

**ARAÇLARIN UYDULAR (GPS - GLOBAL POSITIONING
SYSTEM) YARDIMI İLE DÜNYA ÜZERİNDEKİ KONUM VE
HAREKETLERİNİN SAYISAL HARİTALAR ÜZERİNDE
İZLENMESİ**

**Pamukkale Üniversitesi
Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı**

Önder ÇİVRİL

Danışman: Yrd. Doç. Dr. A. Kadir YALDIR

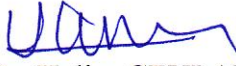
**Aralık 2006
DENİZLİ**

YÜKSEK LİSANS TEZİ ONAY FORMU

Önder Çivril tarafından Yrd. Doç. Dr. A. Kadir YALDIR yönetiminde hazırlanan “Araçların uydular (GPS - Global Positioning System) yardımı ile dünya üzerindeki konum ve hareketlerinin sayısal haritalar üzerinde izlenmesi” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans Tezi olarak kabul edilmiştir.



Prof. Dr. Halil KUMSAR
Jüri Başkanı



Doç. Dr. Halim CEYLAN
Jüri Üyesi



Yrd. Doç. Dr. A. Kadir YALDIR
Jüri Üyesi (Danışman)

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
.../.../..... tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Mehmet Ali SARIGÖL
Müdür

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmaların yapılması ve bulguların analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalışmalara atfedildiđini beyan ederim.

İmza

Öğrenci Adı Soyadı : Önder ÇİVRİL



TEŞEKKÜR

Bu tez çalışmasını Bilimsel Araştırma Projeleri (BAP) kapsamında destekleyen Pamukkale Üniversitesi B. A. P. Birimine; çalışmada kullanılan GPS alıcılarını bize sağlayan ve gerektiğinde yazılım için bana destek veren TAGEM firması yetkili ve çalışanlarına; görüş, öneri ve yardımlarından dolayı Doç. Dr. Şahin CONKUR ve Tez Danışmanım Yrd. Doç. Dr. A. Kadir YALDIR hocalarıma; son olarak, beni yetiştiren ve üzerimde çok büyük emekleri olan annem Esmâ ÇİVRİL ve babam Emin ÇİVRİL'e gönülden teşekkür ederim.

ÖZET

ARAÇLARIN UYDULAR (GPS - GLOBAL POSITIONING SYSTEM) YARDIMI İLE DÜNYA ÜZERİNDEKİ KONUM VE HAREKETLERİNİN SAYISAL HARİTALAR ÜZERİNDE İZLENMESİ

Çivril, Önder

Yüksek Lisans Tezi, Bilgisayar Mühendisliği ABD

Tez Yöneticisi: Yrd. Doç. Dr. A. Kadir YALDIR

Aralık 2006, 65 Sayfa

Araç Takip Sistemleri, yakıt tasarrufu, güvenlik, araç filolarının etkin yönetimi gibi değişik amaçlarla giderek yaygınlaşan bir kullanım alanına sahiptir. Bu sistemler ile, her türlü ulaşım aracı, nesne ve canlı gerçek zamanlı olarak izlenebilmekte, konumları ile ilişkili her türlü detaylı veritabanı analizleri yapılabilmektedir.

Sistem kabaca üç bileşenden oluşur: Uydular yardımı ile aracın yeryüzündeki konumunu saptamak için kullanılan bir GPS alıcısı, saptanan konum bilgilerini belirli bir merkeze iletmek için kullanılan bir iletişim modülü ve iletilen konum bilgilerini kullanarak aracın sayısal haritalar üzerinde izlenebilmesini ve konuma bağlı analizlerin yapılabilmesini sağlayan bir bilgisayar yazılımı.

Bu çalışmayla geliştirilen araç takip sisteminde, GPS alıcısı ile saptanan konum bilgilerinin iletimi için GPRS teknolojisi seçilmiştir. Bu yöntemde, GSM operatörleri ücretlendirmeyi iletilen veri miktarına göre yaptığından maliyet düşük olabilmektedir. Ayrıca, veriler iletilmeden önce sıkıştırılarak maliyetin daha da düşürülmesi olanaklıdır.

Bu çalışma kapsamında, araçların bilgisayar ekranı üzerinden izlenebilmesi için Microsoft Visual C# .NET ortamında bir yazılım geliştirilmiştir. Bu yazılım, MapInfo mif ve gst formatlarında hazırlanmış sayısal haritaların görüntülenebilmesini ve araçların bu haritalar üzerinde izlenebilmesini sağlamaktadır.

Anahtar Kelimeler: Araç Takip Sistemi, GPS, GIS

Prof. Dr. Halil KUMSAR

Doç. Dr. Halim CEYLAN

Yrd. Doç. Dr. A. Kadir YALDIR

ABSTRACT**TRACKING THE POSITIONS AND MOVEMENTS OF VEHICLES ON EARTH BY MEANS OF SATELLITES (GPS - GLOBAL POSITIONING SYSTEM) ON DIGITAL MAPS**

Çivril, Önder

M. Sc. Thesis in Computer Engineering

Supervisor: Assoc. Prof. Dr. A. Kadir YALDIR

December 2006, 65 Pages

The usage of Vehicle Tracking Systems become gradually more common with various purposes like fuel saving, security, effective management of vehicle fleets. With these systems it is possible to track any type of vehicle, object, or even living things, and any type of detailed database analysis based on the positioning data can be conducted.

A vehicle tracking system consists of mainly three components: A GPS receiver used to determine the position of the vehicle, a communication unit used to send the positioning data to the tracking centre, and a software package used to track the vehicle on digital maps and to carry out the positional analysis.

For the system developed in this study, GPRS technology is chosen to send the positioning data determined by the GPS receiver. The communication cost of the system is low with this technology since the costs are determined by the amount of the data transmitted. Moreover, the communication costs can be reduced further by compressing the data before transmission.

In this study, a software was developed in Microsoft Visual C# .NET environment to display the digital maps prepared in MapInfo `mif` and `gst` formats on computer screen and track the vehicles on these maps.

Keywords: Vehicle Tracking System, GPS, GIS

Prof. Dr. Halil KUMSAR

Assoc. Prof. Dr. Halim CEYLAN

Asst. Prof. Dr. A. Kadir YALDIR

İÇİNDEKİLER

	<u>Sayfa No</u>
Yüksek Lisans Tezi Onay Formu.....	i
Bilimsel Etik Sayfası.....	iii
Teşekkür.....	iii
Özet.....	iv
Abstract.....	v
İçindekiler.....	vi
Şekiller Dizini.....	viii
Simge ve Kısaltmalar Dizini.....	ix
1. GİRİŞ.....	1
2. ARAÇ TAKİP SİSTEMLERİ.....	3
2.1 Sistemin Genel Yapısı.....	3
2.2 Araç Takip Sistemlerinin Kullanım Alanları.....	5
2.3 Araç Takip Sistemlerinin Sağladığı Yararlar.....	5
2.4 ATS Yazılımları.....	6
3. KÜRESEL KONUMLAMA SİSTEMİ (KKS).....	9
3.1 Uzay Bölümü.....	10
3.2 Kontrol Bölümü.....	11
3.3 Kullanıcı Bölümü.....	12
3.4 KKS Servisleri.....	12
3.5 KKS Çalışma Prensibi.....	13
3.5.1 Sinyallerin geliş sürelerinin ölçümü.....	13
3.5.2 Konumun hesaplanması.....	14
3.6 KKS Verileri.....	14
4. SAYISAL HARİTALAR.....	16
4.1 Harita Katmanları.....	16
4.2 Konum Bilgilerinin Modellenmesi.....	17
4.2.1 Vektör model.....	18
4.2.2 Raster model.....	18
4.3 Koordinat Sistemleri.....	20
4.4 Coğrafik Koordinat Sistemi ve Referans Elipsoitler.....	21
4.5 Harita Projeksiyonları.....	23
5. ATS YAZILIMI GELİŞTİRİLMESİ.....	25
5.1 İzleme Yazılımı.....	25
5.2 Yazılımda Kullanılan Önemli Sınıflar.....	29
5.3 Yazılımda Kullanılan Teknikler.....	30
5.3.1 Koordinat dönüşümü.....	30
5.3.2 Multithreading.....	32
5.3.3 Çoklu arabellekleme (double buffering).....	33
5.3.4 Dolgu desenleri.....	34
5.3.5 Görüntüleme işleminin hızlandırılması.....	36
5.3.6 Soket programlama.....	38

5.4 Mevcut Ticari Yazılımlar ile Geliştirilen Yazılımın Karşılaştırılması	39
5.5 Yazılımın Test Edilmesi ve Örnek ATS	40
6. SONUÇ VE ÖNERİLER	43
EKLER	45
KAYNAKLAR	63
ÖZGEÇMİŞ	65

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 2.1 Bir araç takip sisteminin genel yapısı.....	4
Şekil 3.1 KKS bölümleri (U.S. Coast Guard Navigation Center 1996)	9
Şekil 3.2 KKS uyduları (U.S. Coast Guard Navigation Center 1996).....	10
Şekil 3.3 KKS ana kontrol merkezi, gözlem istasyonları ve antenler (U.S. Coast Guard Navigation Center 1996).....	11
Şekil 3.4 Uydulardan gelen sinyallerin geliş süresinin ölçümü.....	13
Şekil 3.5 KKS ile konum hesaplanması (EUROCONTROL and IfEN 1998).	14
Şekil 4.1 Harita katmanları (Escobar vd 2001).....	17
Şekil 4.2 Vektör model (Escobar vd 2001).....	18
Şekil 4.3 Raster model (Escobar vd 2001).....	19
Şekil 4.4 Run-length encoding (Escobar vd 2001)	19
Şekil 4.5 İki boyutlu kartezyen koordinat sistemi.	20
Şekil 4.6 Üç boyutlu kartezyen koordinat sistemi.	20
Şekil 4.7 Üç boyutlu kutupsal koordinat sistemi.	21
Şekil 4.8 Coğrafik koordinat sistemi	21
Şekil 4.9 Elipsoit ve parametreleri.....	22
Şekil 4.10 Projeksiyonun harita üzerindeki etkisi (Dana 2000).	23
Şekil 4.11 Harita projeksiyonları	24
Şekil 5.1 Sistemin birinci seviye veri akış diyagramı.....	25
Şekil 5.2 Sistemin ikinci seviye veri akış diyagramı	26
Şekil 5.3 Yazılımın kullanıcı arabirimi.....	27
Şekil 5.4 Bilgisayar ekranının koordinat sistemi	31
Şekil 5.5 Koordinat dönüştürme işlemlerini gerçekleştiren kod örneği.	32
Şekil 5.6 Bir thread oluşturulup başlatılması.....	33
Şekil 5.7 Çoklu arabellekleme kod örneği.....	33
Şekil 5.8 MapInfo dolgu desenleri.....	35
Şekil 5.9 Desenlerle doldurulmuş alanlar.	35
Şekil 5.10 Desenin ön alan renginin değiştirilmesi.	36
Şekil 5.11 Araç ünitesi (a) ve antenler (b).	40
Şekil 5.12 Yazılımdan araç izleme sırasında alınmış bir ekran görüntüsü.....	401

SİMGE VE KISALTMALAR DİZİNİ

API	Application Programming Interface
ATS	Araç Takip Sistemi
CBS	Coğrafya Bilgi Sistemi
GDI	Graphics Drawing Interface
GIS	Geographic Information System
GLONASS	Global Navigation Satellite System
GPRS	General Packet Radio Services
GPS	Global Positioning System
GSM	Group Special Mobile
GUI	Graphical User Interface
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
KKS	Küresel Konumlama Sistemi
NAVSTAR	Navigation System with Time And Ranging
NMEA	National Marine Electronics Association
SMS	Short Message Services
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WGS	World Geodetic System

1. GİRİŞ

Konum ve yön bulma insanoğlunun çok eskiden beri üzerinde uğraştığı önemli bir problemdir. Ancak, özellikle iletişim, uydu ve bilgisayar alanındaki gelişmeler bu problemi ortadan kaldırmış, kurumsal ve bireysel pek çok uygulama olanaklı hale gelmiştir.

Araç takip sistemleri (ATS) de bu gelişmelerle birlikte ortaya çıkan yüksek teknoloji ürünlerinden biridir. Bu sistemlerle sadece ulaşım araçları değil her türlü nesne, hatta canlılar ister gerçek zamanlı, ister konum bilgilerinin kaydedilmesiyle daha sonradan izlenebilmektedir. Bu nedenle ATS, bireysel alandan askeri ve sivil her türlü kurumsal alana kadar son derece geniş bir kullanıma sahiptir.

Bu sistemler kurumsal alanda genellikle güvenlik, araç filolarının yönetimi, yakıt tasarrufu gibi değişik amaçlarla kullanılmaktadır. Aracın hızı, yönü, izlediği güzergah öğrenilebilmekte, sürücünün belirlenen güzergah ve hız limitlerine uyup uymadığı kontrol edilebilmektedir. Böylece kaza olasılıkları azaltılabilmekte, aracın kişisel işlerde kullanılıp kullanılmadığı belirlenebilmektedir. Araç filoları daha iyi bir şekilde organize edilip iş verimliliğinde artış sağlanabilmektedir. Aracın duruş ve mola süreleri kontrol edilip olağan dışı durumlar belirlenebilmekte, özellikle yakıt alımlarında sıkça görülen sahteciliğe karşı da kullanılabilir. Araçların taşıdıkları değerli mallar da bu sistemle izlenebilmektedir. Böylece bir güvenlik önlemi olarak kullanılabilmesi gibi, hırsızlık durumunda araç ya da malların kolaylıkla bulunması sağlanabilmektedir.

Araç takip sistemleri, araçlara takılan elektronik aygıtlar aracılığı ile elde edilen araç konumlarının sayısal haritalar üzerinde izlenmesine olanak veren sistemlerdir. Araç konumlarını elde etmek için kullanılan aygıtlar bu işlem için uydulardan gelen sinyalleri kullanan GPS alıcılarından oluşur. Konum bilgileri bir iletişim modülü kullanılarak izlemenin gerçekleşeceği merkeze iletilir. Bu iki bileşen birlikte araca monte edilen araç modülünü oluşturur. İzleme merkezinde, gelen konum bilgileri bir yazılım aracılığı ile bilgisayar ekranında görüntülenen sayısal haritalar üzerine işlenerek araçlar izlenir.

Bu tez çalışmasında MapInfo gis ve mif formatında hazırlanan sayısal haritaları görüntüleyen ve internet üzerinden gelen konum bilgilerinin bu haritalar üzerinde izlenebilmesini sağlayan bir yazılım geliştirilmiş, bu yazılım kullanılarak örnek bir araç takip sistemi oluşturulmuştur. Yazılım, Microsoft .NET platformu üzerinde çalışan Visual C# programlama dili kullanılarak geliştirilmiştir. Böylece görsel bir programlama dilinin ve Microsoft .NET platformunun getirdiği olanaklardan yararlanılmıştır.

2. Bölümde araç takip sistemleri hakkında genel bilgiler verilmiştir. Araç takip sistemlerinin yapısı açıklanmış, kullanım alanları, sistemin getirdiği yararlılardan söz edilmiştir. Son olarak, yaygın kullanıma sahip ticari yazılımlarla ilgili bilgilere de yer verilmiştir.

3. Bölümde, yeryüzündeki herhangi bir noktanın koordinatlarının elde edilebilmesi için geliştirilen Küresel Konumlama Sistemi – KKS (Global Positioning System GPS) tanıtılmıştır. Araç takip sistemlerinde araçlara, bu sistemi kullanarak konum bilgilerini hesaplayan GPS alıcıları monte edilir. Bu bölümde GPS'in çalışma prensipleri açıklanmıştır.

4. Bölüm sayısal haritalar ile ilgili bilgiler içermektedir. Sayısal haritalar araç takip sisteminde araç konumlarının izlenmesinde kullanılır. Bu nedenle sistemin önemli bir parçasıdır. Harita katmanları, veri modeli, koordinat sistemleri, elipsoitler, harita projeksiyonları gibi bu konuya ait önemli kavramlar bu bölümde açıklanmaktadır.

5. Bölüm geliştirilen yazılıma ayrılmıştır. Yazılım geliştirilirken karşılaşılan sorunlar, bu sorunların çözümleri, kullanılan teknikler ayrıntıları ile açıklanmıştır. Yazılım örnek bir uygulama ile test edilmiş ve elde edilen sonuçlar irdelenmiştir.

Son olarak 6. Bölümde bu çalışma ile elde edilen sonuçlara yer verilmiştir. Görüş ve öneriler yine bu bölümde yer almaktadır.

2. ARAÇ TAKİP SİSTEMLERİ

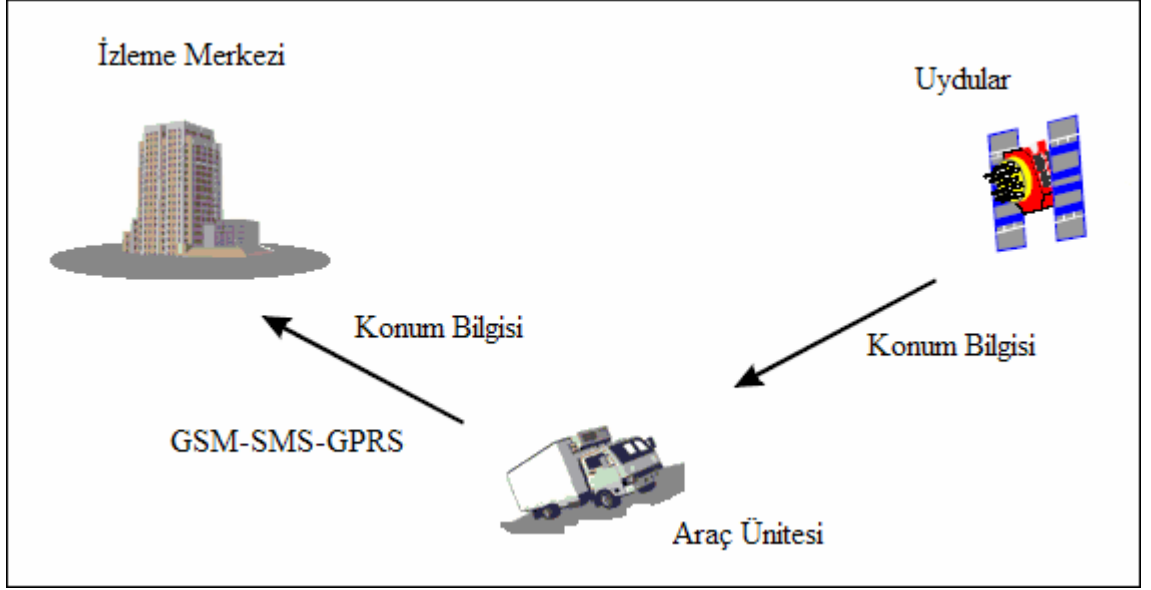
Araç takip sistemleri (ATS), herhangi bir anda bir aracın konumunun bilinmesine ve zaman içinde izlenmesine olanak veren sistemlerdir. İzleme araca monte edilen elektronik aygıtlar aracılığı ile gerçekleştirilir. Modern sistemlerde araç konumlarının kolayca ve en doğru şekilde elde edilebilmesi için GPS modülleri kullanılır. Araç konumlarının izlenmesinde tek tek ya da birlikte kullanılabilen iki yöntem vardır. Bu yöntemler, gerçek zamanlı anlık (online) izleme ya da kapalı devre (offline) izlemedir. Birinci yöntemde araç konumlarının izlemenin gerçekleştirileceği merkeze iletilmesi gerekir. Bu bakımdan araç, GPS modülü ile birlikte bir de iletişim bileşeni içerir. İletişim bileşeni olarak oldukça yaygın bir biçimde GSM modülleri kullanılır. Diğer yöntemde ise bilgiler araç ünitesinde bulunan bir belleğe kaydedilir. Bu bilgiler, daha sonra bilgisayara aktarılarak izleme gerçekleşir. İki yöntemin birlikte kullanılması, iletişimde ortaya çıkabilecek sorunlardan kaynaklanan aksaklıkları da bir ölçüde giderir.

İzleme bir sayısal harita üzerine araç konumlarının işlenmesi şeklinde olur. Sistemde bu işlev internet üzerinden sağlanan bir servisle elde edilebileceği gibi, özel olarak geliştirilen yazılımlar da kullanılabilir.

Bu bölümde öncelikle sistemin genel yapısı verilerek sistemi oluşturan bileşenler ve görevleri açıklanmış, ardından kullanım alanları ve getirdiği faydalardan söz edilmiştir. Son olarak, araç konumlarının sayısal haritalar üzerinde izlenebilmesi için geliştirilen mevcut ATS yazılımları ile ilgili bilgiler verilmiştir.

