. Web servisleri aracılığıyla veri iletimi mimarisinin prototipi

Sistemin genel çalışma prensibi şu şekildedir: Kullanıcı ara yüz aracılığıyla sunucu tarafından almak istediği haritanın türünü (raster veya vektör) seçer. İstek devamlı çalışır ve erişilebilir durumdaki web servise iletilir. Sunucu, istemcinin seçimine göre gerekli işlemi yaptıktan sonra gerekli bilgileri geri istemciye gönderir. Sistemin detaylı açıklaması:

Adım1: Kullanıcı ara yüz aracılığıyla sunucu tarafından almak istediği haritanın türünü (raster veya vektör) seçer.

Adım2: SOA İstek olarak isimlendirilmiş adımdır. Kullanıcının ara yüz aracılığıyla girdiği veri web servis sunucuna gönderilir.

Adım3: Sunucu üzerinde işlem yapılacak görüntüye ulaşır.

Adım4, 5: Kullanıcı raster görüntü isteğinde bulunmuşsa ilgili resim “File2Binary” metoduyla ikili diziye çevrilir, vektör görüntü isteğinde bulunmuşsa yukarıda da bahsedilen indirgeme yöntemiyle basitleştirilir.

Adım6: SOA İstek olarak isimlendirilmiş adımdır. İstenilen görüntünün ikili dizisi veya vertex dizisi istemciye gönderilir.

Adım7, 8: Kullanıcı raster görüntü istemişse ikili diziden raster, vektör görüntü istemişse nokta dizisinden vektör görüntü oluşturulur.

Adım9: Oluşturulan görüntüler belirlenmiş bir dizine resim dosyası olarak kaydedilir.

Adım10: Görüntü ara yüz ekranında gösterilir.

Raster ve vektör harita verisinin performansını ölçmek için yapılan testler ağ bant genişliği 10Mbps olan bir intranette gerçekleştirildi. Testler için web servislerle sunucu-istemci mimarisi (RPC modeli) kuruldu. Farklı büyüklükteki veriler üzerinde iletim ve yanıt sürelerini analiz ettik. Web servisleri oluşturmak için Axis kullanıldı [67]. Uygulama; sunucu tarafı 2.3 GHz işlemci ve 4 GB belleğe sahip Tomcat 6.0 sunucusu, istemci tarafı ise 2.4 GHz işlemci ve 4 GB belleğe sahip bir web istemci üzerinde yapıldı.

İletim zamanı, ikili veya nokta dizisinin sunucudan istemciye gönderilme zamanıdır. Yanıt zamanı ise kullanıcının istediği veri türünü seçmesinden sonucun sunucu tarafından hesaplanıp istemcinin veriyi almasına kadar geçen süredir. Yanıt süresi Eşitlik (5.1)'deki gibi;

$$T_Y(\text{Yanıt})=t_0(\text{istek}) + t_1(\text{file2Binary}) + t_2(\text{iletim}) \quad (5.1)$$

hesaplanır. $t_0(\text{istek})$ zamanı kullanıcı isteğinin istemciden sunucuya iletilmesine, $t_1(\text{file2Binary})$ raster verinin ikili dizisine dönüştürülmesine kadar geçen süredir. $t_2(\text{iletim})$ yukarıda açıklandı. $T_R(\text{yanıt})$ ise indirgeme (vektörleştirme) olmadan geçen yanıt süresidir. Sunucu tarafında vektörleştirme olursa yanıt süresi $T_R'(\text{yanıt})$ Eşitlik (5.2)'deki gibi;

$$T_R'(\text{yanıt})=t_0(\text{istek}) + t_v(\text{vektörleştirme}) + t_1(\text{iletim}) \quad (5.2)$$

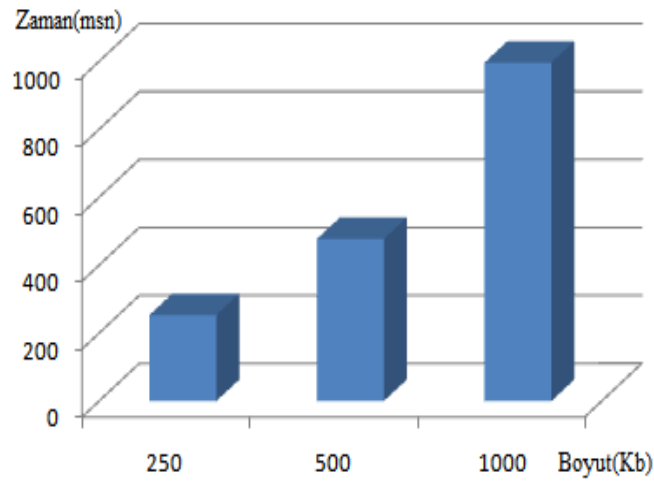
hesaplanır. t_v (vektörleştirme) sunucu tarafında vektörleştirme (indirgeme) için geçen süredir.

5.5. Önerilen Dağıtık Model ile Raster ve Vektör Veri Taşınması Performans Analizleri

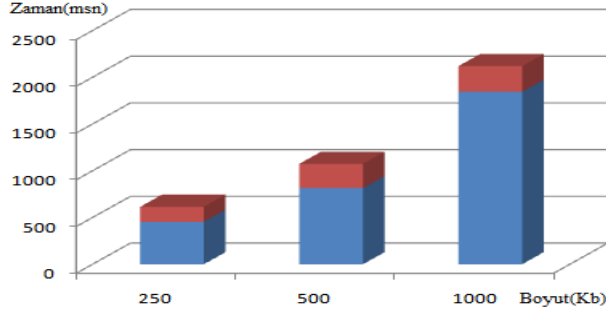
Landsat-7 Gökçeada uydu görüntüsü için vektörleştirme ve vektörleştirme olmadan ki durumlarına göre yanıt süreleri karşılaştırıldı. Ayrıca Gökçeada uydu görüntüsü parçalara ayrılarak farklı büyüklükler (Şekil 5.2) için sistemin ölçeklenebilirliği test edildi.