2.1 Sistemin Genel Yapısı

Bir araç takip sisteminin genel yapısı Şekil 2.1’de verilmiştir. Sistem temel olarak üç bileşenden oluşur. Bu bileşenlerden ilki aracın bulunduğu konumu hesaplar. Bu amaç için genellikle bir GPS alıcısı kullanılır. İkinci bileşen, elde edilen konum bilgilerinin izleme işleminin gerçekleştirileceği merkeze iletilmesini sağlayan bir iletişim modülüdür. Bu iki bileşen birlikte araç ünitesini oluştururlar ve tek bir ünite şeklinde satın alınabilirler. Sonuncu bileşen ise bir bilgisayar yazılımıdır. Bu yazılım, sayısal haritalar üzerinde araç konumlarının izlenebilmesi için kullanılır.



Şekil 2.1 Bir araç takip sisteminin genel yapısı

Konum bilgilerinin elde edilmesi için uydu sistemlerinden faydalanılır. Günümüzde bu amaç için kullanılan iki uydu sistemi vardır: ABD'ne ait Navstar GPS ve Rusya'ya ait olan GLONASS. Ancak GLONASS, başlangıcındaki gizlilik ve sisteme ait bilgi azlığı nedeniyle çok az kullanıcı ekipmanına sahiptir. Üstelik GPS ekipmanları ile karşılaştırıldığında bu ürünler oldukça pahalıdır. Bu yüzden araç takip sistemlerinde GPS alıcıları kullanılmaktadır. Alıcılar, GPS uydularından gelen sinyalleri kullanarak buldukları yerin enlem ve boylamını hesaplayan elektronik aygıtlardır. Böylece monte edilmiş oldukları aracın yeryüzündeki konumu saptanmış olur.

Daha sonra, saptanan bu konum bilgilerin izleme merkezine iletilmesi gerekir. Bu işlev iletişim modülü tarafından yerine getirilir. Burada radyo iletişimi, SMS, GPRS gibi değişik alternatifler söz konusu olabilir. Düşük maliyet nedeniyle GPRS daha yaygın olarak kullanılmaktadır. GPRS, GSM şebekesi üzerinden internet erişimine olanak veren bir tekniktir. Ücretlendirme, transfer edilen veri büyüklüğü üzerinden yapıldığından maliyet de daha düşük olmaktadır. Ancak veriler internet üzerinden geleceğinden izlemenin yapılacağı bilgisayarın internet bağlantısı olması gerekir.

Son olarak, izleme merkezine gelen veriler bir harita üzerine işaretlenerek aracın izlenebilmesi sağlanır. İzleme bir bilgisayar ekranı aracılığıyla gerçekleşir. Bu bileşen, sayısal haritaları görüntüleyen, konuma bağlı veritabanı analizlerinin de yapılabildiği bir bilgisayar yazılımından oluşur.

2.2 Araç Takip Sistemlerinin Kullanım Alanları

Araç takip sistemleri kişisel ve kurumsal, askeri ve sivil, değişik sektörlerde pek çok kullanım alanına sahiptir. Araç konumlarının, değişik amaçlarla bilinmesine gerek duyulan herhangi bir alanda kullanılabilir. Bu alanlar başlıca şu şekilde sıralanabilir:

- Taşımacılık: Şehirlerarası yük ya da yolcu taşımacılığı yapan firmalarda, araç konumlarının izlenmesi ile araç filolarının etkin yönetimi amacıyla kullanılır.
- Taksicilik: Müşteriye en yakın taksinin yönlendirilmesi, gidilecek adreslere en uygun güzergahın bulunması ve güvenlik gibi amaçlarla kullanılır.
- Finans: Bankacılık, sigortacılık gibi alanlarda güvenlik amacıyla kullanılır. Aracın çalınması durumunda konumunun bulunması, merkezden müdahalede bulunularak, motorun durdurulması, kapıların kilitlenmesi, alarmı çalıştırma gibi olanaklar verir.
- Araç kiralama şirketleri: Araç konumlarının öğrenilmesi, müşterinin sözleşme şartlarına (hız sınırı gibi) uyup uymadığının belirlenmesi ve güvenlik gibi amaçlarla kullanılır.
- Acil yardım hizmeti ve mobil hizmet veren kuruluşlar: Polis, itfaiye, hastane gibi acil yardım hizmeti veren kuruluşlarla, mobil hizmet veren kuruluşlarda gidilecek adrese en yakın aracın bulunarak yönlendirilmesi, en kısa güzergahın belirlenmesi gibi amaçlarla kullanılır.
- Kişisel araçlar: Kişisel araçlarda genellikle hırsızlık durumunda aracın bulunması gibi amaçlarla kullanılır.
- Şehir içi taşımacılık ve servis araçları: Duraklara varış ve ayrılışlarının izlenmesi amacı ile kullanılır.

2.3 Araç Takip Sistemlerinin Sağladığı Yararlar

Araç takip sistemlerinin kullanımının bu kadar yaygın olmasının önemli bir nedeni sağladığı yararlarıdır. Bu yararları şu şekilde özetleyebiliriz.

- Güvenlik konusunda oldukça yararlar sağlar. Araç konumları sürekli izlendiğinden olağan dışı durumlar kolaylıkla belirlenip süratle müdahale edilebilmektedir. Gerektiğinde kapılar merkezden kilitleyip, motorlar durdurulabilmekte, alarm çalıştırılabilmektedir. Bu yönleri ile güvenliği artırır.
- Aracın çalınması durumunda aracın kolaylıkla yeri belirlenebilir. Bu nedenle, bu tür donanımlara sahip araçlar, sigorta şirketleri tarafından daha ucuza sigortalanır.
- İşletme maliyetlerini düşürür. Araçların belirlenen güzergah ve hız limitlerine uyup uymadığı sürekli kontrol edilerek yakıt tasarrufu sağlanır. Ayrıca, aracın özel işlerde kullanılıp kullanılmadığı kolaylıkla belirlenebileceğinden bu tür olumsuzlukların önüne geçilir. Yakıt alımları kontrol altına alınır. Bakım giderleri düşürülür.
- Hız limiti gibi uygulanabilecek bazı kurallarla kaza olasılıkları azaltılarak şoför, mal ve yolcu emniyeti artırılır.
- Araç ve merkez arasında sesli ve/veya görüntülü iletişim olanağı verir.
- Araç filoların etkin yönetimini ve daha iyi bir planlama olanağı sağlar. İş verimliliğini artırır.
- Hizmette güvenilirliği ve müşteri isteklerinin süratle yerine getirilmesi ile de müşteri memnuniyetini artırır.

2.4 ATS Yazılımları

Araç takip sistemleri daha çok ticari alanda geliştirilen sistemlerdir. Bu konuda pek çok ticari yazılım bulunmaktadır. Bu sistemler gelişmelerini diğer alanlarda yapılan çalışmalara borçludurlar. Örneğin konumun belirlenebilmesi ve daha hassas bir şekilde hesaplanması GPS alanında yapılan çalışmalara bağlıdır. Sayısal haritalar konusundaki ilerlemeler coğrafi bilgi sistemleri (CBS) alanında yapılan çalışmalarla olmaktadır. Bu alanda pek çok üniversite ve kuruluştta çalışmalar yapılmış ve yapılmaktadır. Bunlardan bazıları CBS geliştirmeye ya da yazılımlara CBS ve sayısal haritalarla ilgili işlevlerin

kazandırılmasına yarayan yazılım araçlarının geliştirilmesine yöneliktir. Bu araçlarla, bu tür işlevler gerektiren yazılımlar süratle ve kolaylıkla geliştirilebilmektedir. Bazı çalışmalar ise açık kaynak kodlu olup tüm dünyadan gönüllü katılımcıların katkılarıyla sürdürülmektedir. Konum bilgilerinin iletimi ise iletişim alanındaki gelişmelerle ilgilidir. Örneğin cep telefonu şebekelerinden internet erişimine olanak veren GPRS teknolojisinin gelişimi, işletme maliyetleri daha düşük araç takip sistemlerinin ortaya çıkmasına neden olmuştur.

İnternet üzerinde Google arama motoru kullanılarak, sadece Türkçe sayfalarda yapılan bir aramada dahi oldukça büyük sayıda ATS yazılımı ile karşılaşıldı. Elbette bunların tümü incelenemedi. Ancak, incelenebilen tüm ticari yazılımlar aynı temel işlevleri yerine getirmekte ve benzer özellikler taşımaktadırlar. Bu sistemleri temel olarak iki sınıfa ayırabiliriz; internet üzerinden hizmet veren sistemler ve özel yazılımlar.

İnternet üzerinden hizmet veren sistemlerde kullanıcılar herhangi bir yazılım satın almak zorunda değildirler. Bunun yerine, aylık, haftalık, vb. ücret karşılığında verilen hizmete abone olur. Araçlara takılan aygıtlar konum bilgilerini hizmeti veren bu merkeze iletirler. Kullanıcı verilen bir kullanıcı adı ve şifre ile ve bir web tarayıcısı kullanarak, hizmeti veren kuruluşun internet sitesine bağlanır. Bu şekilde kendi araçlarını web üzerinden izler.

Özel yazılımlar sistemin bir parçası olarak sayısal haritalar ile birlikte satın alınırlar. Bu yazılımların temel işlevi konumların haritalar üzerinde izlenebilmelerini sağlamaktır. Bunun yanında tümünde haritaların değişik bölümlerinin, değişik zoom seviyelerinde görüntülenebilmesi gibi özellikler vardır. Yazılımların ürettiği raporlar da birbirlerinden çok farklı değildir. Yine tümü, verilerin kaydedilmesi ve geçmişe yönelik sorgulamalar yapılabilmesi gibi ortak özellikler taşırlar. Birden çok aracın izlenmesi, belli bir yere en yakın aracın belirlenmesi, en kısa güzergahın bulunması, hız limitinin aşılması uyarısı gibi özellikler de tespit edilen diğer ortak özelliklerdir.

Araç takip sistemi yazılımları, temel işlevleri yanında sahip oldukları diğer ekstra özellikler bakımından birbirlerinden farklılıklar gösterebilirler. Bu da aslında araç ünitesinin sahip olduğu özellikler ve araca monte edilecek algılayıcılara bağlıdır. Bu yöntemlerle yazılıma katılabilecek özellikler, ses ve görüntü iletimi, kapıların açık ya da

kapalı oldukları bilgisi, yakıt durumu ile ilgili bilgiler, aracın mekanik problemleri ile bilgiler, merkezden müdahale gibi özelliklerdir.

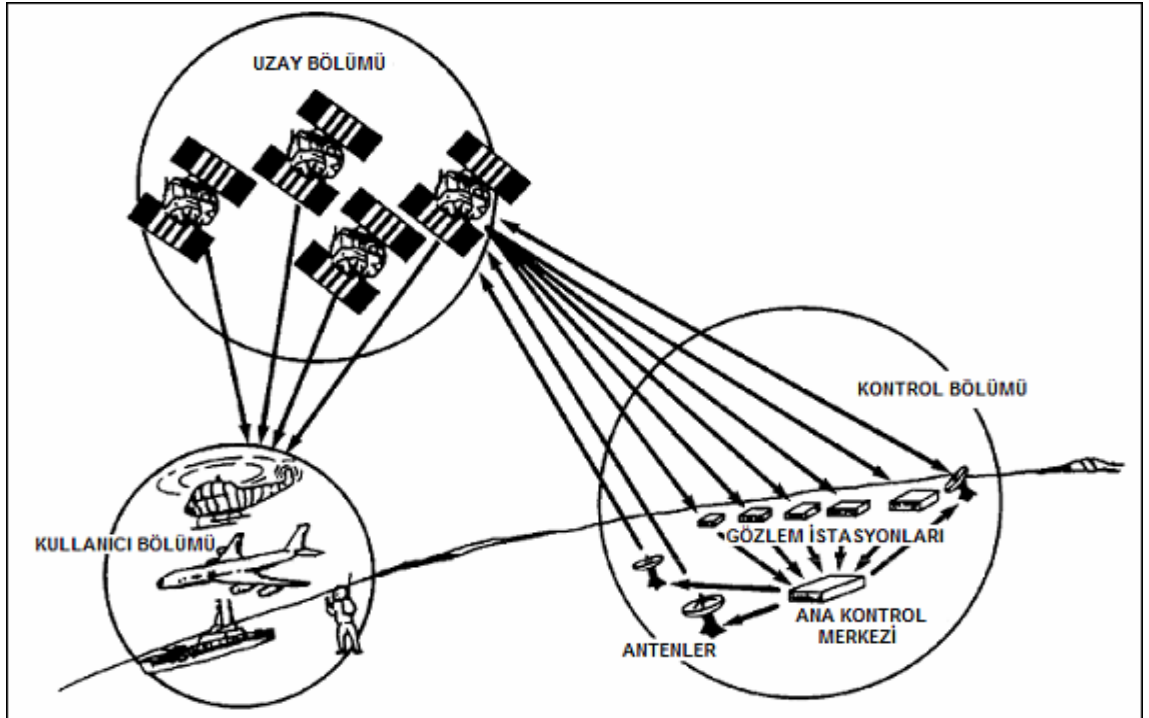
Dikkat çeken bir başka konu ise bu yazılımların, bazı hazır yazılım bileşenleri kullanılarak geliştirilmiş olmasıdır. Bu bileşen ve araçların kullanımı yazılımın çok kısa zamanda ve az sayıda elemanla geliştirilmesini sağlayarak verimliliği artırır, geliştirme maliyetlerini düşürür ve geliştirme sürecini hızlandırır.

Ancak, bu tür hazır bileşenler oldukça pahalı yazılımlardır. Üstelik, bunlar araç takip sistemleri temel alınarak hazırlanmamışlardır. Daha çok, CBS ve sayısal harita işlevselliği gerektiren yazılımlar için tasarlanmışlardır. Bu bileşenler kullanılarak, herhangi bir yazılıma söz konusu işlevler kazandırılabilir. Oysa araç takip sistemi yazılımları için bu bileşenlerin sağladığı olanakların ancak bir kısmı gereklidir, bu olanakların pek çoğu kullanılmaz.

Bir başka konu ise bu bileşenlerin sahip olduğu sayısal harita formatlarıdır. Kullanıcı bu formatlara bağımlı kalmakta, bu formatlarla hazırlanan haritaları edinmek durumunda kalmaktadır. Örneğin MapX (bir CBS yazılım bileşeni) ile geliştirilen bir yazılım kullanıyorsanız, haritanız MapInfo tab, gst veya mif formatında olmalıdır.

3. KÜRESEL KONUMLAMA SİSTEMİ (KKS)

Amerika Birleşik Devletleri Savunma Bakanlığı için geliştirilmiş olan bu sistem her türlü hava şartlarında araçların konum ve hızlarının belirlenmesini olanaklı kılar. KKS, sınırsız sayıda hava, deniz, kara ve uzay kullanıcıları için hassas konum, hız ve zaman bilgisi sağlar. KKS, yetkili kullanıcılar ve diğer kullanıcılar için farklı hassaslık derecelerine sahip hizmetler verir. Ayrıca, KKS hizmetlerini taklit ederek kullanıcıları aldatmaya yönelik girişimler için de güvenlik önlemlerine sahiptir. Sistem üç bölümden oluşur. Uzay bölümü, Kontrol bölümü ve Kullanıcı bölümü (U.S. Coast Guard Navigation Center 1996, EUROCONTROL ve IfEN 1998). Şekil 3.1 bu bölümleri göstermektedir. Uzay bölümü 27 adet uydudan oluşur. Her uydu, radyo frekansında mesafe tayin kodları ve bir seyir verisi mesajı yayımlar. Kontrol bölümü, uyduların yönetimi ve yayınladıkları mesajların güncellenmesi için kullanılan gözlem ve kontrol istasyonlarından oluşur. Kullanıcı bölümü ise mesafe tayin kodları ve seyir verisi mesajlarını alıp işlemek için özel olarak tasarlanmış alıcılardan oluşur.

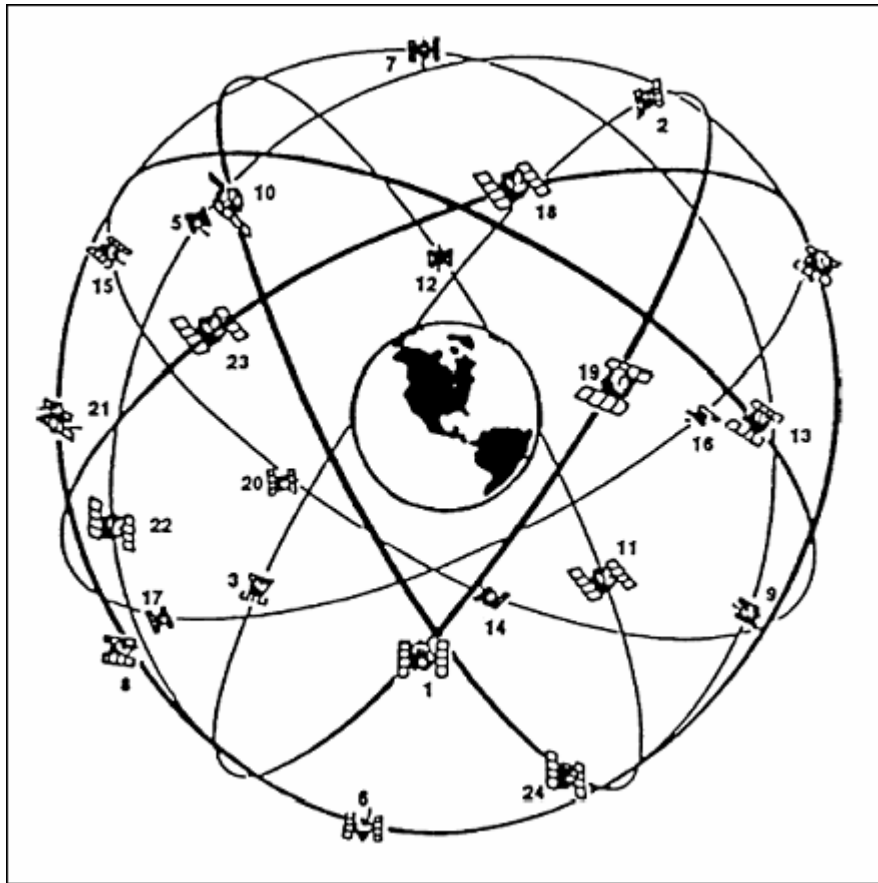


Şekil 3.1 KKS bölümleri (U.S. Coast Guard Navigation Center 1996)

Uydulardan yayınlanan mesafe tayin kodları, alıcının sinyallerin kendine ulaşması için geçen süreyi ölçmesine olanak verir. Böylece alıcı her uydu ile arasındaki mesafeyi belirleyebilir. Seyir verisi mesajı ise alıcının, her uydunun sinyallerin yayınlandığı andaki konumunu hesaplayabilmesini sağlar. Daha sonra alıcı bu bilgileri kullanarak kendi konumunu belirler. Aslında her mesafe ölçümü, merkezinde bir uydu olan bir küre tanımlar. Bu kürelerin, yeryüzü üzerinde veya yakınlarındaki kesişim noktası ise alıcının konumunu belirler. Bu işlem için en az dört uydunun alıcı tarafından görülebilir olması gerekir. Yani en az dört farklı uydudan gelen sinyaller alınabilmelidir. Böylece dört farklı mesafe yapılabilir. Bu ölçümler alıcının üç boyutlu konumunu gösteren bilinmeyen üç parametreyi ve bir dördüncü parametre olarak da alıcı saatindeki hata payının hesaplanabilmesini sağlar.

3.1 Uzay Bölümü

Uzay bölümü, 24'ü işler durumda ve 3'ü yedek olmak üzere toplam 27 adet uydudan oluşur (Lee 2003). Şekil 3.2 bu uyduları göstermektedir.



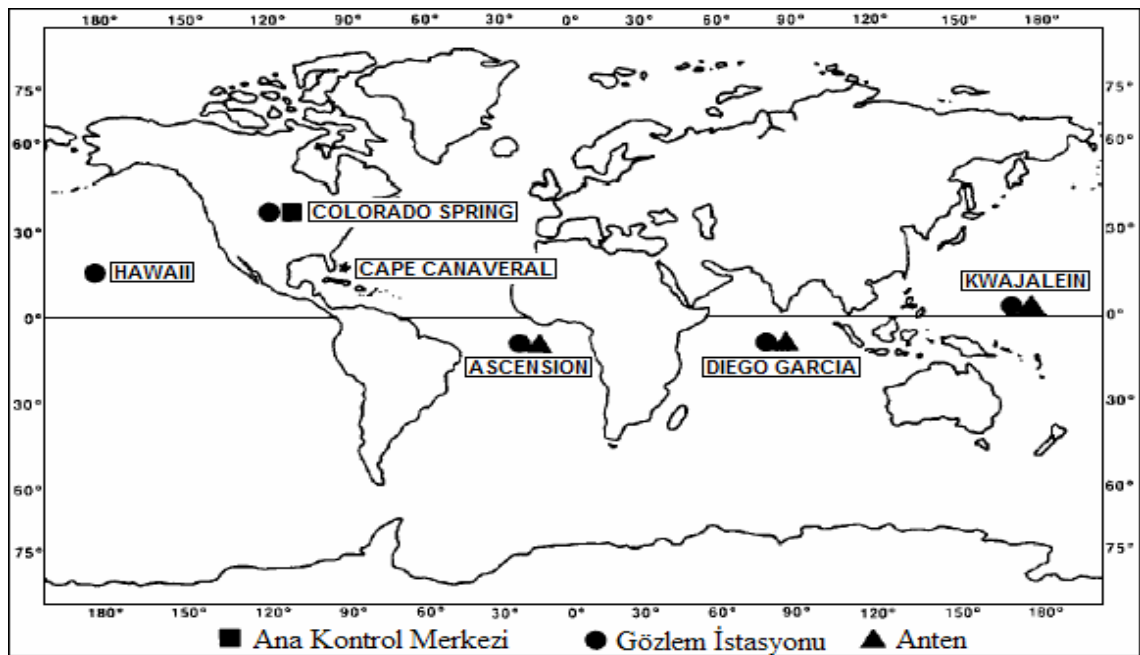
Şekil 3.2 KKS uyduları (U.S. Coast Guard Navigation Center 1996)

Uydular, yeryüzünden 20200 km. yükseklikte 6 farklı yörüngeye yerleştirilmişlerdir. Bu yörüngeler ekvator düzlemi ile 55 derecelik bir açı yaparlar. Yaklaşık olarak dairesel olan bu yörüngelerin her birinde 4 uydu bulunur. Uydular yörüngelerini yaklaşık olarak 12 saatte tamamlarlar ve yeryüzünün herhangi bir yerinden 4 ile 8 tanesi sürekli olarak görülebilecek bir şekilde konumlandırılmışlardır.

Her uydu, KKS alıcıları tarafından konum hesaplamakta kullanılan mesafe tayin sinyalleri iletir. Bu sinyaller iki farklı frekansta iletilirler: 1575.42 MHz Link 1 (L1) ve 1227.6 MHz Link 2 (L2). L1 ve L2 mesafe tayin kodları taşırlar. Bu kodlar 1.023 MHz coarse/acquisition kodu (C/A-kodu) ve 10.23 MHz precision kodu (P-kodu) olarak bilinir. L1 her iki kodu da taşırken, L2 sadece P-kodunu taşır. Bu kodlardan herhangi biri mesafe ölçümü için kullanılabilir, ancak normalde P-kodu sadece yetkili kullanıcılar tarafından kullanılmak üzere şifrelenmiştir. Şifrelenmiş P-koduna Y-kodu adı verilir. C/A-kodu ve P-kodunun her ikisine de 50 Hz bir seyir mesajı eklenmiştir.

3.2 Kontrol Bölümü

Kontrol bölümü, bir ana kontrol merkezi ile yeryüzünün değişik noktalarında bulunan izleme istasyonları ve antenlerden oluşur. Şekil 3.3, bu merkezleri göstermektedir.



Şekil 3.3 KKS ana kontrol merkezi, gözlem istasyonları ve antenler (U.S. Coast Guard Navigation Center 1996)

Kontrol bölümü sistemin çalışır durumda olmasından sorumludur. Ana kontrol merkezi kontrol bölümünün merkezi işlem birimidir. Uyduların yönetimi ve gözlenmesinden sorumludur. Fonksiyonları arasında uydu manevralarının kontrolü, yedek uydu ekipmanlarının yeniden konfigürasyonu, uydular tarafından yayınlanan seyir mesajlarının düzenli olarak güncellenmesi, uyduların sağlıklı bir şekilde çalışıp çalışmadığının gözlenmesi ve bakım gibi birçok işlem vardır. Gözlem istasyonları pasif bir şekilde görüş alanlarına giren tüm uyduları izleyerek veri toplarlar. Toplanan veriler ana kontrol merkezine iletilir. Ana kontrol merkezinde veriler değerlendirilerek uyduların astronomik takvimleri ve saat parametreleri hesaplanıp tahmin edilir. Antenler aracılığıyla hesaplanan bu bilgiler periyodik olarak uydulara yüklenir ve uydu seyir mesajlarında yayınlanması sağlanır.

3.3 Kullanıcı Bölümü

Kullanıcı bölümü KKS uydularından yayınlanan sinyalleri alıp çözümleyerek işlemek için özel olarak tasarlanmış alıcılardan oluşur. Alıcıların tasarımı, uygulama alanı ve kullanım amaçlarına göre pek çok farklılıklar gösterir.

3.4 KKS Servisleri

KKS tarafından sağlanan iki tür servis bulunmaktadır (U.S. Coast Guard Navigation Center 1996, EUROCONTROL and IfEN 1998). Hassas Konumlama Servisi (HKS) ve Standart Konumlama Servisi (SKS).

HKS servisi yalnızca yetkili kullanıcılar içindir. Yetkili kullanıcıların kimler olacağı A.B.D. Savunma Bakanlığı tarafından belirlenir. Bu servise erişim kriptografik yöntemlerle kontrol edilir. Bu yöntemlerden birincisi uydulardan yayınlanan sinyallere hata sinyallerinin ilave edilmesi, ikincisi ise P-kodunun şifrelenmesidir. Yetkili kullanıcılara hata sinyallerinin etkisini yok edecek yöntemler ve şifreleme anahtarları verilir. Böylece, sadece yetkili kullanıcıların bu servisten faydalanması sağlanır.

SKS, HKS' ye göre daha az hassas bir servistir. Herkesin kullanımına açık olan bu servis öncelikli olarak sivil amaçlar için düşünülmüştür.

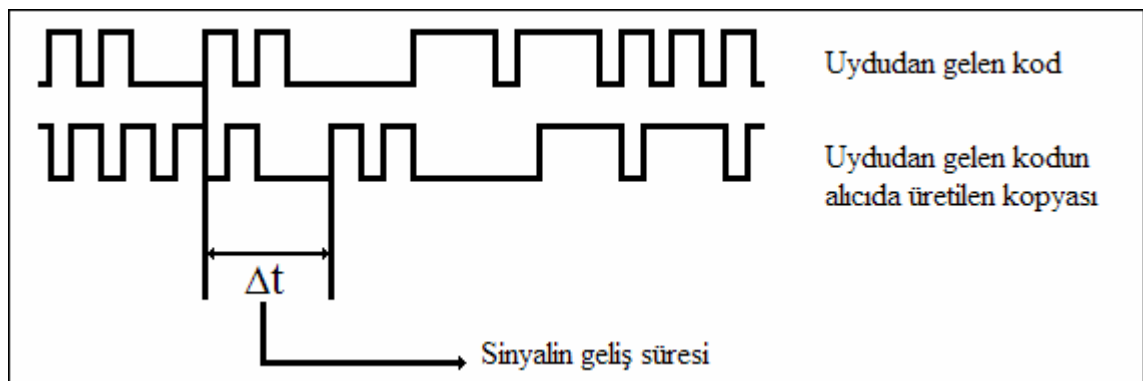
3.5 KKS Çalışma Prensibi

Sistemin çalışma prensibi kabaca şu şekilde özetlenebilir: Alıcı uydudan gelen mesafe kodlarını kullanarak sinyalin geliş süresini belirler. Sinyaller ışık hızında yayımlandığından bu süre ışık hızıyla çarpılarak uydu ile alıcı arasındaki mesafe bulunabilir. Alıcının üç boyutlu konumu için üç bilinmeyen olduğundan üç farklı uydudan gelen sinyallerle üç farklı ölçümün yapılması gerekir. Sinyallerin geliş süresinin ölçümündeki çok küçük hatalar dahi hesaplanan konumda büyük hataların oluşmasına sebep olacaktır. Örneğin sürenin ölçümündeki 1ms hata, konumda yaklaşık 300 km'lik bir hataya karşılık gelir. Bu yüzden dördüncü bir ölçüm yapılır. Hatanın da bilinmeyen olarak toplam dört bilinmeyen bu dört ölçümle hesaplanır. Böylece konumun doğru bir şekilde hesaplanabilmesi için en az dört farklı uydunun görülebilir olması gerekmektedir (U.S. Coast Guard Navigation Center 1996, EUROCONTROL and IfEN 1998).

Alıcının, ölçülen mesafelerden kendi konumunu hesaplayabilmesi için uyduların sinyallerin gönderildiği andaki konumunun da bilinmesi gerekir. Bu bilgiler, seyir mesajında bulunmaktadır.

3.5.1 Sinyallerin geliş sürelerinin ölçümü

Uydudan gelen sinyallerin geliş süresinin ölçümünde mesafe tayin kodları kullanılır. Mesafe tayin kodu belli bir 1 0 bit dizisinden oluşan periyodik bir sinyaldir. Alıcı bu kodun bir kopyasını oluşturur, ve uydudan gelen kodla karşılaştırır. Her iki kod senkron olacak şekilde kodun kopyasını zaman içinde kaydırır. Aradaki fark sinyalin geliş süresini verir (U.S. Coast Guard Navigation Center 1996, EUROCONTROL and IfEN 1998). Şekil 3.4, sinyallerin geliş sürelerinin ölçümünü göstermektedir.



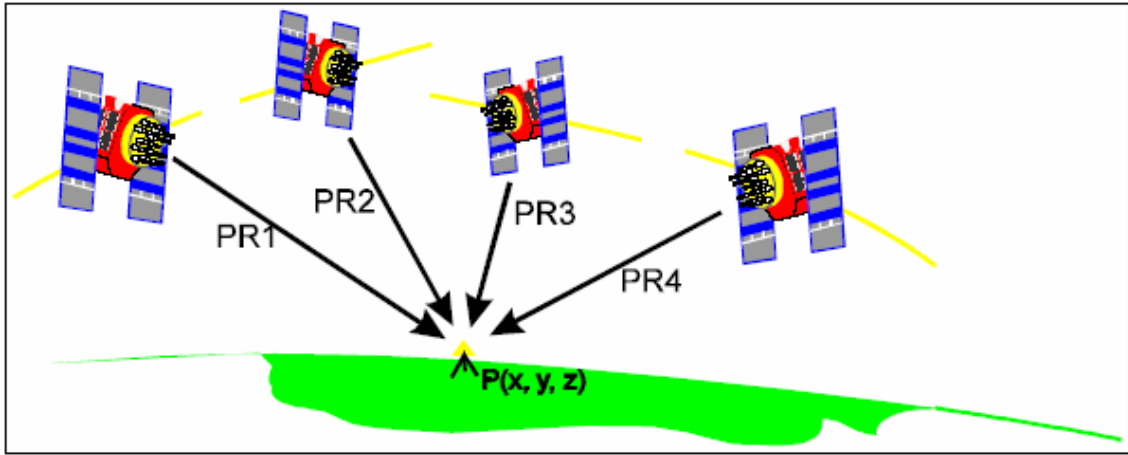
Şekil 3.4 Uydulardan gelen sinyallerin geliş süresinin ölçümü

3.5.2 Konumun hesaplanması

Şekil 3.5 KKS ile konum hesaplanmasını göstermektedir. Bunun için dört farklı uyduya olan mesafenin ölçülmesi gerekir. Her uyduya olan mesafe,

$$PR_i = c * \Delta t_i$$

eşitliği ile hesaplanır. Eşitlikte c , ışık hızını, PR_i ve Δt_i $i = 1,2,3,4$ ise uyduya olan uzaklık ve söz konusu uydudan gelen sinyalin geliş süresini göstermektedir.



Şekil 3.5 KKS ile konum hesaplanması (EUROCONTROL and IfEN 1998).

Mesafe ölçümünde oluşan hata dT ile, uydü konumları X_i , Y_i ve Z_i ile gösterilirse alıcının üç boyutlu konumu X, Y ve Z , dördüncü bilinmeyen dT ile birlikte aşağıdaki denklem sisteminden hesaplanabilir (EUROCONTROL and IfEN 1998).

$$\begin{aligned} (X_1 - X)^2 + (Y_1 - Y)^2 + (Z_1 - Z)^2 + dT &= PR_1^2 \\ (X_2 - X)^2 + (Y_2 - Y)^2 + (Z_2 - Z)^2 + dT &= PR_2^2 \\ (X_3 - X)^2 + (Y_3 - Y)^2 + (Z_3 - Z)^2 + dT &= PR_3^2 \\ (X_4 - X)^2 + (Y_4 - Y)^2 + (Z_4 - Z)^2 + dT &= PR_4^2 \end{aligned}$$

3.6 KKS Verileri

KKS alıcıları, veri iletimi için standart NMEA protokolünü kullanırlar. Bununla birlikte, bazı KKS alıcılarında kendilerine özel protokoller de kullanılabilir. Alıcılar, konum bilgilerini NMEA cümleleri şeklinde iletirler. Aslında, alıcıdan gelen veriler birkaç çeşit cümleden oluşmaktadır. Bu cümlelerin bazıları hız ve yükseklik, bazıları düzeltme verisi, bazıları da uydularla ilgili bilgiler vb. bilgiler içerir. En basit

olanı, sadece enlem, boylam, yükseklik, konum hesaplandığı zaman ve verinin geçerli olup olmadığının anlaşılmasını sağlayan bilgilerden oluşur. Her cümle “\$” karakteri ile başlar, daha sonraki iki karakter veriyi gönderen aygıtı, sonraki üç karakter ise cümle türünü belirtir. Bundan sonra bilgiler virgülle ayrılmış bir şekilde sıralanırlar. Cümle sonunda “*” karakteri ve iki baytlık hata denetleme işleme kullanılan doğrulama (checksum) verisi bulunur (O’Sullivan ve Igoe 2004).

Konum bilgilerini içeren en basit cümle GLL cümlesidir. Bu cümle sadece enlem, boylam ve konumun hesaplandığı zaman bilgilerinden oluşur. Bu bilgiler şu şekildedir:

- Derece ve dakika cinsinden enlem
- Enlem yönü (N:Kuzey, S:Güney)
- Derece ve dakika cinsinden boylam
- Boylam yönü (E:Doğu, W:Batı)
- Greenwich zamanına göre saat, dakika ve saniye cinsinden konumun belirlendiği zaman
- Verilerin geçerliliği
- Doğrulama (Checksum) verisi.

Aşağıda görülen örnek GLL cümlesinde,

\$GPGLL, 3736.2249,N, 02907.5900,E,185310,V,S*54

enlem 37 derece 36,2249 dakika kuzey, boylam 29 derece 7,59 dakika doğu, zaman ise Greenwich zamanına göre 18:53:10 olarak görülmektedir.

4. SAYISAL HARİTALAR

Araç takip sistemlerinde araç konumları sayısal haritalar üzerinde izlendiğinden bu haritaların görüntülenmesi sistemin çok önemli bir parçasıdır. Bu tez çalışmasının en önemli kısmını sayısal haritaların bilgisayar ekranı üzerinde görüntülenebilmesi çalışmaları oluşturmuştur.

Bundan bir süre önce insanlar haritaları elle çizerdi. Verilerin analizi ve bunun sonucunda haritaların oluşturulması işlemleri oldukça yavaş ve işgücüne bağlı bir süreçti. Bilgisayarların düşen maliyet ve artan işlem ve depolama kapasiteleri ile sayısal haritalar yeni olanaklar sağladılar. Artık bilgisayarların yardımı ile çok kısa bir süre içinde, birkaç fare tıklaması ya da bir dizi kod ile, veriler analiz edilebilmekte, haritalar çizilebilmekte ve renklendirilebilmektedir (Mitchell 2005).

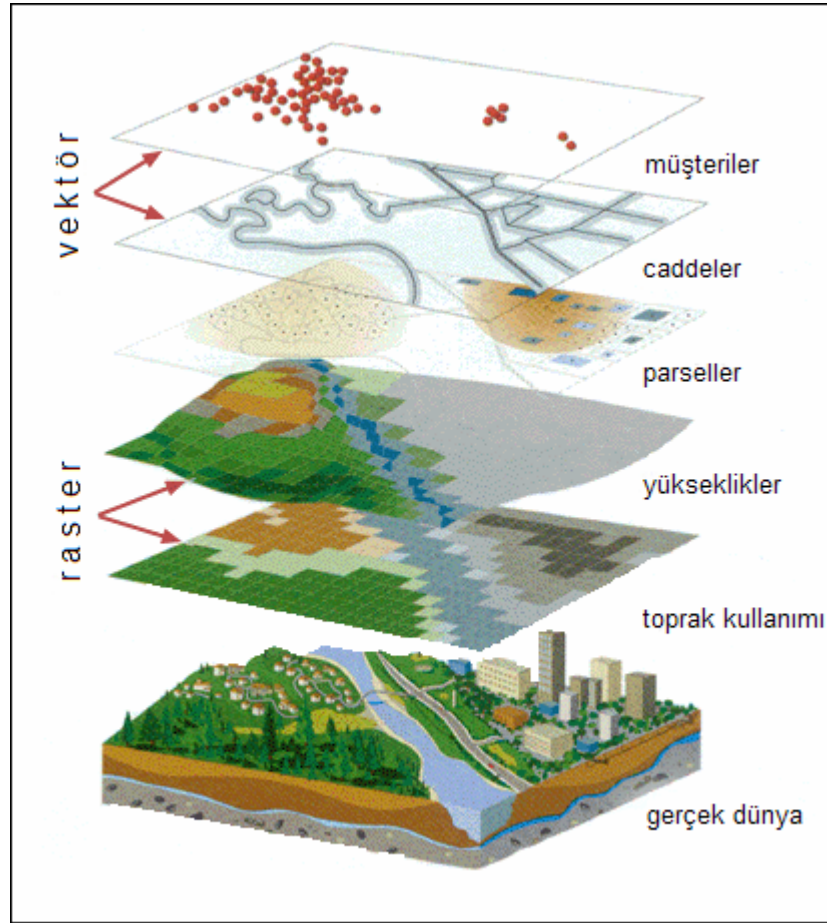
Sayısal haritalar kağıt haritaların elektronik aygıtlar (digitizerlar, tarayıcılar) yardımıyla sayısal forma dönüştürülmeleri ile elde edilebileceği gibi CBS yazılımları ile de üretilebilirler. CBS ile ilgili, farklı açılar veya farklı disiplinlerce geliştirilen pek çok tanım (Chrisman 2002) olmakla birlikte, yeryüzüne ait bilgilerin toplanması, depolanması, analiz edilmesi ve yayınlanması için yazılım, donanım, veri, insanlar, organizasyonlar ve kurumsal düzenlemelerden oluşan bir sistem (Dueker ve Kjerne 1989) olarak tanımlanır.

Bu bölümde sayısal haritalar ve sayısal haritaların bazı önemli noktalarına değinilecektir.

4.1 Harita Katmanları

Sayısal haritalar katmanlar şeklinde organize edilir (Escobar vd 2001). Her katman verilerin belirli bir bölümünü ele alır. Örneğin bir katman nehirlerden, bir başka katman göllerden, bir diğeri anayollardan oluşur. Bu katmanlar belli bir sıra ile birbirleri üzerinde görüntülenerek tüm harita elde edilir. Görüntülenmesine gerek olmayan katmanlar çıkarılıp, istenildiğinde yeni katmanlar da eklenebilir. Şekil 4.1'de dört katmanlı bir sayısal harita örneği görülmektedir. Bu teknik, köklerini bilgisayarlardan

önce, saydam haritaların birbirleri üzerine bindirilerek görüntülediği (McHarg 1969) fotoğrafik yöntemlerden alır.



Şekil 4.1 Harita katmanları (Escobar vd 2001).

4.2 Konum Bilgilerinin Modellenmesi

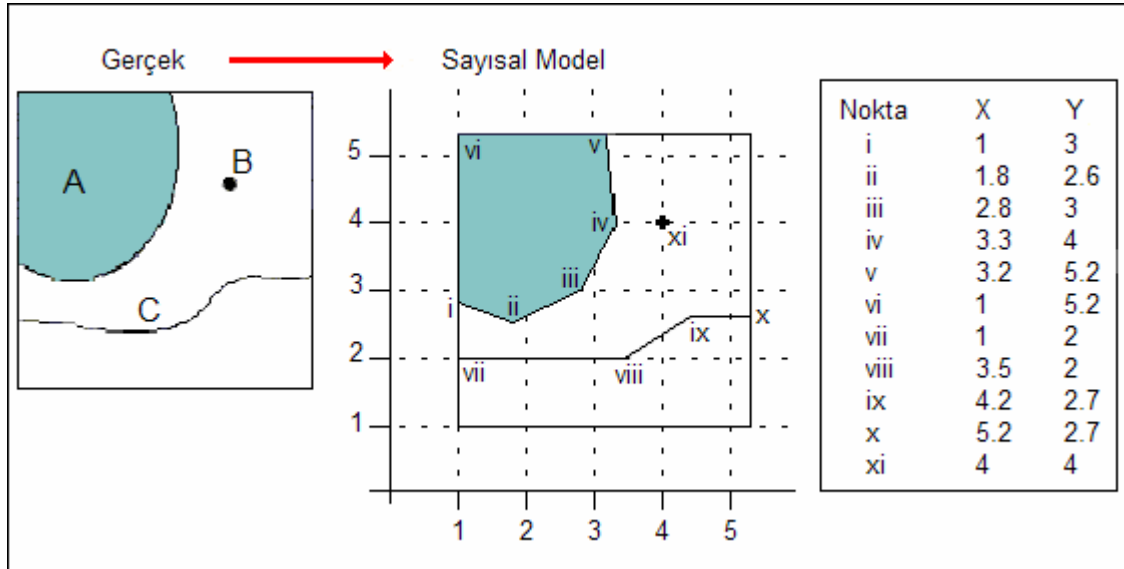
Konum bilgileri koordinatlar, yani belli bir referans sistemine göre yapılan sıralı ölçümler şeklinde gösterilir. Bu ölçümler bir elipsoit üzerindeki açılar (enlem, boylam) olabileceği gibi bir düzlem üzerinde seçilen eksenlere olan uzaklıklar da olabilir (Chrisman 2002). Koordinatlar, her zaman bilgisayarda sayı çiftleri şeklinde modellenir (Chrisman 1984). Bunun bazı olumsuz sonuçları vardır çünkü bilgisayarda bulunan sayı tipleri (integer, floating point, double precision) ile çözünürlük kısıtlanmaktadır. Örneğin, diyagonal çizgileri oluşturan tüm ara noktalar modellenemezler. Bunları bükmeden çizmenin bir yolu yoktur.