Şekil 5.2. Gökçeada uydu görüntüsünün parçaları

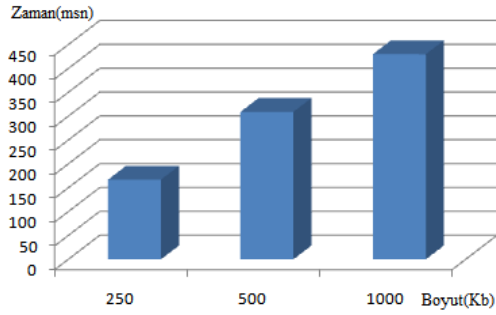


Şekil 5.3. Vektörleştirme işlemi olmadan gerekli yanıt süreleri



Şekil 5.4. Vektörleştirme işlemi yapılırca gerekli yanıt süreleri

Şekil 5.3’de 1000Kb boyutlu Gökçeada uydu görüntüsü ve boyutları 245-525Kb arasında değişen parçaları için vektörizasyon işlemi olmadan gerekli yanıt sürelerini göstermektedir. Şekil 5.4 ise vektörleştirme işlemi yapıldığında gerekli yanıt sürelerini göstermektedir. Vektörleştirme işlemi algoritmanın karmaşıklığından dolayı çok zaman aldığı için vektörleştirme işlemiyle iletim için de çok zaman alıyor. Şekil 5.4’teki mavi kısımlar vektörleştirme için gerekli süreyi göstermektedir. Görüldüğü üzere 1000Kb büyüklüğündeki Gökçeada uydu görüntüsünün yanıt süresinin yaklaşık $\frac{3}{4}$ ’ü (1745 msn) kadardır.

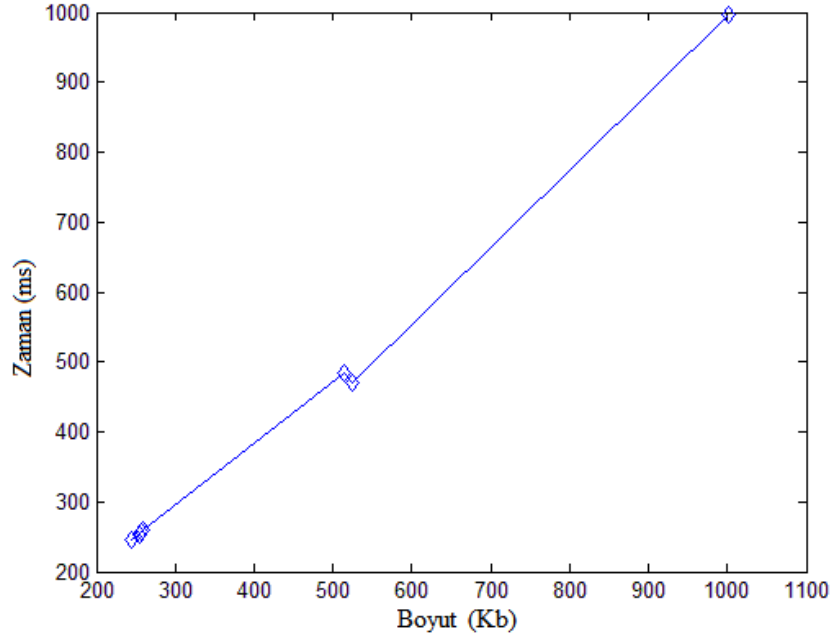


Şekil 5.5. Nokta iletimi için gerekli yanıt süresi

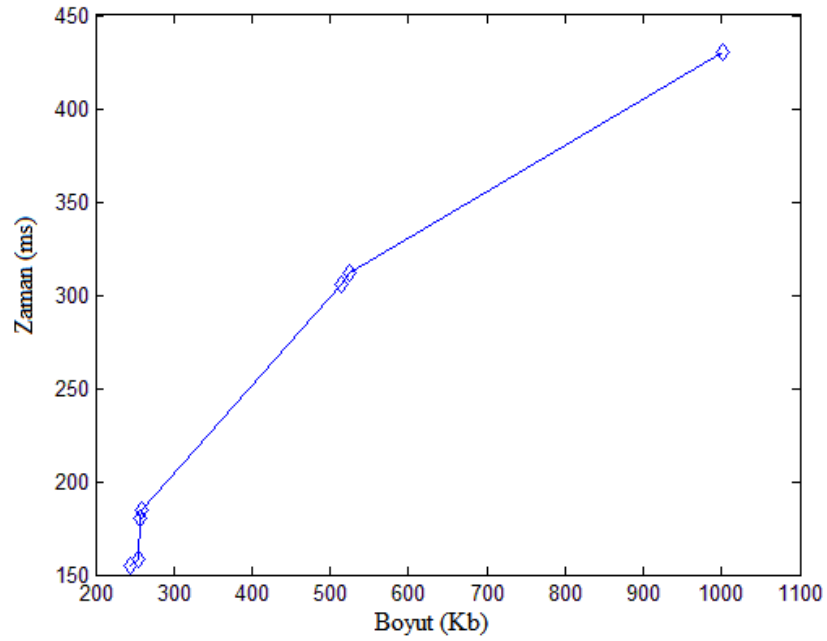
Vektörleştirme işlemi koşum anında çok zaman aldığı için, raster görüntünün vektörleştirme işleminin koşum zamanından önce yapılıp nokta dizilerinin veritabanı veya yerel diskte hazır olduğunu kabul edersek, nokta dizisinin iletimi için gerekli yanıt süresi Şekil 5.5’de verilmiştir. Şekil 5.5 nokta dizisinin iletimi için gerekli yanıt süresinin 1Mb büyüklüğündeki raster veri iletiminden dört kat daha hızlı olduğunu göstermektedir.

Kullanıcı ara yüz aracılığıyla sunucu tarafından vektör formatta bir harita isteyip sunucu tarafında vektörizasyon işlemi yapıldıktan sonraki veya vertex iletiminden

sonraki haritanın kullanıcı tarafında oluşturulması için geçene süreler Şekil 5.6 ve Şekil 5.7’de gösterilmiştir.



Şekil 5.6. Vektörizasyon sonrası harita oluşumu için geçen süre



Şekil 5.7. Vertex iletimi sonrası harita oluşumu için geçen süre

5.6. Sonuç

Yukarıda belirtilen bulgular hızla artan mekânsal veri miktarı ile network bant genişliği arasında ters ilişki olduğunu göstermektedir. Cevap süresinin, istemcinin sunucudan mekânsal veri istemesinden veriyi almasına kadar geçen sürenin kısa olması web-CBS uygulamalarında önemli bir unsurdur. Bu çalışmada, vektör harita verisini transfer etmek için web servis tabanlı bir mimari sunulmuştur. Sunucudan istemciye iletilecek vektör verisinin büyüklüğü sunucu tarafında kenarlaştırma yöntemiyle azaltılarak veri web servis teknolojisi ile iletilmektedir. Yapılan testlerin sonuçları orijinal raster verisinin büyüklüğünün vektörleştirme işlemiyle azaldığını ve sistemin ölçeklenebilir olduğunu; fakat vektörleştirme algoritmasının çok zaman alıcı olmasından dolayı vektörleştirme işleminin çalışma zamanında yapılması iletim hızını azalttığını göstermiştir. Fakat daha sonra iletilecek uydu görüntüsünün vektör hali yerel diskte tutulduğu kabul edildiğinde vektör verinin iletim işleminin raster veri iletiminden daha hızlı olduğu tespit edilmiştir.

6. SONUÇLAR VE ÖNERİLER

Uydu görüntüleri üzerinde yapılan zaman-mekânsal ya da topolojik analizlerde (komşuluk, kesişme vb.) piksel bazlı bilgidен ziyade nokta setleriyle tanımlanan şekil koordinat bilgilerini kullanmak hızlı ve pratik olması yönünden daha çok tercih edilmektedir. Bu çalışmada Landsat uydusu görüntüleri öncelikle poligon nesnesi olarak modellenerek bilgisayarların ve bilgi sistemlerinin kolayca anlayabileceği ve işleyebileceği formata getirilmiştir. Adanın poligon temsili daha sonra nesne-ilişkisel veritabanlarında depolanabilir ve işlenebilir hale getirilmiştir. Nesne-ilişkisel veritabanlarından uzamsal (mekânsal) veritabanları; nokta, çizgi ve poligon gibi kompleks geometri objeleri ile tanımlanabilen mekânsal veriler için hızlı depolama ve sorgu avantajlarına sahiptir. Bu şekilde uydu görüntülerinin zaman-mekânsal ve topolojik analizlerinde, hızla gelişen nesne-ilişkisel veritabanlarındaki gelişmelerden faydalanılmıştır. Sistemin etkinliği ve kullanılabilirliği; deprem, toprak kayması, dolgu gibi çeşitli senaryo durumları ile test edilmiştir. Senaryolar, doğal olayların belirlenen ada üzerindeki etkisinin uydu görüntüleri aracılığıyla ölçülmesi temeline dayanmaktadır.

Tez çalışmasının ana katkıları aşağıdaki gibi maddeler halinde sıralanabilir;

- Ada nesnesinin renk uzay dönüşümü ile segmentasyonu ve bölge gruplama ve etiketlemeye dayalı yöntemlerle vektörel olarak modellenmesi yapılmıştır.
- Nesne-ilişkisel mekânsal veritabanının kurulması ve vektörel modelin veritabanına kaydedilmesi gerçekleştirilmiştir.
- Zaman-mekânsal ve topolojik sorguların ada üzerinde test edilmiştir.
- Doğal olayların (dolgu, aşınma ve deprem vb.) sistem üzerinde simülasyonun gerçekleştirilerek sistemin etkinliğinin ve doğruluğunun test edilmesi sağlanmıştır.

- Dağıtık sistemler üzerinde raster ve vektör formatındaki görüntünün sistem performansına etkisi araştırılmıştır.
- Nesne-ilişkisel veritabanlarının kullanımının özendirilmesi ve yaygınlaştırılması ve hatta bu alanda uydu görüntülerinin de kullanılabilirliğinin ispatlanması sağlanmıştır. Nesne-ilişkisel veri tabanlarındaki hızlı gelişmelerden “uzaktan algılama” ve “örüntü tanıma” alanlarındaki çalışmaların da faydalanabileceği ve bunun getirisinin neler olabileceği gerçek verilerle (ada uydu görüntüleri) yapılan çalışmalarla ortaya konulmuştur.
- Önerilen yöntem tamamen açık kaynak kodlu olduğundan kullanılması ve geliştirilmesi açısından büyük bir avantaj sağlamıştır. Yöntemin açık kaynak kodlu olması sıfır maliyet yanında, kullanıcıya yazılımı değiştirme özgürlüğü sağlamaktadır. Kullanıcılar kodu rahatça kullanabilir, gereksinimlerine göre basitleştirebilir, sağlamlaştırabilir ve güçlendirebilirler.