Haritaların sayısal gösteriminde kullanılan iki model vardır: Vektör model ve Raster model.

4.2.1 Vektör model

Bu model analitik geometriye dayanır. Nehirler, göller, caddeler, yollar, il sınırları, kent merkezleri gibi karmaşık coğrafik şekiller üç basit geometrik şekil kullanılarak oluşturulur: Nokta, çizgi ve poligon (Şekil 4.2).

- Nokta: Tek bir X-Y koordinat çiftinden oluşan soyut bir şekildir. Çizgi veya alan olarak görüntülenemeyecek kadar küçük olan coğrafik şekiller için kullanılır. Haritanın ölçeğine bağlı olarak bu bir bina, bir köy ya da bir il merkezi olabilir.
- Çizgi: Sıralı koordinat çiftlerinden oluşur. Alan olarak görüntülenemeyecek kadar ince olan coğrafik şekiller için kullanılır. Ölçeğe bağlı olarak, bir nehir, cadde veya yollar çizgilerle modellenebilir.
- Poligon: Alanların gösteriminde kullanılan şekildir. Bir poligon, sınırlarını oluşturan çizgiler ve bu çizgilerin içinde kalan bir noktayla tanımlanır.



Şekil 4.2 Vektör model (Escobar vd 2001).

4.2.2 Raster model

Bilgisayarın grafik donanımına oldukça yakınlık gösteren bu modelde görüntülenecek coğrafi alan, Şekil 4.3'te görüldüğü gibi, satır ve sütunlara bölünerek bir matris formu elde edilir. Bu hücrelere piksel denir ve ölçülen bir değer içerir. Konum bilgisi ise hücrenin matris içindeki sırasından anlaşılır.

raster gösterim								piksel değeri	
A	A	A	A	0	0	0	0	1	A
A	A	A	A	A	0	0	0	2	A
A	A	A	A	0	B	0	0	3	A
A	A	A	A	0	0	0	0	4	A
A	A	A	A	A	A	C	C	5	0
A	A	A	A	A	C	0	0	6	0
C	C	C	C	C	0	0	0	7	0
0	0	0	0	0	0	0	0	8	0
0	0	0	0	0	0	0	0	9	A
0	0	0	0	0	0	0	0	10	A
0	0	0	0	0	0	0	0	11	A
0	0	0	0	0	0	0	0	12	A
0	0	0	0	0	0	0	0	13	A
0	0	0	0	0	0	0	0	14	0
0	0	0	0	0	0	0	0	15	0
0	0	0	0	0	0	0	0	16	0
0	0	0	0	0	0	0	0	.	.
0	0	0	0	0	0	0	0	.	.
0	0	0	0	0	0	0	0	62	0
0	0	0	0	0	0	0	0	63	0
0	0	0	0	0	0	0	0	64	0

Şekil 4.3 Raster model (Escobar vd 2001)

Raster model bilgisayarda doğrudan diziler (array) ile gerçekleştirilir. Çözünürlüğe bağlı olarak matrisin hücre sayısı değişir. Yüksek çözünürlüklerde matris çok fazla hücreden oluşur ve bitişik hücrelerin çoğu aynı değeri içerir. Bu durumda bazı sıkıştırma teknikleri kullanılır. Bunlardan biri, özellikle aynı değeri içeren bitişik hücreler çoğunlukta ise önemli miktarda sıkıştırma olanağı veren, Run-Length Encoding adı verilen yöntemdir (Şekil 4.4). Bu yöntemde, her hücre ayrı ayrı depolanmak yerine, hücrenin içerdiği değer ve satır boyunca bu aynı değere sahip hücrelerin sayısı gösterilir (Chrisman 2002).

raster gösterim							
A	A	A	A	0	0	0	0
A	A	A	A	A	0	0	0
A	A	A	A	0	B	0	0
A	A	A	A	0	0	0	0
A	A	A	A	A	A	C	C
A	A	A	A	A	C	0	0
C	C	C	C	C	0	0	0
0	0	0	0	0	0	0	0

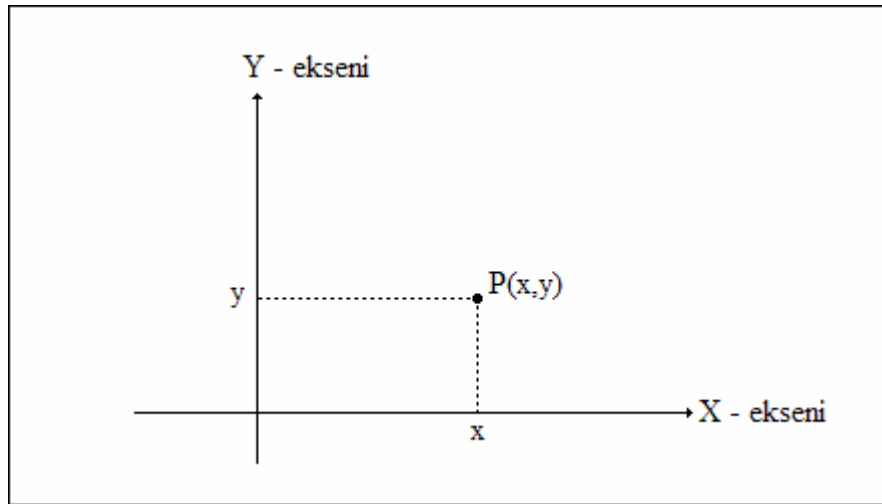
A,4	0,8		
A,5	0,8		
A,4	0,5	B,6	0,8
A,4	0,8		
A,3	0,8	C,8	
0,6	C,6	0,8	
C,6	0,8		
0,8			

Şekil 4.4 Run-length encoding (Escobar vd 2001)

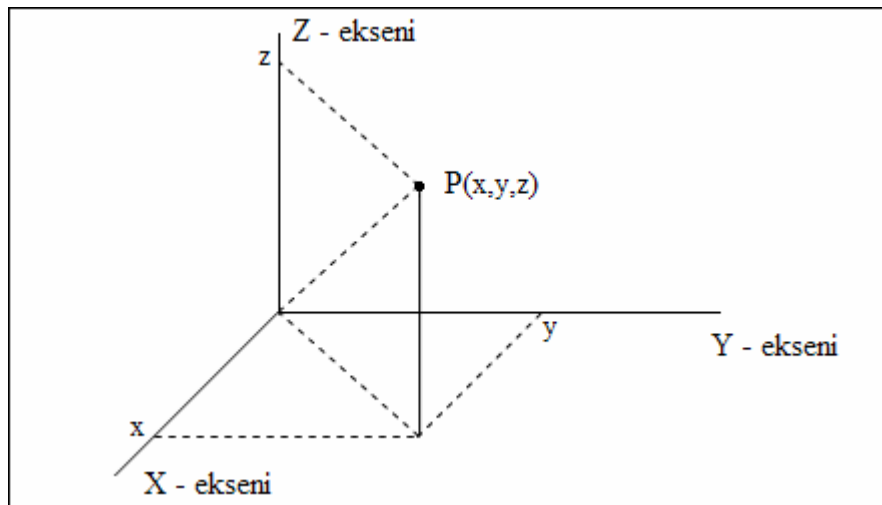
4.3 Koordinat Sistemleri

Uzayda bir noktanın konumu koordinat adı verilen sayılarla belirtilir. Koordinatlar, belli bir sistem referans alınarak, bu sisteme göre yapılan ölçümlerle belirlenir. Kullanıldığı yer ve amaca göre, yerel ve küresel pek çok sistem olmasına rağmen temelde kartezyen ve kutupsal olmak üzere iki sınıfa ayrılabilirler.

Kartezyen koordinat sistemleri birbirleri ile dik açı yapacak şekilde kesişen iki (iki boyutlu sistem, Şekil. 4.5) ya da üç (üç boyutlu sistem, Şekil 4.6) doğrudan oluşan Decartes tarafından geliştirilen sistemdir (Dana 1999). Herhangi bir noktanın yeri bu referans doğrulara olan uzaklık ile belirlenir.

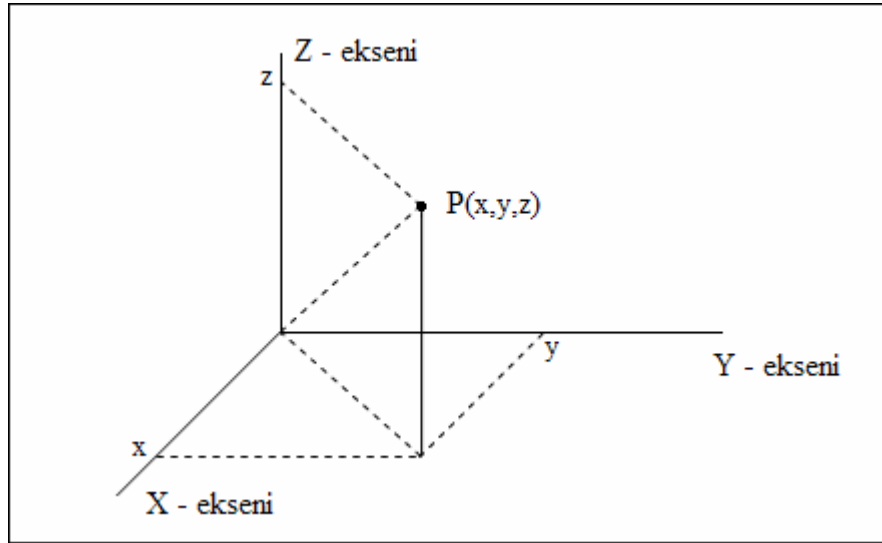


Şekil 4.5 İki boyutlu kartezyen koordinat sistemi.



Şekil 4.6 Üç boyutlu kartezyen koordinat sistemi.

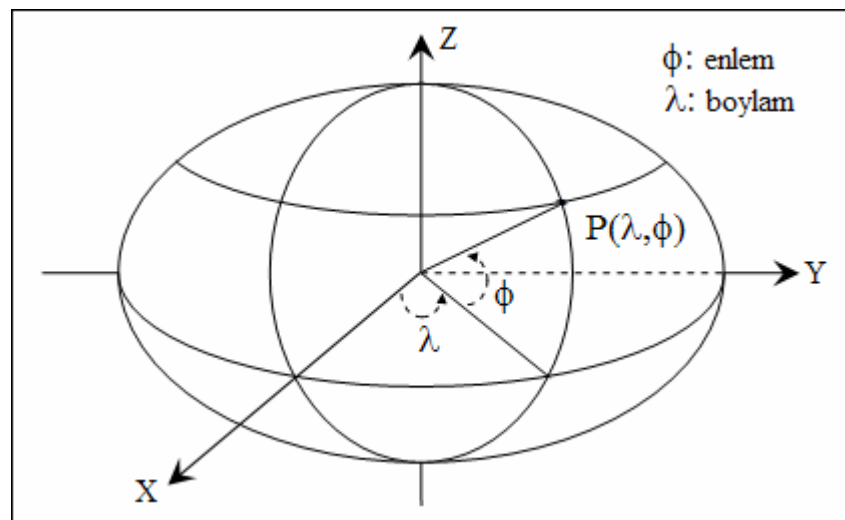
Kutupsal sistemlerde ise noktanın yeri bu referans doğrular ile yapılan açılarla belirlenir (Şekil 4.7).



Şekil 4.7 Üç boyutlu kutupsal koordinat sistemi.

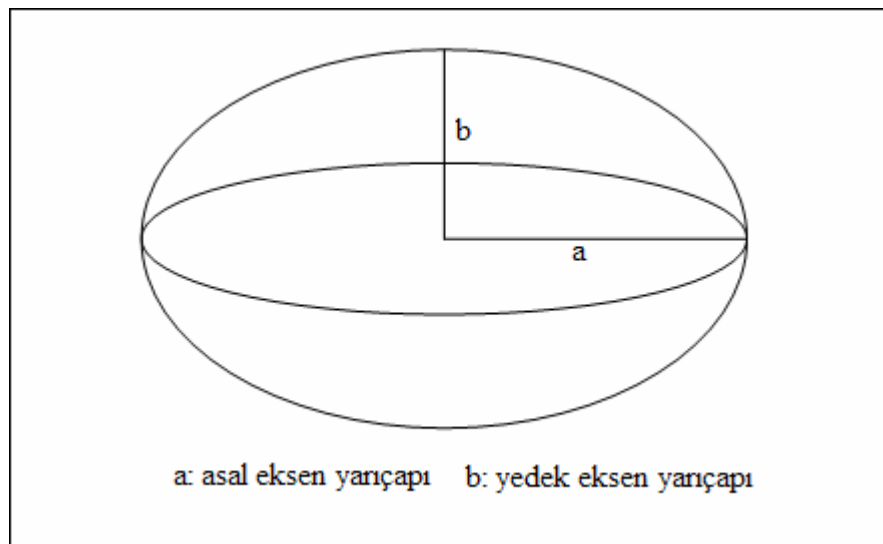
4.4 Coğrafik Koordinat Sistemi ve Referans Elipsoitler

Yeryüzündeki bir noktanın koordinatları için kullanılan koordinat sistemi coğrafik koordinat sistemi olarak bilinir. Bu sistemde yeryüzünün herhangi bir noktasının koordinatları her zaman enlem ve boylam şeklinde belirlenir. Sistem referans olarak başlangıç meridyeni ile ekvator düzlemlerini alır, bir noktanın enlem ve boylamının tanımlanabilmesi için referans olarak seçilmişlerdir. Enlem, noktanın ekvator düzlemi ile, boylam ise başlangıç meridyeni ile yaptığı açı olarak tanımlanır (Şekil 4.8).



Şekil 4.8 Coğrafik koordinat sistemi

Enlem ve boylam özel bir yeryüzü modeli ile bağıntılıdır. Çok kesinlik gerektirmeyen çalışmalarda yeryüzü genellikle bir küre ile modellenir. Fakat daha hassas sonuçların gerektiği durumlar için elipsoit kullanılır (WEB_1 1997). Elipsoit, yeryüzünü çok daha gerçeğe yakın şekilde modelleyebilen, bir elipsin yedek ekseninde çevrilmesi ile oluşan üç boyutlu geometrik şekildir. Her elipsoit, asal ve yedek eksen uzunluklarından oluşan iki temel parametre ile belirlenir (Dana 1999). Şekil 4.9'da örnek bir elipsoit ve parametreleri görülmektedir. Elipsoidin diğer tüm parametreleri bu iki parametre kullanılarak bulunabilir. Bazen yeryüzünün sadece belirli bir bölümünü en uygun şekilde modelleyebilecek elipsoitlere gerek olmaktadır. Bu nedenle birçok ülke ve kuruluş kendi amaçlarına göre değişik elipsoitler geliştirmişlerdir. Enlem ve boylam , bu elipsoitlerden birini referans aldığından bunlara referans elipsoitler adı verilir.

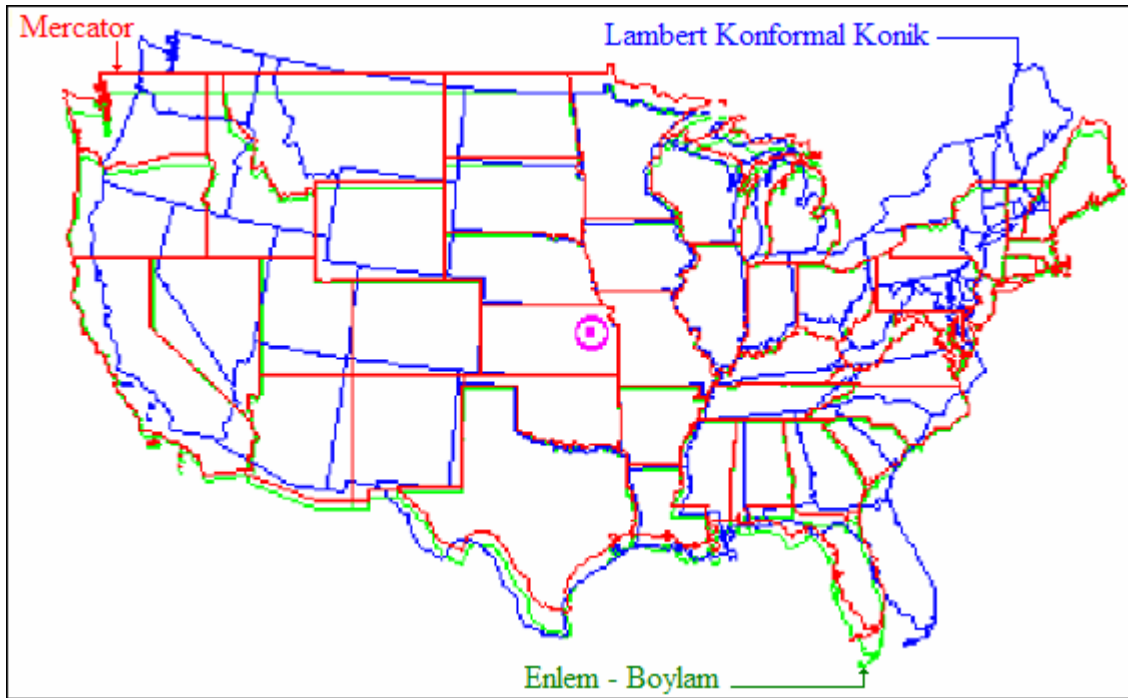


Şekil 4.9 Elipsoit ve parametreleri

Yeryüzünün küresel kullanımı için bir modelinin oluşturulması için de eskiden beri süregelen pek çok çalışma yapılmaktadır. Geliştirilen modeller, zaman içinde daha iyi sonuç veren modellerle değiştirilmektedir. Şu an kullanımda olan en son model World Geodetic System adı verilen ve çalışmalarını 1984 yılında tamamlanan modeldir (WGS 84). WGS 84 önceki benzer çalışmaların (WGS 60, WGS 66 ve WGS 72) sonucusudur ve parametreleri şu şekildedir: Yedek eksen yarıçapı $b = 6356752.3142$ m, asal eksen yarıçapı $a = 6378137.0$ m (U.S. Coast Guard Navigation Center 1996, EUROCONTROL ve IfEN 1998).

4.5 Harita Projeksiyonları

Coğrafik koordinat sistemi, konumun küre benzeri bir şekil üzerinde hassas bir şekilde belirlenebilmesi için geliştirilmiş bir sistemdir. Ancak kağıt haritalar ya da bilgisayar ekranları küresel yüzeyler değildirler. Bu koordinatların bir düzlem üzerine yansıtılması gerekir (Mitchell 2005). Bu işlem projeksiyon olarak bilinir. Bu işlemde kaçınılmaz olarak birtakım hatalar ortaya çıkar. Çünkü, üç boyutlu küresel bir yüzeyin iki boyutlu bir yüzeye aktarılması sırasında bazı bozulmalar doğal olarak olacaktır. Oluşacak haritanın büyüklüğü ve kullanım amacına göre çok farklı projeksiyon yöntemleri vardır. Bunların her biri birbirlerinden farklı sonuçlar verirler. Şekil 4.10 farklı projeksiyonların harita üzerindeki etkisini göstermektedir.