Bu çalışmanın yanı sıra ileride gerçekleştirilmesi planlanan bazı yenilikler aşağıda sunulmaktadır;

- Doğal olaylar ya da insan etmeni nedeniyle yerküre sürekli şekil değiştirmektedir. Doğal olaylara örnek olarak; iklim değişikliği nedeniyle kutuplardaki buzulların erimesi ve su seviyesindeki yükselmeler, deprem, sel, toprak kayması (erozyon) vb. sayılabilir. İnsan etmenine örnek olarak da denize doğru dolgular, barajlar ve suni adalar ya da göllerin oluşturulması sayılabilir. Bunlara ek olarak bazen su seviyesinde azalmadan dolayı yeni adaların ortaya çıkması ya da deniz ve okyanuslardaki sönmüş yanardağların tekrar harekete geçerek yeni adaların oluşması da sayılabilir. Bu liste daha da uzatılabilir. Ancak, bizim bu çalışmada ortaya koyduğumuz yöntem, bu tür değişimlerin analizi şiddetinin ölçülmesi ve hatta önceden tahmini ya da erken uyarı sisteminin geliştirilebilmesi sağlanabilir.

- Tez kapsamında yapılan arařtırmaların sonuçları ve bulgular, bařta coęrafi bilgi sistemleri olmak üzere tıp, astronomi, jeofizik, jeoloji ve harita gibi birok alana uygulama ve teori seviyesinde katkı saęlayabilir. Geliřtirilecek kompleks sınır ıkarım teknikleri dięer disiplinlerde de uygulanabilir. Tıpta beyin grntlerinin analizi ya da gz grntlerinden glokom testlerinin yapılması uygulamaları rnek olarak verilebilir. Coęrafi Bilgi Sistemleri alanında ise (Jeofizik, Jeoloji ve Harita) veri zerinde zaman-meknsal ve topolojik inceleme/analizlerde veriye ulařma ve bilgi paylařmada yaygınlık ve hız kazandırılabilir.

KAYNAKLAR

- [1] Tombre K., Analysis of Engineering Drawings: State of the Art and Challenges, Proceedings of the Graphics Recognition-Algorithms and Systems, Nancy, France, 1997.
- [2] Guru D. S., Shekar B. H., Nagabhushan P., A simple and robust line detection algorithm based on small eigenvalue analysis, *Pattern Recognition Letters*, 2004, **25**, 1-13.
- [3] Li H., Lavin M. A., Le Master R. J., Fast Hough transform: A hierarchical approach, *Computer Vision, Graphics, and Image Processing*, 1986, **36**, 139-161.
- [4] Illingworth J., Kittler J., The adaptive Hough transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987, **9**, 690-698.
- [5] Ben-Tzvi D., Sandler, M. B., A combinatorial Hough transform, *Pattern Recognition Letters*, 1990, **11**, 167-174.
- [6] Princen J., Illingworth J., Kittler J., A hierarchical approach to line extraction based on the Hough transform, *Computer Vision, Graphics, and Image Processing*, 1990, **52**, 57-77.
- [7] Atiquzzaman M., Multiresolution Hough transform-an efficient method of detecting patterns in images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, **14**, 1090-1095.
- [8] Naccache N. J., Shinghal R., An investigation into the skeletonization approach of Hilditch, *Pattern Recognition*, 1984, **17**, 279-284.
- [9] Naccache N. J., Shinghal R., SPTA: a proposed algorithm for thinning ikili patterns, *IEEE Transactions on Systems, Man, and Cybernetics*, 1984, **14**, 409-418.
- [10] Deutsch E. S., Thinning algorithms on rectangular, hexagonal and triangular arrays, *Communications of the ACM*, 1972, **15**, 827-837.

- [11] O’Gorman L., $k \times k$ thinning, *Computer Vision, Graphics, and Image Processing*, 1990, **51**, 195-215.
- [12] Rosenfeld A., Pfaltz J. L., Sequential operations in digital picture processing, *Journal of the ACM*, 1966, **13**, 471-494.
- [13] Rosenfeld A., Pfaltz J. L., Distance functions in digital pictures, *Pattern Recognition*, 1968, **1**, 33-61.
- [14] Davies E. R., Plummer A. P. N., Thinning algorithms: a critique and a new methodology, *Pattern Recognition*, 1981, **14**, 53-63.
- [15] Lee S., Lam L., Suen C. Y., Performance Evaluation of Skeletonization Algorithms for Document Image Processing, *Proceedings of the 1st International Conference on Document Analysis and Recognition*, France, 1991.
- [16] Lam L., Suen C.Y., Evaluation of Thinning Algorithms from an OCR Viewpoint, *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, Tsukuba Science City, 1993.
- [17] Jaisimha M. Y., Haralick R. M., Dori D., A Methodology for the Characterization of the Performance of Thinning Algorithms, *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, Tsukuba Science City, 1993.
- [18] Cordella L.P., Marcelli A., An alternative Approach to the Performance Evaluation of Thinning Algorithms for Document Processing Applications, *Lecture Notes in Computer Science*, 1996, **1072**, 13-22.
- [19] Martinez-Perez M.P., Jimenez J., Navalon J.L., A thinning algorithm based on contours, *Computer Vision, Graphics, and Image Processing*, 1987, **39**, 186-201.
- [20] Song J., Su F., Li H., Cai S., Raster to Vector Conversion of Construction Engineering Drawings, *Automation in Construction*, 2002, **11**, 597-605.
- [21] Lin X., Shimotsuji S., Minoh M., Sakai T., Efficient Diagram Understanding with Characteristic Pattern Detection, *Computer Vision, Graphics and Image Processing*, 1985, **30**, 84-106.
- [22] Dori D., Liu W., Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, **21**, 202-215.

- [23] Zhang X., Zheng G., Fu H., Cao L., Research On Image Vectorization Based on Mathematical Morphology, *International Conference on Mechatronic Science, Electric Engineering and Computer*, Jilin, 2011.
- [24] Mohamed N., Melki N., Atef H., Line extraction algorithm based on image vectorization, *Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation*, Xi'an, 2010.
- [25] Tam R., Heidrich W., Feature-preserving medial axis noise removal, *Lecture Notes Computer Science*, 2002, **2351**, 672-686.
- [26] Aniket B., Fast Vectorization and Upscaling Images with Natural Objects using Canny Edge Detection, *3rd International Conference on Electronics Computer Technology (ICECT)*, Kanyakumari, 2011.
- [27] Lee K., Cho S., Choy Y., Automated vectorization of cartographic maps by a knowledge-based system, *Engineering Applications of Artificial Intelligence*, 2000, **13**, 165-178.
- [28] Kirsanov A., Vavilin A., Jo K-H., Contour-based algorithm for vectorization of satellite images, *IFOST 2010 Proceedings*, Ulsan, 2010.
- [29] Dunfey R. I., Gittings B. M., Batcheller J. K., Towards an open architecture for vector GIS, *Computers & Geosciences*, 2006, **32**, 1720-1732.
- [30] Ballagh L. M., Raup B. H., Duerr R. E., Khalsa S. J. S., Helm C., Fowler D., Gupte A., Representing scientific datasets in KML: Methods and challenges, *Computers & Geosciences*, 2011, **37**, 57-64.
- [31] Ahmed Z., Mounir S., Farhat F., Edge Segmentation of Satellite Image using Phase Congruency Model, *International Journal of Signal Processing*, 2010, **6**, 39-44.
- [32] Marina P. C., Claudio A. D., Gerardo M. E. P., Piccolo M. C., Superresolution border segmentation and measurement in remote sensing images, *Computers and Geosciences*, 2012, **40**, 87-96.
- [33] Jubai A., Jing B., Yang J., Combining fuzzy theory and a algorithm for satellite image edge detection, *International Journal of Remote Sensing*, 2006, **27**, 3013-3024.

- [34] Shah V., Choudhary A., Tewari K., River extraction from satellite image, *International Journal of Computer Science Issues*, 2011, **8**, 386-391.
- [35] Jodouin S., Bentabet L., Ziou D., Vaillancourt J., Armenakis C., Spatial database updating using active contours for multispectral images: application with Landsat 7, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2003, **57**, 346-355.
- [36] Mangala T. R., Bhirud S. G., An Effective ANN-Based Classification System for Rural Road Extraction in Satellite Imagery, *European Journal of Scientific Research*, 2010, **47**, 574-585.
- [37] Ioannis P., Anastasios N. V., *Nonlinear Digital Filters: Principles and Applications*, 1st ed., Springer, 1990.
- [38] Lee J. S., Speckle suppression and analysis for Synthetic Aperture Radar images, *Optical Engineering*, 1986, **25**, 636-643.
- [39] Wiener N., *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, The MIT Press, 1964.
- [40] Frost V. S., Stiles J. A., Shanmugan K. S., Holtzman J. C., A model for radar images and its application to adaptive digital filtering of multiplicative noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1982, **4**, 157-166.
- [41] Kuan D. T., Sawchuck A. A., Strand T. C., Chavel P., Adaptive noise smoothing filter for images with signal-dependent noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985, **7**, 165-177.
- [42] Perona P., Malik J., Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, **12**, 629-639.
- [43] Thomas L., Ralph W. K., *Remote Sensing and Image Interpretation*, 4th ed., Wiley, 1999.
- [44] Kraus K., *Photogrammetrie Band 1*, 7th ed., Walter de Gruyter; 2004.
- [45] Aplin P., Atkinson P., Curran P., Fine Spatial Resolution Satellite Sensors for the Next Decade, *International Journal of Remote Sensing*, 1997, **18**, 3873-3881.

- [46] Başpehlivan C., *Uydu Teknolojileri ve Haritacılık*, Piri Reis Harita ve CBS Çalışanları Yardımlaşma Derneği, 2004.
- [47] Ayşe Y.E., *Uydu Görüntüleri Rehber Kitapçığı*, NİK Sistem, 2005.
- [48] <http://elogeo.nottingham.ac.uk>, GIS for Educators - Topic 2: Vector Data, (Ziyaret Tarihi: 15 Şubat 2012).
- [49] Schalkoff R. J., *Digital Image Processing and Computer Vision*, 1st ed., Wiley, 137-145, 1989.
- [50] Bong D. B. L., Lai K. C., Joseph A., Automatic Road Network Recognition and Extraction for Urban Planning, *Proceedings of World Academy of Science, Engineering and Technology*, 2009.
- [51] Bertolotto M., Egenhofer M. J., Progressive vector transmission, *Proceedings of the 7th ACM international symposium on Advances in geographic information systems*, Kansas City, 1999.
- [52] Gonzales R. C., Woods R. E., *Digital Image Processing*, 2nd ed., Prentice Hall, 2002.
- [53] Martin M. B., Bell A. E., New image compression techniques using multiwavelets and multiwavelet packets, *IEEE Transactions on Image Processing*, 2001, **10**, 500-510.
- [54] Barnsley M. F., *Fractals Everywhere*, 2nd ed., Morgan Kaufmann Pub., 1989.
- [55] Woon W. M., Shuen Ho A. T., Yu T., Tam S. C., Tan S. C., Yap L. T., Achieving high data compression of self-similar satellite images using fractal, *Proc. Of IEEE International Symposium on Geoscience and Remote Sensing*, Honolulu, 2000.
- [56] Yu T., Shuen Ho A. T., Tam S. C., A Novel Hybrid Bi-orthogonal Wavelets/ADPCM Algorithm for Very Low Bit Rate Satellite Image Compression, *Proc. Of IEEE International Symposium on Geoscience and Remote Sensing*, Hamburg, 1999.
- [57] Rauschenbach U., Schumann H., Demand-driven image transmission with levels of details and regions of interest, *Computer and Graphics*, 1999, **23**, 857-866.