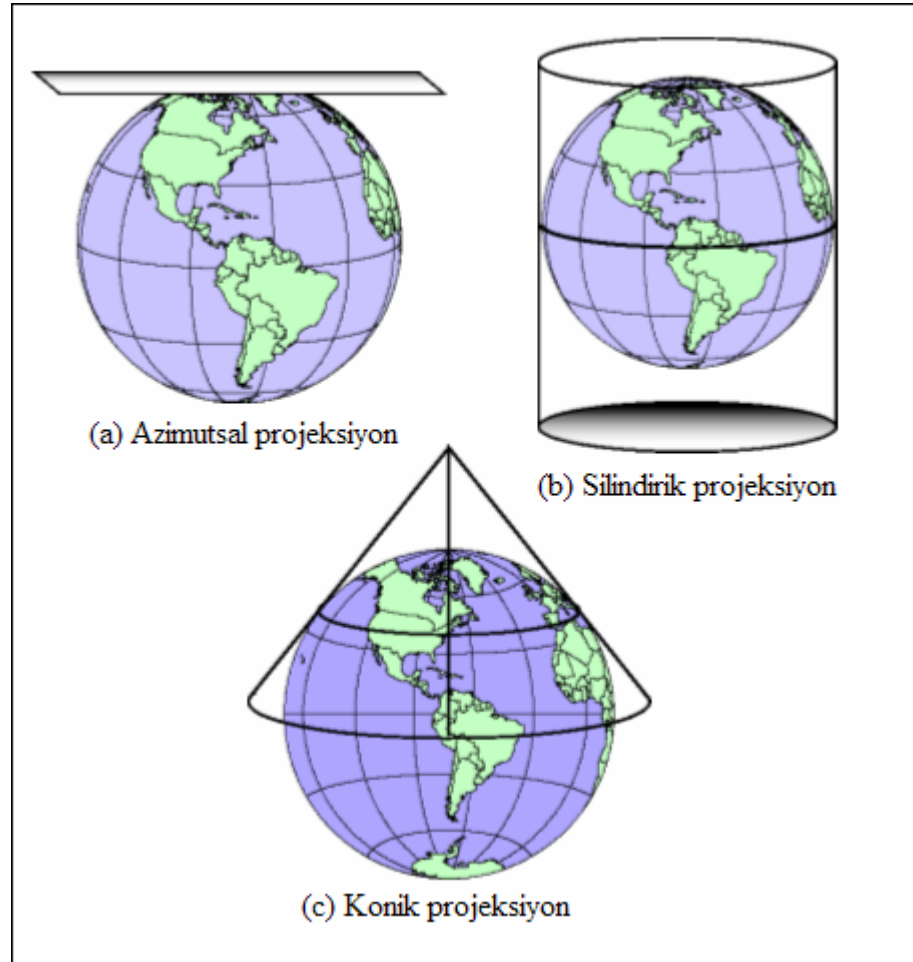


Şekil 4.10 Projeksiyonun harita üzerindeki etkisi (Dana 2000).

Temel olarak dört yöntem vardır (Dana 2000):

- Silindirik Projeksiyonlar: Küresel bir yüzeyin bir silindir üzerine izdüşümünün alınmasıyla oluşan projeksiyonlardır (Şekil 4.11a).
- Konik Projeksiyonlar: Küresel bir yüzeyin bir koni üzerine izdüşümünün alınmasıyla oluşan projeksiyonlardır (Şekil 4.11b).

- Azimutsal Projeksiyonlar: Küresel bir yüzeyin bir düzlem üzerine izdüşümünün alınmasıyla oluşan projeksiyonlardır (Şekil 4.11c).
- Diğerleri: Bu üç sınıfa girmeyen farklı yöntemlerle oluşturulan projeksiyonlardır.



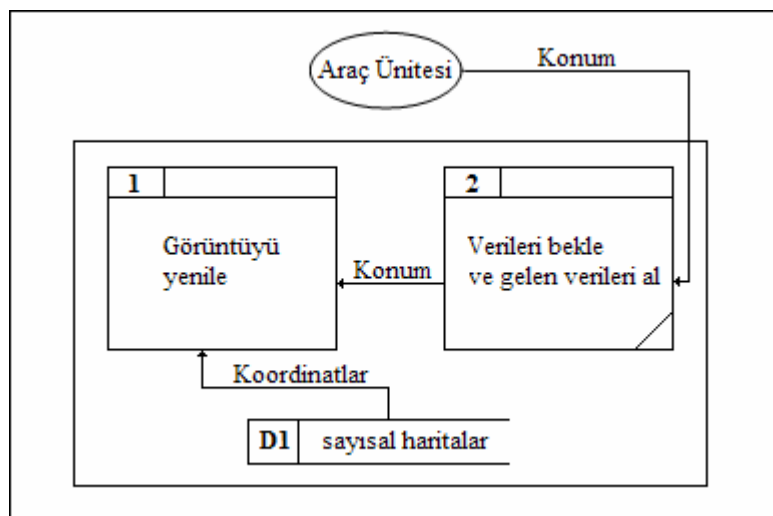
Şekil 4.11 Harita projeksiyonları

5. ATS YAZILIMI GELİŞTİRİLMESİ

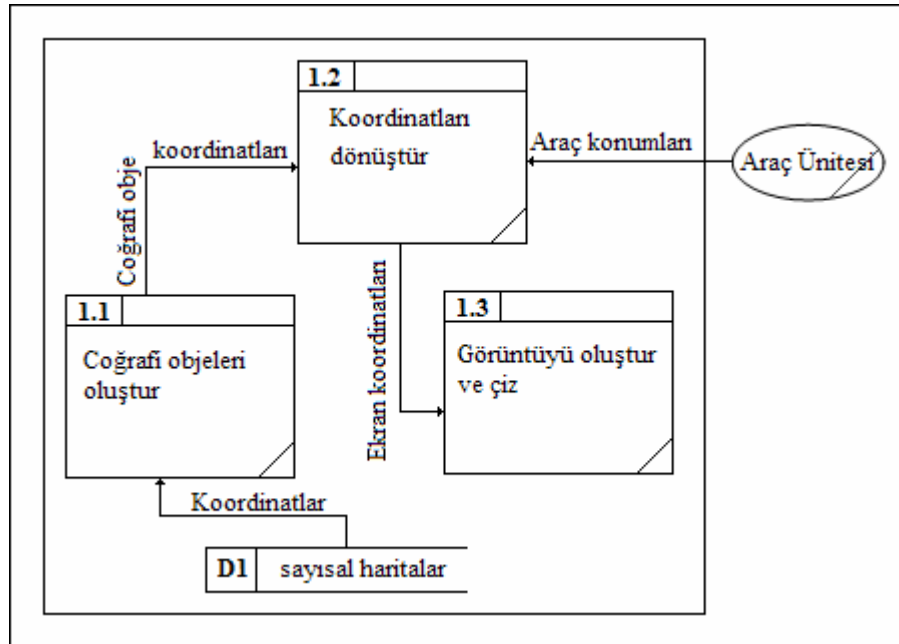
Bu bölümde araçların bilgisayar ekranı üzerinde izlenebilmesi için geliştirilen yazılımdan söz edilecektir. Yazılım, MapInfo gst ve mif formatlarında hazırlanmış sayısal haritaları görüntüleyip, araçtan internet aracılığıyla gelen konum bilgilerini görüntülenen sayısal harita üzerinde işleyerek, aracın bilgisayar ekranı üzerinde izlenebilmesini sağlamaktadır. Takip eden kısımlarda, sistemin genel bir modeli ile birlikte çalışma biçimi ayrıntılarıyla açıklanacak, ardından kullanılan sınıflara ait bilgilere yer verilecektir. Sayısal haritaların ve haritaya ait bazı detayların görüntülenmesinde kullanılan teknikler ve karşılaşılan sorunlar, çözümleri ile birlikte verildikten sonra mevcut ticari yazılımlar ile karşılaştırılacak ve sistemin test edilmesi için gerçekleştirilen örnek uygulamadan söz edilecektir.

5.1 İzleme Yazılımı

İzleme yazılımının modeli, birinci ve ikinci seviye veri akış diyagramları yardımıyla Şekil 5.1 ve 5.2’de verilmiştir. Yazılım iki bölümden oluşmaktadır: Temel olarak haritaların görüntülenmesinden sorumlu bir ana bölüm ve internet üzerinden gelecek konum bilgilerini bekleyen bir dinleyici bölümü. Dinleyici geri planda çalışmakta ve konum bilgilerini her alışında, bu bilgileri ana bölüme ileterek ekran görüntüsünün yeni durumu yansıtacak şekilde yenilenmesini sağlamaktadır.



Şekil 5.1 Sistemin birinci seviye veri akış diyagramı



Şekil 5.2 Sistemin ikinci seviye veri akış diyagramı

Sayısal haritaların görüntülenmesi pek çok işlemi de beraberinde getirmektedir. Bu işlemler izleme yazılımının ana bölümü tarafından gerçekleştirilir. Öncelikle, görüntülenecek haritayı oluşturan dosyalar okunarak, haritayı oluşturan katmanlar ve her katmanı oluşturan coğrafi objeler, belleğe yerleştirilir. Sayısal haritanın, daha önce de belirtildiği gibi, MapInfo gst ve mif formatlarında hazırlanmış olması gerekir. Bu iki tip dosya birlikte bir sayısal harita oluşturur. Haritanın görüntülenmesine ait temel bilgiler ve haritayı oluşturan katmanlar gst dosyasında bulunmaktadır. Bu dosyalara coğrafya seti (geoset) ismi verilmektedir. Seti oluşturan her katman ise ayrı bir mif dosyasıdır. Bu dosya katmanı oluşturan coğrafi objeler ile ilgili bilgileri içerir. Haritanın görüntülenebilmesi için söz konusu tüm dosyalar aynı klasörde bulunmalıdır.

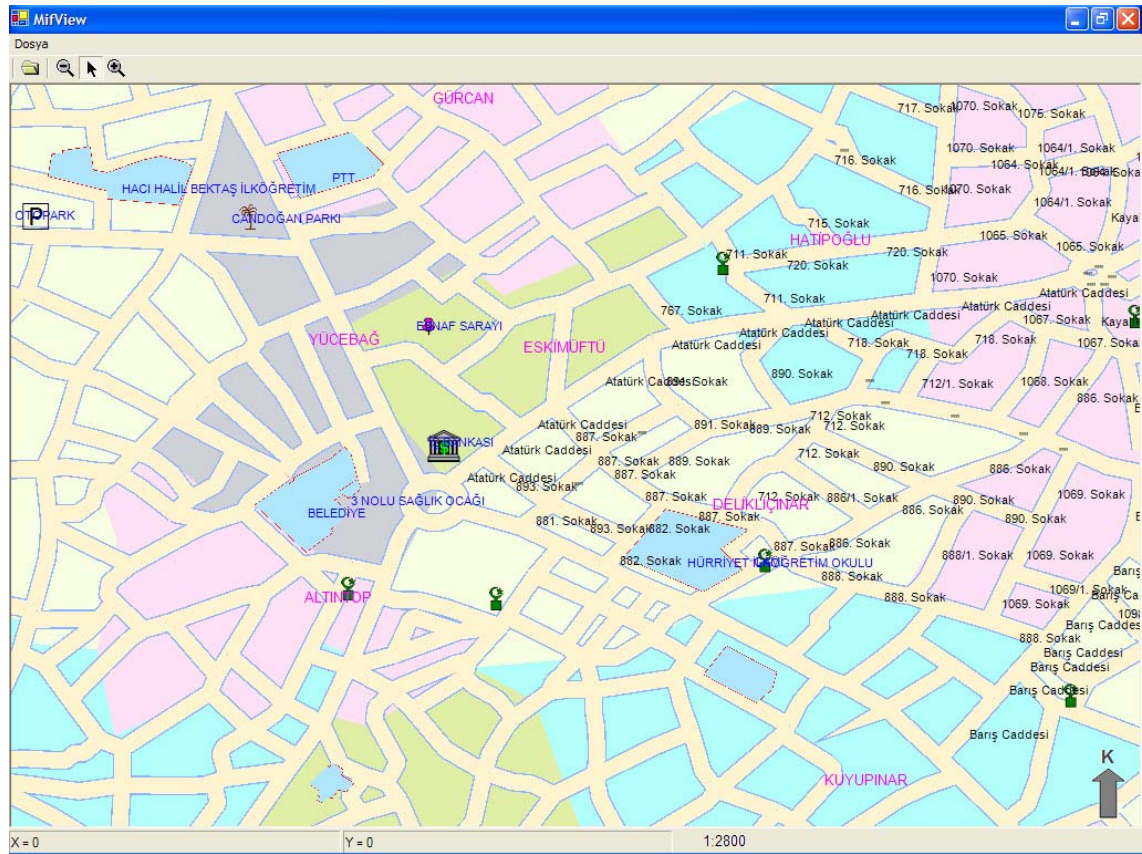
Görüntülenecek harita formatı olarak gst ve mif formatlarının seçilmesi çeşitli nedenlere dayanmaktadır:

- MapInfo yaygın olarak kullanılan bir CBS yazılımıdır. Bu yazılım kullanılarak hazırlanmış bir çok sayısal bulunmaktadır.
- Bu dosyalar anlaşılması kolay metin dosyalarıdır.
- Bu dosyaların yapıları MapInfo tarafından açıklanmıştır.

Söz konusu dosyaların yapıları EK1 ve EK2’de ayrıntılı bir şekilde verilmiştir.

Öncelikle gst dosyası okunarak haritanın temel bilgileri ve haritaya ait katmanları oluşturan mif dosyalarının isimleri elde edilir. Daha sonra her katman ilgili dosyadan okunarak belleğe alınır.

Görüntüleme işlemi her katmanın dosyada belirtilen sırada birbirleri üzerine çizilmesiyle gerçekleşir. Çizim sırasında, çizilen katmanı oluşturan coğrafi objelerin koordinatlarının, ekranın koordinat sistemlerine göre yeniden hesaplanması gerekmektedir. Yani bir koordinat dönüşümü yapmak gerekmektedir. Çünkü ekran ve harita farklı koordinat sistemlerine sahiptirler. Çizim işlemi bittikten sonra harita ekranda görüntülenir. Şekil 5.3 geliştirilen yazılımın kullanıcı arabirimini göstermektedir. Şekilde Denizli ili için yapılmış bir sayısal haritadan Delikliçınar meydanı ve civarı görüntülenmektedir.



Şekil 5.3 Yazılımın kullanıcı arabirimi

Kullanıcı haritayı arzu ettiği zoom seviyelerinde görüntüleyebilir. Zoom seviyesi, haritanın görüntülediği alanın, haritadaki uzunluk birimi cinsinden yatay uzaklıktır. Zoom seviyesi haritada görüntülenecek ayrıntı miktarını belirler, seviye ne kadar küçükse ayrıntı miktarı o kadar çok olur. Ayrıca, kullanıcı haritanın değişik bölümlerini

de görüntüleyebilir. Bu işlem haritanın istenen miktarda sağa, sola, yukarı ya da aşağı kaydırılmasıyla gerçekleşir. Kullanıcının tüm bu isteklerini yerine getirmek gene ana bölümün görevidir. Böyle bir istek durumunda ana bölüm, ekranın koordinat sistemini yeniden ayarlar, coğrafi objeleri bu yeni ekran koordinat sistemine göre dönüştürür ve görüntüyü yeniler.

Yazılımın geliştirilen dinleyici bölümü ise arka planda çalışarak internet üzerinden gelecek konum bilgilerini beklemektedir. Bu işlem kısaca dinleme olarak bilinir. Konum bilgileri internet üzerinden geleceği için yazılımın çalıştığı bilgisayarın internet erişimine ve statik bir IP numarasına sahip olması gerekir.

Bilgisayar ağlarında iletişimin gerçekleşmesi birtakım kurallara göre olur. Bu kurallara iletişim kuralları ya da protokol denmektedir. İnternet adı verilen küresel bilgisayar ağı için bu protokol TCP/IP adı verilen protokol takımındır. Bu protokol takımı aslında birçok protokolden oluşur. Bunlardan biri bilgisayarların internet üzerinden birbirleri ile iletişim kurmalarını sağlamak için kullanılan 32-bit işaretsiz sayılara ismini veren IP protokolüdür (Rodriguez vd 2001). Yazılımın çalıştığı bilgisayarın IP numarası araç ünitesinde konum bilgilerinin gönderilmesi için kullanıldığından bu numara değişken olmamalı yani statik bir numara olmalıdır. TCP/IP protokol takımı veri iletimi için iki ayrı protokol içerir: TCP ve UDP. Yazılımın alıcı bölümünde UDP kullanılmıştır. Araç ünitesinden gelen verilerin küçük boyutlu olması, iki yönlü iletişime gerek duyulmaması gibi nedenler bu protokolün kullanımını daha uygun hale getirmektedir.

Konum bilgilerinin gönderileceği bilgisayarın IP numarası veri iletişimi için tek başına yeterli değildir. Çünkü iletişim gerçekte bilgisayarlar arasında değil, bilgisayarlar üzerinde çalışan yazılımlar arasında gerçekleşmektedir. Bu yüzden IP numarasının yanında verilerin gönderileceği yazılımın da belirlenmesi gerekir. Bu işlem port adı verilen 16-bit numaralar arıcılığı ile gerçekleşir (Rodriguez vd 2001). İnternet üzerinden iletişim kurmak isteyen yazılıma bir numara verilir. 0-1023 arasındaki numaralar IANA isimli kuruluş tarafından kontrol edilir. 1024-65535 arasında kalan numaralar ise herhangi bir kuruluş tarafından kontrol edilmezler ve bilgisayar programcıları tarafından geliştirilen sıradan uygulamalara verilebilirler. Bu çalışmada yazılımın alıcı bölümü için 6065 nolu UDP portu belirlenmiştir. Bir başka deyişle alıcı, 6065 nolu UDP portunu dinlemektedir.

Gelen konum bilgileri alıcı tarafından öncelikle ana bölümün kullanabileceği bir formata çevrilir. Sonra, ana bölümü bilgi gelişinden haberdar eder, hazırlanan bilgileri iletir ve yeniden gelecek bilgileri beklemeye başlar. Tüm işlemler süresince hep geri planda çalışır ve ana bölümün çalışmasını hiçbir şekilde kesmez.

Ana bölüm, elde ettiği yeni konum bilgilerini görüntülenmekte olan üzerine işleyerek ekran görüntüsünü yeniler. Böylece araç gerçek zamanlı olarak bilgisayar ekranı üzerinde izlenir.

Araç ünitesi tarafından iletilen veriler sadece konum bilgisinden ibaret değildir. Tarih, saat, hız, GSM modülüne ait IMEI numarası gibi bilgiler de gönderilir. Ayrıca araç ünitesinin özelliklerine göre, ses, görüntü, araç kapılarının açık ya da kapalı olması, yakıt durumu gibi bilgiler de gönderilebilmekte, bu bilgiler birden fazla aracın izlenmesine olanak verdiği gibi, sisteme yeni işlevler de kazandırabilmektedir.

Yazılım, nesneye dayalı görsel bir programlama dili olan Microsoft Visual C# .NET ile geliştirilmiştir. Haritaların görüntülenmesi için en son grafik programlama arabirimi olan GDI+ kullanılmış, internet iletişimi ise Windows soket programlama ile gerçekleştirilmiştir.

5.2 Yazılımda Kullanılan Önemli Sınıflar

Geliştirilen yazılımda, sayısal haritaların görüntülenmesi, konum bilgilerinin elde edilmesi gibi işlemlerin yerine getirilmesi için gerekli birçok işlev ve veri yapıları sınıflar şeklinde gerçekleştirilmişlerdir. Bu sınıflardan önemli olan bazıları şu şekildedir.

- MifMap Sınıfı: Yazılımda bir coğrafi sete karşılık gelir. Tüm katmanlar için ortak olan özelliklere sahiptir. `AddLayer()` ve `RemoveLayer()` yöntemleri ile sete yeni katmanlar eklenebilir veya varolan katmanlar setten kaldırılabilir. `UpdateBounds()` ise harita sınırlarının güncellenmesini sağlar.
- MifLayer Sınıfı: Harita katmanlarına karşılık gelen sınıftır. Katmandaki tüm coğrafi objeleri içerir. `AddObject()` yöntemi ile dosyadan okunan

coğrafi objeler katmana eklenir ve katman sınırları güncellenir. Sınıfta bulunan pek çok özellik çizimde kullanılacak stil bilgelerini içerir.

- MapObject Sınıfı: Yazılımda coğrafi objelerin karşılığıdır. Bir coğrafi objeye ait pek çok özellik bulundurur. Objenin geometrik şekli, objeyi oluşturan noktalar, objeyi çevreleyen dikdörtgen gibi.
- PointD Sınıfı: X ve Y koordinatlarını belirten iki double değerden oluşan sınıftır. Noktaya karşılık gelir.
- RectangleD Sınıfı: Sol üst ve sağ alt köşesinin koordinatlarından oluşturulan bir dikdörtgene karşılık gelir. Koordinatlar double tipinde sayılardır. Coğrafi objeleri çevreleyen sınırlar, katman ve harita sınırları, haritanın görünen alanın sınırları hep bu sınıftan bir dikdörtgenle gerçekleştirilirler. `Expand()` yöntemi dikdörtgeni verilen bir noktayı içerecek şekilde genişletir. `Shift()` ve `Scale()` yöntemleri dikdörtgeni sırasıyla kaydırmaya ve ölçeklendirmeye yararlar. `Contains()` ve `Intersects()` yöntemleri ise dikdörtgenin verilen bir noktayı içerip içermediğini ve verilen bir dikdörtgenle kesişip kesişmediğinin anlaşılması için kullanılır. Ayrıca bu sınıf iki dikdörtgenin birleşiminden oluşan dikdörtgenin bulunmasına yarayan bir de operatöre sahiptir.
- Listener Sınıfı: Bu sınıf konum verilerinin alınmasında kullanılır. `Start()` yöntemi ile gelecek veriler beklenmeye başlanır. Sınıfta bulunan `Received()` olayı veri geldiğinde tetiklenir ve yazılımda bu olay yazılımda verilerin alınıp görüntüleme işleminin yenilenmesi için kullanılır.

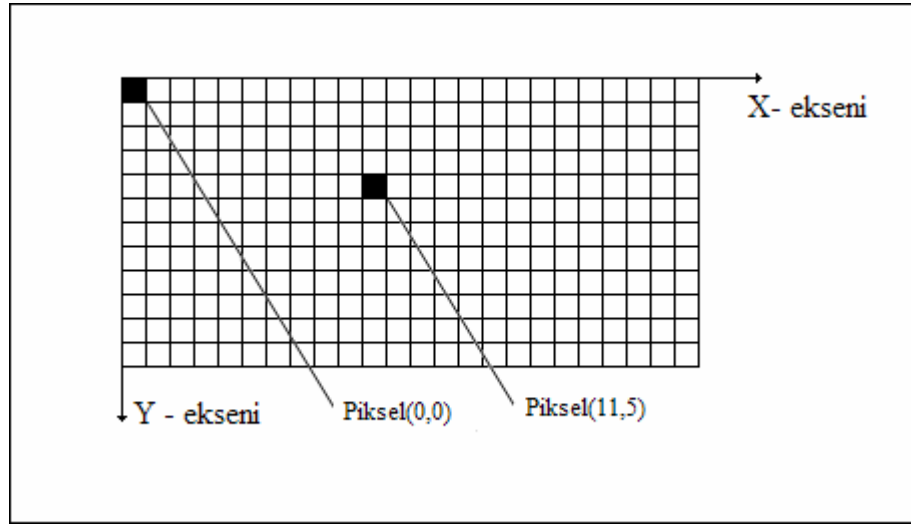
5.3 Yazılımda Kullanılan Teknikler

Yazılımın geliştirilmesi süresince değişik sorunlarla karşılaşmış, bu sorunlar gerek çözümler bulunarak, gerekse bazı yazılım teknikleri uygulanarak aşılmıştır. Alt bölümlerde bu çözümler ve teknikler ele alınmıştır.

5.3.1 Koordinat dönüşümü

Sayısal haritaların görüntülenmesinde dikkat edilmesi gereken noktalardan biri

ekran ve haritanın farklı koordinat sistemi ve birimlere sahip olmasıdır. Ekran koordinat sisteminde orijin sol üst köşede bulunur. X-ekseni sağa, Y-ekseni ise aşağı doğrudur (Şekil 5.4). Ekranın koordinat sisteminin birimi ise pikseldir. Bu durumda haritanın görüntülenen kısmı beklenin tersine baş aşağı olacaktır.



Şekil 5.4 Bilgisayar ekranının koordinat sistemi

Başka bir konu ise istenen zoom seviyesine göre koordinat sisteminin yeniden ölçeklendirilmesinin ve haritanın farklı bölümlerinin görüntülenebilmesi için de kaydırılmasının gerekmesidir.

Ekranında görüntülenmek istenen coğrafi objeleri tanımlayan tüm koordinatlar, görüntüleme işleminden önce bu yeni koordinat sistemine dönüştürülmelidirler. Bu işlem bir dizi matematiksel işlemin tüm koordinatlara uygulanmasını gerektirir. Böyle bir çözüm, yazılımın görüntüleme işleminin yavaşlamasına neden olur.

Bu yazılımın geliştirilmesinde, koordinat dönüşümlerinin daha hızlı bir şekilde yapılabilmesi için, Microsoft Visual Studio .NET tarafından sağlanan *Matrix* sınıfından faydalanıldı. Bir *Matrix*, bir koordinat dönüşümünü temsil eder (Microsoft Corporation 2002) ve koordinat dönüşümleri için gereken araçları içerir. *Matrix* sınıfının kullanımı ile hem koordinat dönüşümü problemi çözülmekte hem de bu işlemin süresi kısaltılarak görüntüleme işlemi biraz hızlandırılmaktadır. Ayrıca, bu sınıfın kullanımı işlemin daha az ve daha anlaşılır kod ile başarılmasını sağlamıştır. Şekil 5.5'de koordinat dönüşüm işlemini gerçekleştiren bir C# kod örneği görülmektedir.

```

MyMatrix = new Matrix(1, 0, 0, -1, 0, 0);
MyMatrix.Scale((float)zoom, (float)zoom);
MyMatrix.Translate((float)-dispX, (float)-dispY);
MyMatrix.TransformPoints(pts);

```

Şekil 5.5 Koordinat dönüştürme işlemlerini gerçekleştiren kod örneği.

Şekil 5.5’de verilen örnek kod parçasında önce Y-ekseninin yönünün yukarı doğru olmasını sağlayan bir *Matrix* oluşturulmuştur. Daha sonra bu *Matrix* istenilen zoom seviyesine göre ölçeklenmiş, ardından da haritanın görüntülenmek istenilen bölümüne göre gereken miktarda kaydırılmıştır. Son olarak bir coğrafi objeyi temsil eden tüm koordinatlar (*pts*), bu yeni koordinat sistemine dönüştürülmüşlerdir.

5.3.2 Multithreading

Bu çalışma ile geliştirilen yazılımda aynı anda birden fazla işlemin yapılabilmesi gerekmektedir. Örneğin, bir taraftan haritaların görüntülenmesi, görüntünün güncellenmesi, kullanıcı isteklerine yanıt verilmesi gibi işlemler yapılırken, diğer taraftan, gelecek konum bilgileri beklenmeli, gelen veriler kullanılacak bir formata dönüştürülmelidirler. Ancak, konum bilgilerinin beklenmesi işlemi, bu işlemi gerçekleştiren kodların sürekli çalışmasını gerektirdiğinden, yazılımın diğer bölümlerinin çalışmasını kesmekte, örneğin kullanıcı isteklerine yanıt verilemez bir duruma gelmekte ya da görüntü yenilenememektedir.

Multithreading bu tür sorunlara çözüm getiren yazılım tekniğidir. Bir bilgisayarda aynı anda, Microsoft Word, Microsoft Excel vs. gibi birden çok yazılımın çalıştırılabilmesi anlamına gelen multiprocessing tekniğinin aksine, bir yazılımın aynı anda birden çok işlemi yapıyor görünmesini sağlar (Wigley ve Wheelwright 2003). Multithreading ile bir yazılım aynı anda çalışan ve thread ismi verilen birden çok işlem ünitesine bölünebilir.

Yazılımda bu teknik uygulanarak yukarıda söz edilen sorunlar giderildi. Bunun için yazılım bir ana bölüm ve bir alıcı bölüm olmak üzere iki ayrı işlem ünitesine ayrıldı. Alıcı bölüm gelen konum bilgilerini beklerken, ana bölüm sayısal haritaları görüntülemekte, görüntüyü yenilemekte, kullanıcı isteklerine yanıt vermektedir. Şekil 5.6’da görülen örnek kod bu tekniğin nasıl uygulandığını göstermektedir. Örnekteki

kodda önce Thread sınıfından recThread() isimli yeni bir işlem ünitesi oluşturulmakta ve ardından çalıştırılmaya başlanmaktadır. Bu işlem ünitesinde Receive() ismindeki yöntem çalışmaktadır.

```
recThread = new Thread(new ThreadStart(Receive));
recThread.Start();
```

Şekil 5.6 Bir thread oluşturulup başlatılması

5.3.3 Çoklu arabellekleme (double buffering)

Görüntüleme işlemlerinde karşılaşılan problemlerden biri, ekran görüntüsünde oluşan titremelerdir. Bu sorunla özellikle haritanın görünen alanının değiştirilmesi için yapılan görüntü kaydırma işlemleri sırasında karşılaşıldı. Bu sorun çoklu arabellekleme adı verilen bir programlama tekniği kullanılarak giderildi.

Çoklu arabellekleme, görüntünün ekran dışında hazırlanıp daha sonra ekrana kopyalanmasından oluşan bir GUI programlama tekniğidir (Blanchette ve Summerfield 2006). Bu teknikte ekranda oluşması istenen görüntü öncelikle bellekte oluşturulan bir resim dosyasına çizilerek hazırlanır. Görüntünün çizilmesi işlemi bittikten sonra dosya ekranda görüntülenir. Şekil 5.7’de verilen C# kodları anlatılan bu işlemin C# programlama dili ile nasıl gerçekleştirildiğini göstermektedir.

```
private void DrawMap()
{
    bm = new Bitmap(
        pnlMap.ClientRectangle.Width,
        pnlMap.ClientRectangle.Height
    );
    Graphics g = Graphics.FromImage(bm);
    Pen mapRgnPen = new Pen(Color.Black);
    g.DrawPolygon(mapRgnPen, pts);
    g.Dispose();
}

private void pnlMap_Paint(object sender,
    System.Windows.Forms.PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawImage(bm, 0, 0);
}
```

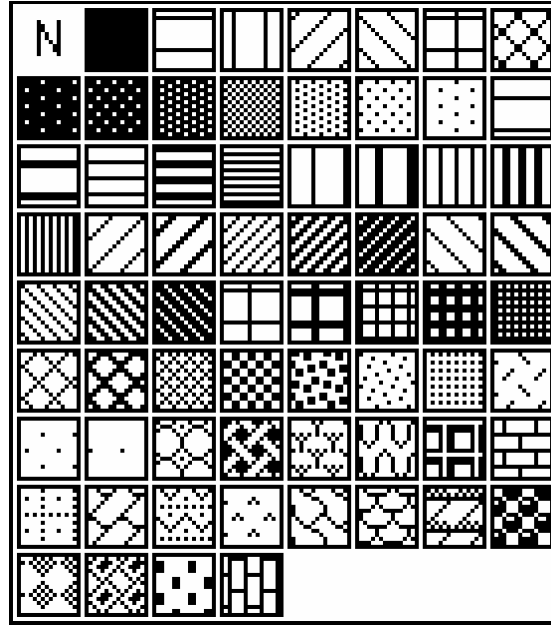
Şekil 5.7 Çoklu arabellekleme kod örneği

Şekil 5.7’de görülen `DrawMap()` adlı yöntemde bellekte görüntülemenin yapılacağı alanın boyutunda `Bitmap` sınıfından bir resim dosyası, daha sonra bu dosyaya çizim yapabilmek için bir `Graphics` ve bir de `Pen` sınıfından iki obje oluşturulur. Oluşturulan `Graphics` objesine ait `DrawPolygon()` yöntemi ile dosyaya siyah renkte bir poligon çizilir. Bu şekilde ekran görüntüsü bellekte hazırlandıktan sonra, ekran görüntüsü yenileneceği zaman tetiklenen `Panel` sınıfına `Paint` olayında, `Graphics` sınıfına ait `DrawImage()` yöntemi ile ekrana kopyalanır.

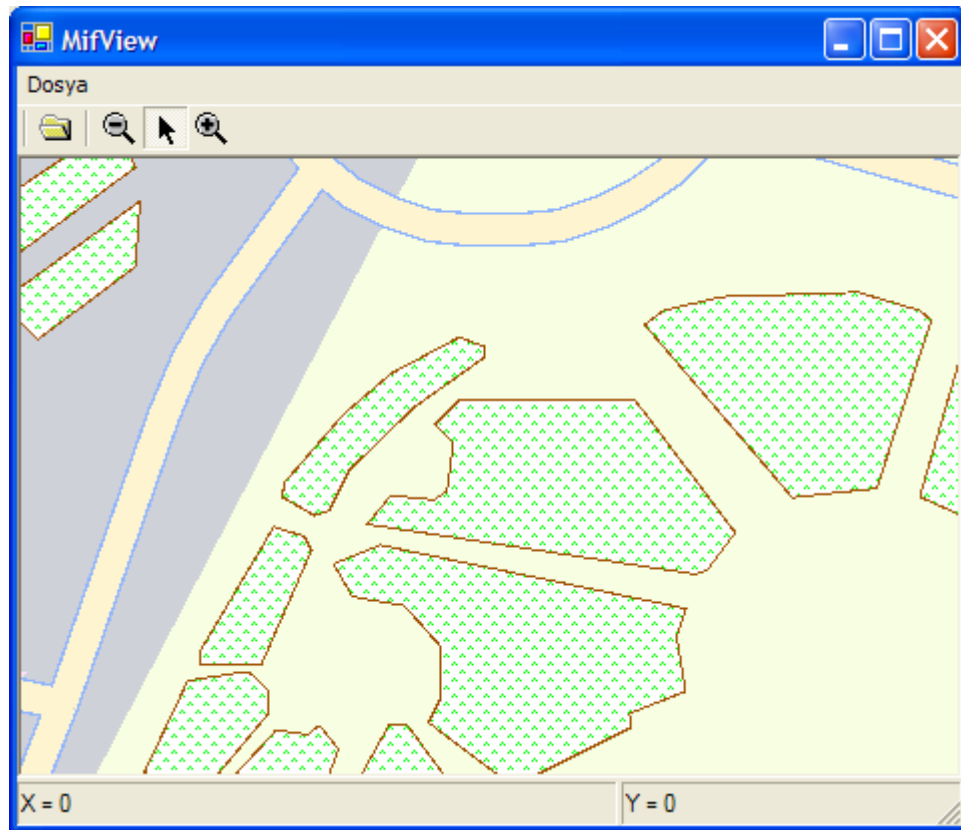
5.3.4 Dolgu desenleri

`MapInfo` sayısal haritaları, üzerlerindeki etiketler, çizgiler, alanların doldurulması, semboller ve metinlerin görüntülenmesi ile ilgili pek çok stil içerirler. Bunların pek çoğu kolaylıkla gerçekleştirilmiştir. Örneğin bazı semboller, özel bir font dosyasından bir karakterin görüntülenmesinden ibarettir. Bazıları ise diskte ilgili klasörden bir `Bitmap` dosyasının görüntülenmesi ile elde edilir. `Visual C#` metin biçimlendirme ve görüntüleme ile ilgili gerekli araçlara sahiptir. Bu konuda da bir zorlukla karşılaşmadı. Ancak alanlardan oluşan şekillerin bir desenle doldurulması için bir yöntem bulunması gerekiyordu. Aslında, bir alanın bir resim dosyası ya da bilinen bazı desenlerle doldurulabilmesi için de gerekli araçlar vardır. Örneğin bir alan yatay, dikey, çapraz, dik veya çapraz kesişen çizgilerle istenilen renklerde doldurulabilir. Sorun bunların dışında kalan desenlerdedir. Bu desenlerin elde edilmesi, istenilen renklere göre ön ve arka alanlarının renklendirilmesi, ya da saydam (transparent) desenlerin oluşturulması gerekmektedir.

Şekil 5.8’de görülen dolgu desenleri, `Windows` işletim sistemi ile birlikte gelen `Paint` isimli resim düzenleme programı ile 16×16 piksel boyutlarında ve arka alan beyaz ve ön alan siyah renk olacak şekilde tek tek pikseller ilgili renge boyanarak elde edildi. Daha sonra bu resimler `ImageList` sınıfından bir objeye eklendi. Bir `ImageList` resimlerden oluşan bir dizi olarak düşünülebilir. Katmana ait dosyadan, istenilen desenin elde edilen numarası ile bellekte başka bir yere kopyalanan desen, daha sonra yine dosyada belirtilen ön alan ve/veya arka alan renklerine göre yeniden düzenlenir. Bu işlem için resmin her pikselinin rengi siyah ve beyaz renklerle karşılaştırılır. Beyaz ise arka alan, siyah ise ön alan rengi ile değiştirilir. Çizilmekte olan coğrafi objenin sınırladığı alan, yeni oluşturulan bu resim ile doldurulur. Şekil 5.9, bu yöntemle doldurulan bir alanı göstermektedir.



Şekil 5.8 MapInfo dolgu desenleri



Şekil 5.9 Desenlerle doldurulmuş alanlar.

Şekil 5.10'da ise bir desenin ön alan renginin nasıl değiştirileceğini gösteren örnek kodlar görülmektedir. Burada ön alan siyah pikseller yeşil renk ile değiştirilmektedirler. Kodda işaretçi kullanımına izin veren `unsafe` bloğu bulunmaktadır. `LockBits()`

yöntemi işlemin daha hızlı gerçekleşmesini sağlamıştır. Bu kodla elde edilen desen daha sonra TextureBrush sınıfından bir obje aracılığı ile bir alanın doldurulmasında kullanılacaktır.

```

unsafe
{
    BitmapData bd = brushImg.LockBits(
        new Rectangle(0,0,16,16),
        ImageLockMode.ReadWrite,
        PixelFormat.Format32bppArgb
    );
    uint * p = (uint*)bd.Scan0;
    for(int x=0;x<256;x++)
        if(p[x] == 0xff000000)
            p[x]=0xff00ff00;
    brushImg.UnlockBits(bd);
}

```

Şekil 5.10 Desenin ön alan renginin değiştirilmesi.

5.3.5 Görüntüleme işleminin hızlandırılması

Bu çalışma ile geliştirilen yazılım gibi grafik görüntüleme işlemlerinin yoğunlukta olduğu yazılımlarda görüntüleme işlemlerinin hızı çok önemlidir. Görüntüleme hızı bu çalışma ile geliştirilen yazılımda en önemli sorun olarak karşımıza çıkmıştır. Herhangi bir katmanın on binlerce noktadan oluştuğu düşünülürse, görüntüleme işlemi bazen dakikalarca sürebilmekte, bu da yazılımın pratikte kullanımını olanaksız hale getirmektedir.

Bu sorunun çözümünde uygulanabilecek bazı teknikler vardır. Örneğin görüntüleme işleminden önce yapılması gereken hesaplamaları (rendering) hızlandırmak. Görüntüleme öncesi yapılan hesaplamalarda mümkün olduğunca tamsayı tipinde değişken kullanmak uygulanabilecek çözümlerden biridir. Çünkü tamsayılarla yapılan işlemler işlemciler tarafından çok daha hızlı bir şekilde yerine getirilir. Fakat hesapların duyarlılığı göz önüne alındığında bu çözümden yeterince faydalanılamamıştır. Bunun yerine, daha önce de değinildiği gibi, görüntüleme öncesi yapılan koordinat dönüştürme işlemlerinde Matrix sınıfının kullanımı yapılan işlemlere hız kazandırmıştır.

Daha önce de değinildiği gibi çizim işlemleri önce bellekte oluşturulan bir resim dosyası üzerinde gerçekleştirilmekte, bu işlemler bittikten sonra dosya ekranda görüntülenmektedir. Bu dosyanın boyutu da hızı etkileyen unsurlardan biridir. Dosya ne

kadar büyük olursa, hem dosya bellekte daha fazla yer kaplayacak, hem bu dosyaya yapılan çizim işlemleri daha uzun sürecek, hem de dosyanın görüntülenmesi daha çok zaman alacaktır. Bu yüzden dosya boyutu, tüm haritayı içerecek şekilde değil de sadece ekranda görüntülenecek alanla sınırlandı. Bu şekilde zoom seviyesi küçüldükçe büyüyen dosya boyutu ve oluşan yavaşlamanın önüne geçildi.

Ekran görüntüsünün sadece değişen bölümünün çizilmesi de görüntüleme işlemini hızlandırır. GDI+ bunun için gereken araçlara sahiptir. Örneğin, ekranda görüntülenen bir resmi bir miktar sürükleyip bıraktığımızı düşünelim. Resmin ekranda görünen kısmı zaten çizilmiş durumdadır ve sadece sürüklemenin gerçekleştiği kısım yeniden çizilmelidir. GDI+ kullanılarak ekranın sadece istenilen bölümünün yeniden çizilmesi gerçekleştirilebilir.