- [58] Magillo L. P., Puppo E., Compressing triangulated irregular networks, *GeoInformatica*, 2000, **4**, 67-88.
- [59] Park D., Cho H., Kim Y., A TIN compression method using Delaunay triangulation, *International Journal of Geographical Information Science*, 2001, **15**, 255-270.
- [60] Wessel P., Compression of large data grids for internet transmission, *Computers & Geosciences*, 2003, **29**, 665-671.
- [61] Bertolotto M., Egenhofer M. J. , Progressive transmission of vector map data over the World Wide Web, *GeoInformatica*, 2001, **5**, 345-373.
- [62] Buttenfield B. P., Transmitting vector geospatial data across the Internet, *Lecture Notes in Computer Science*, 2002, **2478**, 51-64.
- [63] Han H., Tao V., Wu H., Progressive vector data transmission, *Proceedings of the 6th AGILE*, Lyon, France, 2003.
- [64] Yang B. S., Purves R. S., Weibel R., Implementation of progressive transmission algorithms for vector map data in web-based visualization, *XXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Istanbul, Turkey, 2004.
- [65] Ai T., Li Z., Liu Y., Progressive transmission of vector data based on changes accumulation model, *11th International Symposium on Spatial Data Handling*, Leicester, UK, 2004.
- [66] Eken S., Sayar A., Vector Modelling of Island Satallite Images for Spatial Databases, *Proc. Of International Science and Technology Conference (ISTEC 11)*, Istanbul, Turkey, 2011.
- [67] <http://ws.apache.org/axis> (Ziyaret Tarihi: 20 Mart 2012).

EKLER

Ek-A GELİŞTİRİLEN SİSTEM İÇİN YAZILAN KAYNAK KODLAR

SINIFLARIN ÖZETLERİ	
Class MedianFilter	3x3'lük ortanca işlemini gerçekleyen filtre sınıfı
Class ColorSpaceConversion	RGB renk uzayından HSV ve YCbCr renk uzaylarına dönüşüm yapan sınıf
Class Segmentation	HSV ve YCbCr değerlerine göre segmentasyon işlemini gerçekleyen sınıf
Class RegionGroupingLabeling	Görüntüdeki objeleri şekil ve büyüklüklerine göre karakterize edip ve her birini numarandıran sınıf
Class Closing	İkili görüntü üzerinde morfolojik kapanma işlemi gerçekleyen sınıf
Class TracingBorders	Sınır piksellerini takip ederek poligon yapısı oluşturan sınıf
Class StoringSDB	Vektörel bilgiyi veritabanına kaydeden sınıf
Class QueryingSDB	Mekânsal veritabanı üzerinde sorgulama yapmayı sağlayan sınıf

Class MedianFilter
<pre>public int[] medianfilterPixels(int width, int height, int[] inPixels) Yüksekliği, genişliği ve RGB piksel değerleri arguman olarak verilen görüntü üzerinde ortanca filtreleme işlemini yapar ve geriye filtrelenmiş RGB değerlerini döndürür. int index = 0; int[] argb = new int[9]; int[] r = new int[9]; int[] g = new int[9]; int[] b = new int[9]; int[] outPixels = new int[width * height]; for (int y = 0; y < height; y++) { for (int x = 0; x < width; x++) { int k = 0; for (int dy = -1; dy <= 1; dy++) { int iy = y+dy; if (0 <= iy && iy < height) { int ioffset = iy*width; for (int dx = -1; dx <= 1; dx++) { int ix = x+dx; if (0 <= ix && ix < width) { int rgb = inPixels[ioffset+ix]; argb[k] = rgb; r[k] = (rgb >> 16) & 0xff;</pre>

```

g[k] = (rgb >> 8) & 0xff;
b[k] = rgb & 0xff;
k++;
}
}
}
}
while (k < 9) {
argb[k] = 0xff000000;
r[k] = g[k] = b[k] = 0;
k++;
}
outPixels[index++] = argb[rgbMedian(r, g, b)];
}
}
return outPixels;
}

int rgbMedian(int[] r, int[] g, int[] b) {
int sum, index = 0, min = 2500;

for (int i = 0; i < 9; i++) {
sum = 0;
for (int j = 0; j < 9; j++) {
sum += Math.abs(r[i]-r[j]);
sum += Math.abs(g[i]-g[j]);
sum += Math.abs(b[i]-b[j]);
}
if (sum < min) {
min = sum;
index = i;
}
}
return index;
}

```

Class ColorSpaceConversion

```
public void yhsv_Conversion(int R1,int G1,int B1)
```

Arguman olarak verilen RGB değerlerinden H, S, V ve Y değerlerini hesaplar.

```

float Y;
float H;
float S;
float V;
float R = (float)(R1 / 255.0);
float G = (float)(G1 / 255.0);
float B = (float)(B1 / 255.0);
Y = (float)(65.481*R + 128.553*G + 24.966*B + 16.0);
V = (float)((R + G + B) / 3);

```

```

S = (float)(1 - (3 / (R + G + B)) * min(R, G, B));
float fipay = (float)(0.5 * (2 * R - G - B));
float fipayda = (float)(Math.sqrt(Math.pow((R - G), 2) + (R - B) * (G - B)));
float fi = (float)(1.0 / Math.cos(fipay / fipayda));
if (B <= G)
    H = fi;
else H = 360 - fi;

float min(float R, float G, float B)
{
    float min = 5;
    if (R < min)
        min = R;
    if (G < min)
        min = G;
    if (B < min)
        min = B;
    return min;
}

```

Class Segmentation

public **BufferedImageCondition**(**BufferedImage** image)

Ada olmayan objelerin eliminasyonunu gerçekler.

```

    BufferedImage new_bitmap;
new_bitmap = (BufferedImage)
createImage(image.getWidth(),image.getHeight());
    for (int i = 0 ; i<image.getWidth(); i++)
{
    for (int j = 0; j <image.getHeight(); j++)
    {
        if ((resim_HSV[i][j].H <0.1)&&
(resim_HSV[i][j].Y<80)&&((resim_HSV[i][j].V < 0.2) || (resim_HSV[i][j].V>0.3))
        {
            Color color = Color.BLACK;
            int x = color.getRGB();
new_bitmap.setRGB(i, j, x);
        }
        else
        {
            Color color = Color.WHITE;
            int x = color.getRGB();
new_bitmap.setRGB(i, j, x);
        }
    }
}
}
}
}