Görüntüleme işlemlerinden önce yapılan hesaplamaların en aza indirgenmesi uygulanabilecek bir başka tekniktir. Yazılımın görüntüleme hızı sorununun çözümünde en çok bu teknik yararlı olmuştur. Görüntüleme öncesi yapılan işlemlerin çok büyük bir kısmı koordinat dönüşüm işlemidir. Bu işlemlerin miktarı görüntülenecek katman sayısına bağlıdır. Ne kadar çok katman varsa o kadar çok işlem gerekir. Bu sayının azaltılması görüntüleme hızını artıracaktır. Çoğu zaman haritayı oluşturan katmanların tümünün birden görüntülenmesine gerek olmaz. Örneğin, bir Türkiye haritası görüntülemek istediğimizi varsayalım. Böyle bir durumda herhangi bir il merkezinde bulunan cadde ve sokaklarını görüntüleme istersek de Türkiye sınırlarını barındıran bir katmanın görüntülenmesine gerek yoktur. Bu iki görüntü birbirlerinden zoom seviyesi bakımından farklıdırlar. Türkiye haritasının görüntülediği zoom seviyesinde bir il merkezi ekranda küçük bir nokta olarak görünecek ve o il merkezine ait ayrıntılar istense de açık bir şekilde görüntülenemeyecek üstelik görüntünün bozulmasına neden olacaktır. Bu yüzden katmanların belli zoom seviyelerinde görüntülenmesi gerekir. Sayısal harita ve ona ait coğrafya seti dosyasında her katman için bu seviyeler belirtilmiştir. Yazılımda katmanlar için oluşturduğumuz `MifLayer` sınıfına ait `ZoomMin` ve `ZoomMax` isimli özellikler kullanılarak bir katmanın görüntülenip görüntülenmeyeceği kontrol edilir. O anki zoom seviyesi bu özelliklerde belirtilen sınırlar arasında ise katman görüntülenir, aksi halde görüntülenmez. Böylece herhangi bir anda görüntülenecek katman sayısı sınırlanarak görüntüleme işlemine hız kazandırılmış olur.

Görüntüleme öncesinde yapılan işlemleri azaltmanın bir başka yolu ise bir katmanda bulunan tüm coğrafi objeleri görüntülemeye çalışmamaktır. Çoğu zaman ekranda haritanın sadece bir bölümü görüntülenir. Bu yüzden bu objelerin çok büyük bir kısmı haritanın görünen alanı dışında kalır. Görünen alan dışında kalan bu objeleri oluşturan koordinatların da görüntüleme öncesi yapılan işlemlerde yer almasına gerek kalmaz ve böylece bu işlemler oldukça azaltılmış olur. Bir coğrafi objenin görünen alan içinde olup olmadığının anlaşılması için o objenin sınırlarının ve görünen alanın koordinatlarının bilinmesi gerekir. Bazı sayısal harita dosyaları her objenin koordinatları ile birlikte onu sınırlayan dikdörtgenin koordinatlarını da içerirler. Ne yazık ki mif dosyaları bu bilgiyi içermez. Bu yüzden yazılımda bu bilgi objelere ait koordinatlar ilgili dosyadan okunurken hesaplanmaktadır. Bu hesaplama okuma işlemlerini biraz yavaşlatsa da elde edilen bilginin görüntüleme hızına olan katkısı çok büyüktür. Yazılımda coğrafi objelerin karşılığı olarak oluşturduğumuz `MapObject` sınıfındaki `Bounds` özelliği o objenin sınırlarını belirleyen `RectangleD` sınıfı bir dikdörtgendir. Bir başka `RectangleD` sınıfı dikdörtgen ise yazılımda görünen alanın koordinatları için kullanılmaktadır. Bu iki dikdörtgen kesişiyorsa objenin görünen alan içinde olduğu kabul edilerek obje görüntülenir. Anlatılan bu yöntem görüntüleme hızını çok büyük ölçüde artırmıştır.

5.3.6 Soket programlama

Konum bilgileri internet üzerinde geleceğinden yazılımın internet erişimine sahip olması yani TCP/IP protokolünü kullanabilmesi gerekir. Bu işlem soket programlama olarak bilinir.

Soketler, ilk olarak Berkeley Üniversitesinde geliştirilen ve bir yazılımda TCP/IP protokolünün kullanılmasına olanak veren API'ler (Stevens 1993), yani uygulama geliştirme arayüzleridir. Bir soket, veri iletimi TCP ya da UDP protokolünü kullanır. Küçük boyutlu veriler için genellikle UDP kullanılır. Bu çalışmada geliştirilen yazılımda da bu protokol kullanılmıştır. Ayrıca soket belli bir portta çalışacağından bir port numarası verilmelidir. Yazılımda kullanılan port numarası 6065 olarak belirlenmiştir. Bu numaranın seçiminin özel bir nedeni yoktur. 1024 ile 65535 arasında herhangi bir numara seçilebilirdi.

Soket programlama, `Socket` sınıfı kullanılarak gerçekleştirilir. Önce bu sınıfa ait

bir obje oluşturulur. Sınıfın `Bind()` yöntemi ile soket belirlenen portta çalıştırılır. Daha sonra `ReceiveFrom()` yöntemi ile gelecek veriler beklenir.

5.4 Mevcut Ticari Yazılımlar ile Geliştirilen Yazılımın Karşılaştırılması

Ticari yazılımlara ait kaynak kodlar elde edilemeyeceğinden, ne yazık ki, bu yazılımların birtakım sorunları ne şekilde çözüldüğüne dair bilgi sahibi olunamamaktadır. Bu yüzden, kodların ve çözüm algoritmalarının karşılaştırılması mümkün olmamaktadır. Ancak, bu yazılımlar da bir şekilde soket programlama, grafik programlama, multithreading gibi yazılım tekniklerini kullanmaktadırlar. Bu teknikler doğrudan yazılımları geliştiren kişilerce uygulanmasa dahi, yazılımların geliştirilmesinde kullanılan yazılım bileşenleri uygulamaktadırlar.

Yazılımların sahip olduğu özellikler bakımından karşılaştıracak olursak, ticari yazılımlarda bulunan en temel özellikler, sayısal haritaların görüntülenmesi, zoom yapılabilmesi, görüntünün kaydırılması, birden fazla aracın izlenebilmesi gibi, bu çalışmada geliştirilen yazılımda da bulunmaktadır.

Ticari sistemlerin sahip oldukları diğer özellikler ise geliştirdiğimiz yazılımda bulunmamaktadır. Ancak bu doğrudan yazılımlar arasındaki bir fark olarak düşünülmemelidir. Çünkü bu özellikler biraz da araçlara monte edilen aygıtların özelliklerine bağlıdır. Aygıtların özelliklerine göre yazılımlara da yeni özellikler kazandırılabilir.

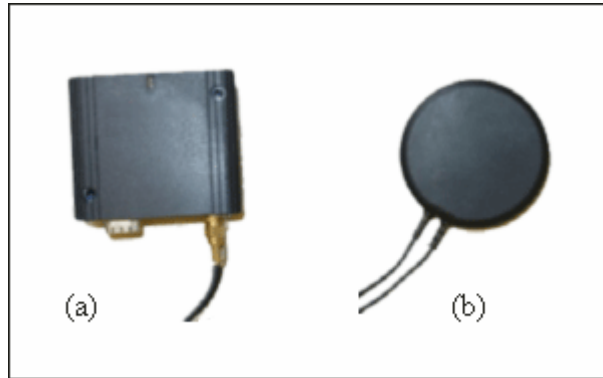
Bu çalışmada geliştirilen yazılımın, ticari yazılımlardan en önemli farkı, sayısal haritaların görüntülenmesi işleminde herhangi bir hazır yazılım bileşeni kullanılmaması ve sadece araç takip sistemleri düşünülerek geliştirilmiş olmasıdır. Bu şekilde sistemde kullanılmayacak pek çok özelliğin bulundurulmasına gerek kalmaz. Ayrıca gereken durumlarda koda müdahale edilerek çözüm algoritmalarının değiştirilip iyileştirilmesi mümkün olabilir. Çıkabilecek sorunların çözümü de bu şekilde çok daha kolay ve hızlı olacaktır. Çünkü işlemleri gerçekleştiren kodlar elinizdedir. Hazır bileşenlerle yapılan yazılım geliştirme işlemlerinde ya kullanılan bileşenlerin yeni sürümleri, ya da yeni bileşenler satın alınmak zorunda kalınır. Oysa hazır bir bileşen kullanarak harita görüntülüyor iseniz, bu bileşeni satın aldığınız yerden yardım almak zorunda kalabilirsiniz. Bu da çoğu kez çözümün çok daha yavaş olmasına sebep olur. Bir diğer

konu ise bu bileşenler yurt dışından satın alınmaktadır. Bu tezde yapılan tür çalışmalarla ülkemizin yazılım ithalatı azaltılabilir.

Geliştirdiğimiz yazılım sadece gst ve mif formatlarındaki haritaları görüntülemektedir. Bu bir eksiklik olabilir. Ancak, bir çok ticari yazılım da, geliştirildikleri araçlara bağlı olarak benzer kısıtlamalara sahiptir.

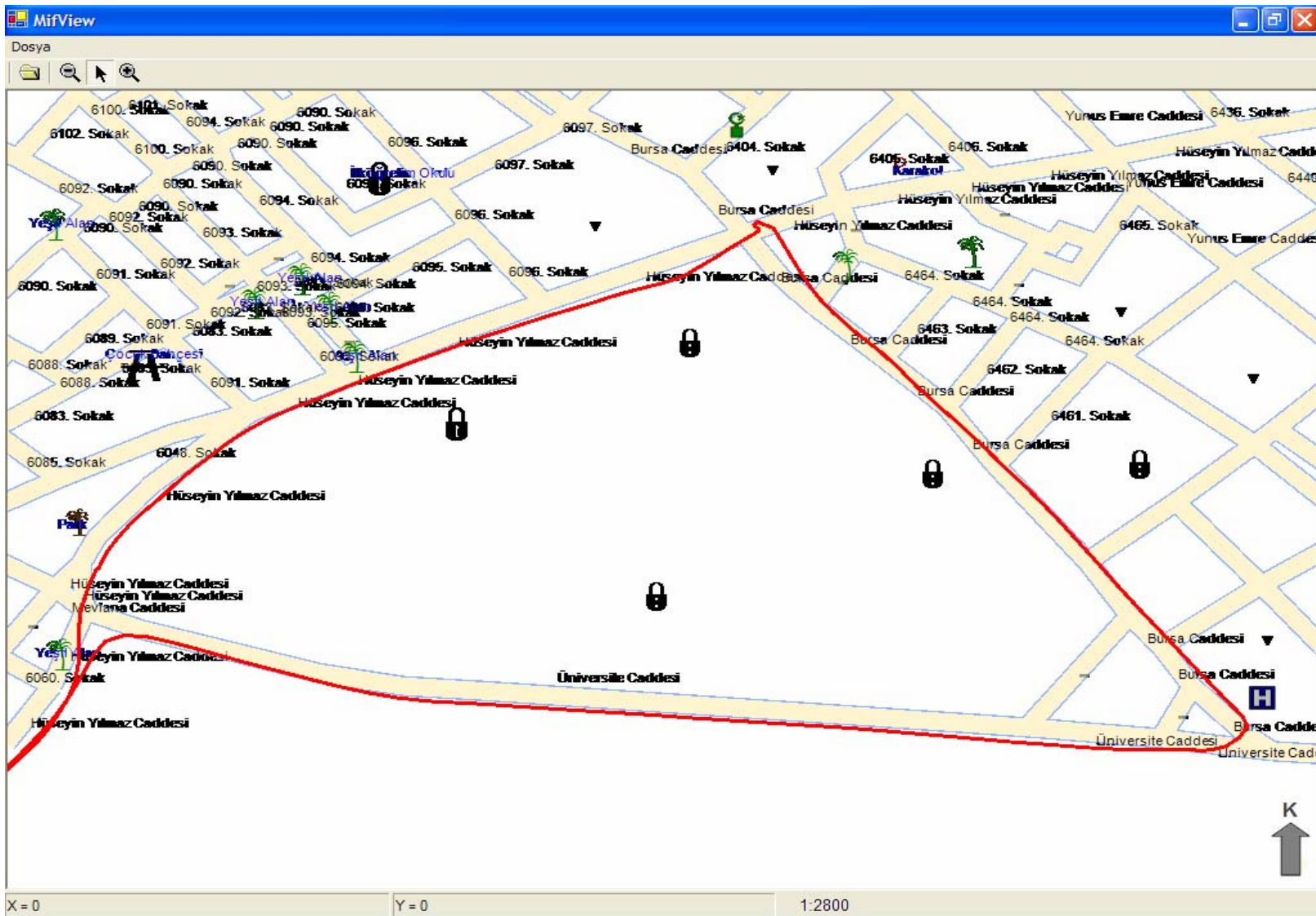
5.5 Yazılımın Test Edilmesi ve Örnek ATS

Yazılım bir araç takip sistemi oluşturularak test edildi. Sistemde konum bilgileri için bir GPS alıcısı, veri iletimi için bir GSM modülü ve izleme için de çalışmada geliştirilen yazılım kullanıldı. GSM modülü ve GPS alıcısı tek bir araç ünitesi (Şekil 5.11a) olarak gerekli diğer aparatlarla birlikte Tagem adlı firmadan elde edildi. Bu aparatlar, GPS uydularından gelen sinyallerin alınması ve GSM iletişimi için GPS ve GSM antenleri (Şekil 5.11b), bilgisayarın seri portundan bağlanarak aygıt ayarlarının yapılabilmesi için bir seri kablo ve araca monte etmek için kullanılan kablolardan oluşmaktadır.



Şekil 5.11 Araç ünitesi (a) ve antenler (b).

Öncelikle araç ünitesi bilgisayara bağlanarak, verilerin gönderileceği bilgisayarın IP numarası, gönderilecek port numarası, kullanılacak protokol, iletişimde kullanılacak teknoloji, veri gönderme sıklığı gibi ayarlar bir terminal programı yardımı ile yapıldı. Protokol olarak UDP, iletişim teknolojisi olarak GPRS seçildi. Araç ünitesi, bir SIM kartı ve antenler de takılarak kullanıma hazır hale getirildi. Daha sonra 12V DC ile çalışan ünite, izlenmek istenen aracın aküsüne bağlanıp antenler araç üzerine yapıştırıldı. Denizli caddelerinde gezdirilen araç, üniversite kampüsünde internet erişimine sahip bir bilgisayar üzerinde çalışan geliştirdiğimiz yazılımla gerçek zamanlı olarak bir süre izlendi (Şekil 5.12).



Şekil 5.12 Yazılımdan araç izleme sırasında alınmış bir ekran görüntüsü.

Şekil 5.12, Üniversite Caddesi, Bursa Caddesi ve Hüseyin Yılmaz Caddesi boyunca yol alıp tekrar başladığı noktaya dönen bir aracın ekran görüntüsüne aittir. İzleme, çok küçük sapmalar dışında oldukça başarılı bir şekilde gerçekleştirilmiştir. GPS alıcıları, konum bilgilerini belirli bir hata payı içinde üretirler. KKS tarafından sivil kullanıcılar için verilen standart servis, GPS alıcılarının hesapladığı konum bilgilerinin hassasiyetini belirler. Konum bilgileri genellikle 5 m.'lik bir hata payı içerirler.

Ayrıca, çalışmada kullanılan harita formatı, cadde, sokak vb. gibi coğrafi objelerin görüntüleme biçimine ait bilgiler içerir. Bu bilgiler arasında kullanılacak çizgi kalınlıkları gibi bilgiler de bulunmaktadır. Ekranda, aynı genişlikte görülen caddeler gerçekte böyle olmayabilir. Aracı, yolun dışına taşmış gibi gösteren sapmaların bir nedeni de budur.

6. SONUÇ VE ÖNERİLER

Bu çalışma ile araç takip sistemlerinde kullanılabilir bir yazılım geliştirildi. Yazılımın geliştirilmesi çalışmalarında karşılaşılan bazı zorluk ve sorunlar, uygulanan yazılım teknikleri ve bulunan çözümlerle aşılabilir pratikte uygulanabilir bir yazılım elde edildi. Özellikle, grafik uygulamalarında çok önemli olan hız, bu çalışmada karşılaşılan en büyük sorundu. Onbinlerce koordinattan oluşan harita katmanlarının çizim ve görüntü yenileme süreleri, uygulanan teknik ve çözüm yöntemleri ile dakikalar mertebesinde neredeyse anlık denebilecek kadar kısa sürelerle çekildi. Elde edilen yazılım örnek bir araç takip sistemi ile deneyerek yazılımın pratikte uygulanabilir bir yazılım olduğu görüldü. Araç ünitesinin monte edildiği araç, geliştirilen yazılım ile gerçek zamanlı olarak bilgisayar ekranı üzerinde oldukça başarılı bir şekilde izlenebilmiştir.

Bu yazılımın geliştirilmesi aşamasında, ticari bir araç takip sisteminin sahip olduğu tüm özelliklerin gerçekleştirilmesine çalışılmamıştır. Bunun yerine, yazılımın sistemin en temel işlevlerine sahip olması ile yetinilerek, çabalar sayısal haritaların görüntülenmesi işlemine odaklanmıştır. Söz konusu işlemler hazır CBS geliştirme araçları ve CBS yazılım bileşenleri kullanılarak çok kolay ve süratli bir şekilde gerçekleştirilebilmektedir. İncelenebilen tüm yazılımlar da bu tür araç ve bileşenler kullanılarak geliştirilmişlerdir. Bu çalışmada geliştirilen yazılımda bu araç ve bileşenler kullanılmamıştır. Bu yüzden çalışmanın başlangıçta belirlenen hedeflere ulaştığını söylemek mümkündür.

Yazılım sadece MapInfo mif formatındaki sayısal haritaları görüntülemektedir. Ancak bu çok önemli bir eksiklik sayılmaz. Benzer kısıtlamalar mevcut ticari yazılımlarda da bulunmaktadır. Ayrıca, dosya yapısı açıklanmış sayısal haritaların yazılıma eklenmesi çok zor bir iş olmayacaktır. Yapılması gereken sadece bu dosyaların okunup coğrafi objelerin bu dosyalardan belleğe alınmasından ibarettir.

Bu yazılımın aslında iki önemli eksiği bulunmaktadır. Birincisi harita projeksiyonlarının tümü gerçekleştirilmemiştir. İkincisi ise en kısa güzergahın tespiti gibi sisteme nitelik kazandıracak bazı özelliklerdir. Söz edilebilecek çok önemli

olmayan bir eksiklik ise karmaşık çizgi stillerinin gerçekleştirilmemiş olmasıdır. Çizgi stili göz ardı edilebilecek bir konu olmasına rağmen bu eksikliği giderilmesi de sayısal haritaların görselliği bakımından yerinde olur.

Kullanılacak ekipmanların sisteme kazandıracığı niteliklerle birlikte düşünüldüğünde geliştirilen yazılım kolaylıkla ticari bir araç takip yazılımına dönüştürülebilir.

Yazılım, araç takip sistemleri temel alınarak geliştirilmiş olmasına karşın kullanım alanının bu sistemlerle kısıtlandığı düşünülmemelidir. Yazılım sayısal harita işlevselliği gerektiren başka herhangi bir alanda küçük değişikliklerle kullanılabilir, yazılım geliştirme araç ve bileşenleri üretilebilir.

EKLER

EK-1 MapInfo Geoset (gst) Dosyaları

Coğrafya setleri uzantısı gst olan ve seti oluşturan katmanlarla ilgili bilgileri içeren dosyalardır (MapInfo Corporation 2002). Dosya, aslında her satırında bir anahtar sözcük ve o anahtara ait bir değer içeren metin dosyasıdır. Anahtar sözcükler ve değerler çift tırnak içinde yazılırlar ve hiyerarşik bir yapıdadırlar. Aşağıda verilen şekil örnek bir geoset dosyasını göstermektedir.

```
!GEOSET
!VERSION 450
begin_metadata
"\GEOSET\NAME" = "denizli_buyuk1"
"\GEOSET\PROJECTION" = ""
"\GEOSET\CENTER" = "29.0801,37.7766"
"\GEOSET\CENTER_USER" = "29.0801,37.7766"
"\GEOSET\ZOOMLEVEL" = "0.469943"
"\GEOSET\AUTOLAYER" = "FALSE"
"\TABLE\1\FILE" = "onemli_kapi.no.TAB"
"\TABLE\1\ISVISIBLE" = "TRUE"
"\TABLE\1\ZOOM\MIN" = "0"
"\TABLE\1\ZOOM\MAX" = "0.776714"
"\TABLE\1\AUTOLABEL" = "TRUE"
"\TABLE\1\SHOWLINEDIRECTION" = "FALSE"
"\TABLE\1\SHOWNODES" = "FALSE"
"\TABLE\1\SHOWCENTROIDS" = "FALSE"
"\TABLE\1\SELECTABLE" = "TRUE"
"\TABLE\1\EDITABLE" = "FALSE"
"\TABLE\1\LABEL\ZOOM\MIN" = "0"
"\TABLE\1\LABEL\ZOOM\MAX" = "0.932057"
"\TABLE\1\LABEL\DUPLICATE" = "FALSE"
"\TABLE\1\LABEL\LINESTYLE" = "0"
"\TABLE\1\LABEL\OFFSET" = "5"
"\TABLE\1\LABEL\OVERLAP" = "TRUE"
"\TABLE\1\LABEL\PARALLEL" = "TRUE"
"\TABLE\1\LABEL\POSITION" = "7"
"\TABLE\1\LABEL\PARTIALSEGMENTS" = "FALSE"
"\TABLE\1\LABEL\MAXLABELS" = "10"
"\TABLE\1\LABEL\FONT\STYLE" = "1"
"\TABLE\1\LABEL\FONT\EXTSTYLE" = "0"
"\TABLE\1\LABEL\FONT\DESCRIPTION" = "Arial TUR"
"\TABLE\1\LABEL\FONT\SIZE" = "8"
"\TABLE\1\LABEL\FONT\FORECOLOR" = "255"
end_metadata
```

Her anahtar, "GEOSET" ya da "TABLE" sözcüklerinden biri ile başlar. "GEOSET" ile başlayan anahtarlar tüm haritanın özellikleri için, "TABLE" ile başlayanlar ise ilgili katmanın özellikleri içindir. Geoset dosyalarında kullanılan tüm anahtarlar ve anlamları aşağıda verilmiştir.