```

Class Closing

```
public int[] erodeFilterPixels( int width, int height, int[] inPixels)
```

İkili görüntü üzerinde morfolojik erozyon işlemi gerçekler.

```
    int[] outPixels = new int[width * height];
    for (int i = 0; i < iterations; i++) {
        int index = 0;

        if (i > 0) {
            int[] t = inPixels;
            inPixels = outPixels;
            outPixels = t;
        }
        for (int y = 0; y < height; y++) {
            for (int x = 0; x < width; x++) {
                int pixel = inPixels[y*width+x];
                if (blackFunction.isBlack(pixel)) {
                    int neighbours = 0;

                    for (int dy = -1; dy <= 1; dy++) {
                        int iy = y+dy;
                        int ioffset;
                        if (0 <= iy && iy < height) {
                            ioffset = iy*width;
                            for (int dx = -1; dx <= 1; dx++) {
                                int ix = x+dx;
                                if (!(dy == 0 && dx == 0) && 0 <= ix && ix < width) {
                                    int rgb = inPixels[ioffset+ix];
                                    if (!blackFunction.isBlack(rgb))
                                        neighbours++;
                                }
                            }
                        }
                    }

                    if (neighbours >= threshold) {
                        if (colormap != null)
                            pixel = colormap.getColor((float)i/iterations);
                        else
                            pixel = new Color();
                    }
                    outPixels[index++] = pixel;
                }
            }
        }

        return outPixels;
    }
}
```



```
public int[] dilateFilterPixels( int width, int height, int[] inPixels)
İkili görüntü üzerinde morfolojik genişleme (dilation) işlemi gerçekler.
```

```
    int[] outPixels = new int[width * height];
    for (int i = 0; i < iterations; i++) {
        int index = 0;

        if (i > 0) {
            int[] t = inPixels;
            inPixels = outPixels;
            outPixels = t;
        }
        for (int y = 0; y < height; y++) {
            for (int x = 0; x < width; x++) {
                int pixel = inPixels[y*width+x];
                if (!blackFunction.isBlack(pixel)) {
                    int neighbours = 0;

                    for (int dy = -1; dy <= 1; dy++) {
                        int iy = y+dy;
                        int ioffset;
                        if (0 <= iy && iy < height) {
                            ioffset = iy*width;
                            for (int dx = -1; dx <= 1; dx++) {
                                int ix = x+dx;
                                if (!(dy == 0 && dx == 0) && 0 <= ix && ix < width) {
                                    int rgb = inPixels[ioffset+ix];
                                    if (blackFunction.isBlack(rgb))
                                        neighbours++;
                                }
                            }
                        }
                    }

                    if (neighbours >= threshold) {
                        if (colormap != null)
                            pixel = colormap.getColor((float)i/iterations);
                        else
                            pixel = newColor;
                    }
                }
                outPixels[index++] = pixel;
            }
        }
        return outPixels;
    }
}
```

Class TracingBorders

```

public void vectorize (BufferedImageResim)

    boolean bitir = true;
    boolean ilk_bayrak = true;
    for (int i = 1 ; i< Resim.getWidth()-1; i++)
    {
        for (int j = 1; j <Resim.getHeight()-1; j++)
        {
            Color cl = new Color(Resim.getRGB(i, j));
            if (cl.getRed() == 0)
            {
                if (ilk_bayrak == true)
                {
                    Point p = new Point(i, j);
                    bas = p;
                    _point.add(p);
                    _basx = i;
                    _basy = j;
                    _alx.add(i);
                    _aly.add(j);
                    ilk_bayrak = false;
                    komsuluk(i,j);
                }
                bitir = false;
            }
        }
        if (bitir == false)
        break;
    }
    if (bitir == false)
    break;
}

void komsuluk(int x,int y)
{
    int k=0;
k++;
    int sayac = 0;

    Color cl = new Color(Resim.getRGB(x + 1, y));
    if (cl.getRed() == 0)
    {
        Point p1 = new Point(x + 1, y);
        if (_point.contains(p1))
        {
            if (k > 5)
                if ((x + 1 == _basx) && (y == _basy))
                    return;
        }
    }
else

```

```

{
    _alx.add(x+1);
    _aly.add(y);
    _point.add(p1);
    komsuluk(x+1, y);
}
}

cl = new Color(Resim.getRGB(x, y + 1));
if (cl.getRed() == 0)
{
    Point p1 = new Point(x, y + 1);
    if (_point.contains(p1))
    {
        if (k > 5)
            if ((x == _basx) && (y + 1 == _basy))
                return;
    }
    else
    {
        _alx.add(x);
        _aly.add(y+1);
        _point.add(p1);
        komsuluk(x, y+1);
    }
}

cl = new Color(Resim.getRGB(x + 1, y + 1));
if (cl.getRed() == 0)
{
    Point p1 = new Point(x + 1, y + 1);
    if (_point.contains(p1))
    {
        if (k > 5)
            if ((x + 1 == _basx) && (y + 1 == _basy))
                return;
    }
    else
    {
        _alx.add(x + 1);
        _aly.add(y + 1);
        _point.add(p1);
        komsuluk(x + 1, y + 1);
    }
}

cl = new Color(Resim.getRGB(x - 1, y));
if (cl.getRed() == 0)
{
    Point p1 = new Point(x - 1, y);
    if (_point.contains(p1))
    {

```

```

        if (k > 5)
            if ((x - 1 == _basx) && (y == _basy))
                return;
    }
else
    {
        _alx.add(x - 1);
        _aly.add(y);
        _point.add(p1);
        komsuluk(x - 1, y);
    }
}

cl = new Color(Resim.getRGB(x , y - 1));
if (cl.getRed() == 0)
    {
        Point p1 = new Point(x , y - 1);
        if (_point.contains(p1))
            {
                if (k > 5)
                    if ((x == _basx) && (y - 1 == _basy))
                        return;
            }
else
    {
        _alx.add(x);
        _aly.add(y - 1);
        _point.add(p1);
        komsuluk(x, y - 1);
    }
}

cl = new Color(Resim.getRGB(x - 1, y - 1));
if (cl.getRed() == 0)
    {
        Point p1 = new Point(x - 1, y - 1);
        if (_point.contains(p1))
            {
                if (k > 5)
                    if ((x - 1 == _basx) && (y - 1 == _basy))
                        return;
            }
else
    {
        _alx.add(x - 1);
        _aly.add(y - 1);
        _point.add(p1);
        komsuluk(x - 1, y - 1);
    }
}

cl = new Color(Resim.getRGB(x + 1 , y - 1));

```

```

    if (cl.getRed() == 0)
    {
        Point p1 = new Point(x + 1, y - 1);
        if (_point.contains(p1))
        {
            if (k > 5)
                if ((x + 1 == _basx) && (y - 1 == _basy))
                    return;
        }
        else
        {
            _alx.add(x + 1);
            _aly.add(y - 1);
            _point.add(p1);
            komsuluk(x + 1, y - 1);
        }
    }

    cl = new Color(Resim.getRGB(x - 1, y + 1));
    if (cl.getRed() == 0)
    {
        Point p1 = new Point(x - 1, y + 1);
        if (_point.contains(p1))
        {
            if (k > 5)
                if ((x - 1 == _basx) && (y + 1 == _basy))
                    return;
        }
        else
        {
            _alx.add(x - 1);
            _aly.add(y + 1);
            _point.add(p1);
            komsuluk(x - 1, y + 1);
        }
    }
}
}
}

```

Class StoringSDB
public void addingData(string vectorPolygon) Vektörel formdaki görüntüyü mekânsal veritabanına kaydeder.
try { conn = DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/vectorization", "postgres", "postgres"); PreparedStatement st = conn.prepareStatement("SELECT

```

ST_AsText(ST_ConvexHull(ST_Collect(ST_GeomFromText(""+sorgu+""))) as
Convex, "
    +
"ST_AsText(ST_ConvexHull(ST_Collect(ST_GeomFromText(""+sorgu2+""))) ) as
Convex2");
    ResultSet rs = st.executeQuery();
    while(rs.next()){
        sorgudis=(rs.getString("Convex").toString());
        sorgudis2=(rs.getString("Convex2").toString());
    }
rs.close();
st.close();
    PreparedStatement st1 = conn.prepareStatement("INSERT INTO dbpolygon
VALUES(""+poligon_name+"",""+sorgudis+"","alan+","cevre+","")");
    ResultSet rs1 = st1.executeQuery();
    rs1.close();
    st1.close();
conn.close();
} catch (SQLException e) {
System.err.println( "Connection attempt failed" );
System.err.println( e.getMessage());
}

```

KİŞİSEL YAYINLAR VE ESERLER

A. Uluslararası Hakemli Dergi Yayınları

1. Sayar A., **Eken S.**, An Automated Technique to Determine Spatiotemporal Changes in Satellite Island Images with Vectorization and Spatial Queries (Gönderildi), 2012, *Advanced Science Letters*.
2. **Eken S.**, Sayar A., Vectorization and Spatial Query Architecture on Island Satellite Images (Kabul edildi), 2012, ELSEVIER, *Procedia Technology*.
3. **Eken S.**, Türkoğlu S., Sayar A., Integration of OpenGL Graphic Libraries with Spatial Database as an Analysis and Visualization Tool (Gönderildi), 2012, *Teknik-Online Dergi*.

B. Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler

1. **Eken S.**, Sayar A., Performance Evaluations of Vector-Raster Satellite Image Transfers through Web Services (Kabul edildi), *COMPSAC 2012 : The IEEE Signature Conference on Computer Software & Applications*, İzmir, 16–20 July, 2012.
2. Ulvi M., **Eken S.**, Sayar A., Service Oriented Visual Interpretation Tool for Time Series Data (Kabul edildi), 8. *International Symposium of Statistics*, Eskişehir, 11-13 Ekim, 2012.
3. **Eken S.**, Sayar A., Vector Modelling of Island Satellite Images for Spatial Databases, *Proceedings Of International Science and Technology Conference*, İstanbul, 7-9 Aralık, 2011.
4. Kakız M. T., Gün C., **Eken S.**, Sayar A., REACT: REMote Access and ConTrol with .NET Technology (Gönderildi), *9th International Conference on Electronics Computer and Computation*, Ankara, 2012.

C. Ulusal Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında Basılan Bildiriler

1. Özel A., Kaya Ç., Eşmeli R., **Eken S.**, Sayar A., Web Servisler ile Paralel Görüntü İşlem Mimarisi: Raster İmgelerde Kenar Belirleme Uygulaması (Kabul

edildi), *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, Trabzon, 3-4 Temmuz, 2012.

2. **Eken S.**, Bostancı R., Sayar A., Dağıtık Sistemlerde Uydu Görüntüsü Transferi: Raster ve Vektör Temsil Karşılaştırması (Kabul edildi), *IV. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu (UZAL-CBS 2012)*, Zonguldak, 16-19 Ekim, 2012.

ÖZGEÇMİŞ

1985 yılında Konya’da doğdu. İlk, orta ve lise öğrenimini Konya’da tamamladı. 2004 yılında girdiği Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği’den 2009 yılında bölüm 4. olarak mezun oldu. 2010 yılında başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bilgisayar Bilimleri Anabilim Dalı’ndaki Yüksek Lisans eğitimine devam etmektedir. 2010 yılından itibaren Kocaeli Üniversitesi Bilgisayar Mühendisliği araştırma görevlisi olarak çalışmaktadır.