\GEOSET\

NAME	Haritanın ismini belirleyen bir metin değer.
PROJECTION	Haritada kullanılan projeksiyon.

CENTER	Sırasıyla harita merkezinin enlem ve boylamını belirleyen iki sayı.
\MBR\LOWERLEFT	Sırasıyla haritanın sol alt köşesinin enlem ve boylamını belirleyen iki sayı.
\MBR\UPPERRIGHT	Sırasıyla haritanın sağ üst köşesinin enlem ve boylamını belirleyen iki sayı.
ZOOMLEVEL	Haritanın zoom seviyesini belirleyen bir sayı.
AUTOLAYER	Katmanların otomatik olarak sıralanıp sıralanmayacağını belirleyen TRUE ya da FALSE değer.
MAPUNIT	Haritada kullanılacak birimi belirleyen bir sayı.
ROTATION	Haritanın döndürme açısını belirleyen bir sayı.
\TABLE\<numara>\	
FILE	Dosyanın yerini belirleyen metin değer.
DESCRIPTION	Katmanı tanımlayan metin değer.
ISVISIBLE	Katmanın görüntülenip görüntülenmeyeceğini belirleyen TRUE ya da FALSE değer.
AUTOLABEL	Katmandaki objelerin otomatik olarak etiketlenip etiketlenmeyeceğini belirleyen TRUE ya da FALSE değer.
DRAWLABELSAFTER	Etiketlerin katmandan sonra görüntülenip görüntülenmeyeceğini belirleyen TRUE ya da FALSE değer.
SHOWLINEDIRECTION	Çizgilerin yönlerini gösteren okların görüntülenip görüntülenmeyeceğini belirten TRUE ya da FALSE değer.
SHOWNODES	Geometrik şekilleri oluşturan noktaların görüntülenip görüntülenmeyeceğini belirten TRUE ya da FALSE değer.
SHOWCENTROIDS	Bir alana sahip geometrik şekillerin merkezlerinin görüntülenip görüntülenmeyeceğini belirten TRUE ya da FALSE değer.
EDITABLE	Katmanda seçilen bir geometrik şeklin yerinin değiştirilmesi, silinmesi veya boyutlarının değiştirilmesi gibi işlemlerin yapılıp yapılamayacağını belirten TRUE ya da FALSE değer.
SELECTABLE	Katmanın seçilip seçilemeyeceğini belirten TRUE ya da FALSE değer.
REGISTERINGEODICT	Bu çalışmada kullanılmıyor.
ZOOMMIN	Katmanın görüntüleneceği en küçük zoom seviyesini belirten sayı.
ZOOMMAX	Katmanın görüntüleneceği en büyük zoom seviyesini belirten sayı.
\DISPLAY\BRUSH\	
PATTERN	Bir alana sahip geometrik şekillerin doldurulmasında kullanılacak deseni belirten bir sayı.
FORECOLOR	Bir alana sahip geometrik şekillerin doldurulmasında kullanılacak renk.
BACKCOLOR	Bir alana sahip geometrik şekillerin doldurulmasında kullanılacak arka alan rengi.
TRANSPARENT	Bir alana sahip geometrik şekillerin doldurulmasında kullanılan desenin saydam olup olmayacağını belirten TRUE ya da FALSE değer.

DISPLAY\PEN

LINEWIDTH	Bir alana sahip geometrik şekillerin sınırlarının çiziminde kullanılacak çizgi kalınlığını belirten bir sayı.
LINESTYLE	Bir alana sahip geometrik şekillerin sınırlarının çiziminde kullanılacak çizgi stilini belirten bir sayı.
COLOR	Bir alana sahip geometrik şekillerin sınırlarının çiziminde kullanılacak çizgi rengini belirten bir sayı.

DISPLAY\LINEPEN

LINEWIDTH	Çizgilerden oluşan geometrik şekillerin çiziminde kullanılacak çizgi kalınlığını belirten bir sayı.
LINESTYLE	Çizgilerden oluşan geometrik şekillerin çiziminde kullanılacak çizgi stilini belirten bir sayı.
COLOR	Çizgilerden oluşan geometrik şekillerin çiziminde kullanılacak çizgi rengini belirten bir sayı.

DISPLAY\FONT

STYLE	Metinlerin gösteriminde kullanılacak font stilini belirten bir sayı.
EXTSTYLE	Metinlerin gösteriminde uygulanacak ekstra stilleri belirten bir sayı.
DESCRIPTION	Kullanılacak fontu belirten metin.
SIZE	Kullanılacak Font büyüklüğünü belirten bir sayı.
FORECOLOR	Metin rengini belirten bir sayı.
BACKCOLOR	Arka alan rengini belirten bir sayı.
OPAQUE	Arka alan renginin görüntülenip görüntülenmeyeceğini belirten TRUE ya da FALSE değer.

DISPLAY\SYMBOL

TYPE	Kullanılan sembol tipini belirten bir sayı.
CODE	Görüntülenecek sembol karakterini belirten bir sayı.
COLOR	Görüntülenecek sembol rengini belirten bir sayı. Bu sayı vektör, true type font ya da bitmap türlerinden birini belirtir.
POINTSIZE	Sembol büyüklüğünü belirten bir sayı.
FONT	True type font türü semboller için kullanılacak fontu belirtir. DISPLAY\FONT ile aynıdır.
FONT\ROTATION	Sembolün görüntülenmesine kullanılacak dönüş açısını belirten bir sayı.
BITMAP\NAME	Bitmap türü semboller için görüntülenecek resim dosyasının adını veren metin.
BITMAP\OVERRIDECOLOR	Sembol renginin resim dosyasına uygulanıp uygulanmayacağını belirten TRUE ya da FALSE değer.
BITMAP\TRANSPARENT	Sembolün saydam olup olmayacağını belirten TRUE ya da FALSE değer.

LABEL

ZOOM\MIN	Geometrik şekillere ait etiketlerin görüntülenmesinde kullanılacak en küçük zoom seviyesi.
ZOOM\MAX	Geometrik şekillere ait etiketlerin görüntülenmesinde

	kullanılacak en büyük zoom seviyesi.
FONT	Etiketlin görüntülenmesinde kullanılacak font. DISPLAY\FONT ile aynıdır.
DUPLICATE	Aynı etiketin birden fazla kez görüntülenip görüntülenmeyeceğini belirten TRUE ya da FALSE bir değer.
PARALLEL	Çizgilerden oluşan geometrik şekillerin etiketlenmesinde etiketin çizgiye paralel olup olmayacağını belirten TRUE ya da FALSE değer.
OVERLAP	Etiketlerin birbirleri üstünde görüntülenmesine izin verilip verilmeyeceğini belirten TRUE ya da FALSE değer.
PARTIALSEGMENTS	Sadece küçük bir bölümü görünen geometrik şekillerin etiketlenip etiketlenmeyeceğini belirten TRUE ya da FALSE değer.
MAXLABELS	Bir katman için otomatik olarak üretilecek en fazla etiket sayısını belirten bir sayı.
LINETYPE	Etiketi, ait olduğu geometrik şekle bağlayan çizginin türünü belirten bir sayı.
OFFSET	Etiket ile ait olduğu geometrik şekil arasındaki mesafeyi belirten bir sayı.
POSITION	Etiketlin, ait olduğu nesneye göre olan konumunu belirten bir sayı.

EK-2 MapInfo Mif Dosyaları

MapInfo mif dosyaları iki bölümden oluşan metin dosyalarıdır: bir başlık bölümü ve bir de data bölümü. Başlık bölümünde dosyanın versiyonu, katman için kullanılacak koordinat sistemi ve katmandaki coğrafik objelerle ilgili bilgileri içeren mid uzantılı dosyanın yapısı ile ilgili bulunur (MapInfo Corporation 2004).

Data bölümü ise coğrafik objeler, koordinat bilgileri ve bu objelerin çiziminde uygulanacak stillerle ilgili bilgileri içerir. Bu bölüm DATA sözcüğü ile başlar. İstenildiği kadar coğrafik obje belirtilebilir. Her coğrafik obje, mid dosyasındaki bir satıra karşılık gelir. Mif dosyasındaki ilk obje mid dosyasındaki ilk satır ile, ikinci obje ikinci satır ile v.b. eşleşir. Eğer mid dosyasındaki herhangi bir satıra karşılık gelen bir coğrafik obje yok ise bu durum objenin bulunması gereken yerde NONE sözcüğü ile belirtilir. Aşağıda örnek bir mif dosyası verilmektedir:

```

Versi on 300
Charset "Neutral "
Delimi ter ", "
CoordSys Earth Projecti on 1, 104 Bounds (-1000, -1000) (1000, 1000)
Col umns 1
  ADI Char(100)
Data

Poi nt 29. 101732 37. 75372
  Symbol ("PIN6-32. BMP", 0, 24, 0)
Poi nt 29. 096486 37. 762889
  Symbol ("MOSQ1-32. BMP", 0, 18, 0)
Regi on 1
  5
29. 092079 37. 76244
29. 091526 37. 762104
29. 09187 37. 761484
29. 092478 37. 761663
29. 092079 37. 76244
  Pen (1, 2, 16711680)
  Brush (5, 11593215, 16777215)
  Center 29. 092002 37. 761962
Li ne 29. 079729 37. 786114 29. 080263 37. 786637
  Pen (1, 191, 8421504)
Pl ine 3
29. 080263 37. 786637
29. 080767 37. 787088
29. 080858 37. 787224
  Pen (1, 191, 8421504)
Li ne 29. 080858 37. 787224 29. 081147 37. 787892
  Pen (1, 191, 8421504)
Pl ine 3
29. 081147 37. 787892
29. 081666 37. 787699
29. 081694 37. 787684
  Pen (1, 191, 8421504)

PLINE MULTIPLE 2
3

```

```

29.090617 37.759658
29.09096 37.75912
29.091169 37.758812
4
29.090748 37.75808
29.090518 37.758006
29.090036 37.758079
29.089845 37.758051
  Pen (1, 191, 8421504)
Elipse 29.095570105882 37.782561021022 29.096945 37.783652978977
  Pen (4, 2, 255)
  Brush (1, 0, 16777215)
Poi nt 29.093097 37.761352
  Symbol (64, 6303744, 20, "MapI nfo Mi scel l aneous", 0, 0)

```

Data bölümünde kullanılabilecek coğrafik objelere ait anahtar sözcükler ve anlamları şu şekildedir:

POINT : Nokta, Haritada görüntülenecek bir sembol.

LINE : Çizgi.

PLINE : Bir çok bölümden oluşan çizgi.

REGION : Bir ya da daha fazla poligondan oluşan kapalı şekil.

ARC : Ark.

TEXT : Metin.

RECT : Dikdörtgen.

ROUNDRECT : Köşeleri yuvarlanmış dikdörtgen.

ELLIPSE : Elips.

MULTIPOINT : Birden fazla nokta.

COLLECTION : Yukarıdaki şekillerin birkaçını içerir.

Bir POINT, iki parametre ile tanımlanır, x ve y koordinatları.

POINT x y

Bir LINE, çizginin iki uç noktasının koordinatlarından oluşan dört parametre ile tanımlanır.

LINE x1 y1 x2 y2

Bir PLINE, bir ya da daha fazla bölümden oluşabilir. Birden fazla bölümden oluştuğunda PLINE sözcüğünün hemen ardından MULTIPLE anahtar sözcüğü ve bölüm sayısı gelir. Daha sonra her bölüm için nokta sayısı ve x-y koordinatları belirtilir.

```
PLINE [MULTIPLE bölümsayısı]
    noktasayısı1
    x1 y1
    x2 y2
    .
    .
    [noktasayısı2
    x1 y1
    x2 y2
    .
    .
    ]
    .
    .
```

Bir REGION, bir ya da daha fazla poligondan oluşur. REGION sözcüğünden hemen sonra poligon sayısı belirtilir. Daha sonra, her poligon için nokta sayısı ve x-y koordinatları sıralanır.

```
REGION poligonsayısı
    noktasayısı1
    x1 y1
    x2 y2
    .
    .
    [noktasayısı2
    x1 y1
    x2 y2
    .
    .
    ]
    .
    .
```

Bir ARC, onu sınırlayan dikdörtgenin karşılıklı köşegenlerinin koordinatları ile çıkış ve giriş açıları kullanılarak belirtilir. Açılar için pozitif yön saat yönünün tersi ve birimi derecedir.

```
ARC x1 y1 x2 y2
    çıkışaçısı girişaçısı
```

Bir TEXT, 255 karakter uzunluğuna kadar olan bir metinden oluşur. Birden fazla satırdan oluşan metinlerde satır araları için satır sonu karakteri kullanılmalıdır. Ayrıca metni çevreleyen dikdörtgenin karşılıklı köşelerinin koordinatları da belirtilmelidir.

```
TEXT "metin"
      x1 y1 x2 y2
```

Bir RECT, karşılıklı köşelerinin koordinatları ile belirtilir.

```
RECT x1 y1 x2 y2
```

Bir ROUNDRECT, köşeleri yuvarlanmış bir dikdörtgendir. Karşılıklı köşelerinin koordinatları ile birlikte köşeler için bir de yuvarlama açısı belirtilir.

```
RECTANGLE x1 y1 x2 y2
           yuvarlamaaçısı
```

Bir ELLIPSE, kendini sınırlayan dikdörtgenin karşılıklı koordinatları ile belirtilir.

```
ELLIPSE x1 y1 x2 y2
```

Bir MULTIPOINT, nokta sayısı ve ardından birlikte x-y koordinatları ile belirtilir.

```
MULTIPOINT noktasayısı
           x1 y1
           x2 y2
           .
           .
```

Bir COLLECTION değişik objelerden oluşur. İçerdiği obje sayısının ardından her obje, yukarıdaki formatlara uygun biçimde belirtilir.

```
COLLECTION objesayısı
           POINT
           REGION
           PLINE
           .
           .
           .
```

EK-3 MapInfo çizgi stilleri

Çizgi stilleri, bir alana sahip objelerin sınırları ya da çizgiler ile gösterilen objeler için renk, kalınlık çizgi biçimi gibi nitelikleri için kullanılır. Bu stiller dosyanın data bölümünde hemen objenin ardından gelir ve şu şekilde belirtilir:

PEN(kalınlık, biçim, renk)

Kalınlık, birimi piksel ya da nokta sayısı olan bir tamsayıdır. Kalınlık 1-7 için birim piksel, 11-2047 için birim nokta sayısıdır.

Renk 24-bit RGB rengi gösteren bir tamsayıdır.

Biçim ise bir çizgi biçimine karşılık gelen bir tamsayıdır. Şekilde MapInfo çizgi biçimlerini ve numaralarını görülmektedir.

01		31	→→→→→→→→	61	→→→→→	91	■ ■ ■ ■ ■ ■ ■ ■
02	—————	32	- - - - -	62	—	92	■ ■ ■ ■ ■ ■ ■ ■
03	33	→→→→→	63	———	93	■ ■ ■ ■ ■ ■ ■ ■
04	———	34	- - - - -	64	———	94	▲
05	35	→→→→→	65	———	95	———▲
06	- - - - -	36	→→→→→	66	———	96	▲
07	- - - - -	37	→→→→→	67	———	97	▲▲▲▲▲▲▲▲
08	- - - - -	38	→→→→→	68	———	98	▲▲▲▲▲▲▲▲
09	39	→→→→→	69	———	99	▲▲▲▲▲▲▲▲
10	40	→→→→→	70	———	100	▲▲▲▲▲▲▲▲
11	- - - - -	41	→→→→→	71	- - - - -	101	▲▲▲▲▲▲▲▲
12	- - - - -	42	→→→→→	72	———	102	←
13	- - - - -	43	→→→→→	73	———	103	→
14	- - - - -	44	→→→→→	74	———	104	←
15	- - - - -	45	→→→→→	75	———	105	●●●●●●●●
16	- - - - -	46	→→→→→	76	———	106	●●●●●●●●
17	- - - - -	47	→→→→→	77	———	107	●●●●●●●●
18	- - - - -	48	→→→→→	78	●———	108	■ ■ ■ ■ ■ ■ ■ ■
19	- - - - -	49	→→→→→	79	———●	109	■ ■ ■ ■ ■ ■ ■ ■
20	- - - - -	50	→→→→→	80	●———●	110	~~~~~
21	- - - - -	51	→→→→→	81	●●●●●●●●	111	~~~~~
22	- - - - -	52	→→→→→	82	●●●●●●●●	112	~~~~~
23	- - - - -	53	→→→→→	83	●●●●●●●●	113	~~~~~
24	- - - - -	54	→→→→→	84	■ ■ ■ ■ ■ ■ ■ ■	114	■ ■ ■ ■ ■ ■ ■ ■
25	- - - - -	55	→→→→→	85	■ ■ ■ ■ ■ ■ ■ ■	115	■ ■ ■ ■ ■ ■ ■ ■
26	———	56	→→→→→	86	■———	116	■ ■ ■ ■ ■ ■ ■ ■
27	———	57	→→→→→	87	———■	117	■ ■ ■ ■ ■ ■ ■ ■
28	———	58	→→→→→	88	■———■	118	
29	———	59	→→→→→	89	■ ■ ■ ■ ■ ■ ■ ■		
30	———	60	→→→→→	90	■ ■ ■ ■ ■ ■ ■ ■		

EK-4 MapInfo dolgu stilleri

Dolgu stilleri bir alana sahip objelerin alanlarının dolgu deseni, rengi gibi nitelikleri için kullanılır. Bu stiller dosyanın data bölümünde objenin ardından gelir ve şu şekilde belirtilir:

BRUSH(desen, ön alan rengi, arka alan rengi)

Arka alan rengi isteğe bağlıdır ve belirtilmeyebilir. Her iki renk de 24-bit RGB rengi belirten bir tamsayıdır.

Desen ise 1-71 arasında bir tamsayıdır. 9-11 arasındaki desen numaraları rezerve edilmiştir ve kullanılmamaktadır. Dolgu desenlerinden birine karşılık gelir (Bkz. Şekil 5.8).

EK-5 MapInfo sembol stilleri

Sembol stilleri bir noktanın görünüşünü belirler. Sembol stilleri üç sınıfa ayrılır:

- MapInfo 3.0
- TrueType Font
- Bitmap

Ek 5.1 MapInfo 3.0 sembol stilleri

Bu sınıf MapInfo'nun eski versiyonlarında kullanılır. Sembol şu şekilde belirtilir:

SYMBOL(şekil, renk, büyüklük)

Şekil, MapInfo 3.0 sembollerinden birine karşılık gelen 31-67 arasında bir tamsayıdır.

Renk 24-bit RGB rengi gösteren bir tamsayıdır.

Büyüklik ise 1-48 arasında noktanın büyüklüğünü belirten bir tamsayıdır. Bu semboller ve numaraları aşağıda verilmiştir.

31		41	☆	51	✳	61	🛡
32	■	42	△	52	✈	62	🛡
33	◆	43	▽	53	🏠	63	🏠
34	●	44	◼	54	🚩	64	✕
35	★	45	▲	55	🏠	65	🚩
36	▲	46	●	56	+	66	🚩
37	▽	47	➔	57	▲	67	🚩
38	□	48	➔	58	⚓		
39	◇	49	+	59	⊕		
40	○	50	✕	60	🏠		

Ek-5.2 TrueType Font sembol stilleri

Belirli bir fontta bir karaktere dayalı sembol için kullanılır. Sembol şu şekilde belirtilir:

SYMBOL(şekil, renk, büyüklük, font adı, font stili, aç)ı

Şekil, renk ve büyüklük önceki gibidir. Fakat burada şekil, belirtilen fontta bir karaktere karşılık gelir.

Font adı sembolün bulunduğu fontu belirten bir metindir.

Font stili sembolün görünüşünü kontrol eden bir tamsayıdır. Aşağıda kullanılabilir stiller tablo halinde verilmiştir.

Font Stil Değeri	Etkisi
0	Düz metin
1	Kalın
16	Sembol çevresinde siyah çerçeve
32	Gölge
256	Sembol çevresinde beyaz çerçeve

Birden fazla etki oluşturmak için tablonun sol sütunundan ilgili değerlerin toplamının verilmesi gerekir.

Açı ise sembolün ne kadar çevrileceğini belirten derece cinsinden bir reel sayıdır.

Ek-5.3 Bitmap sembol stilleri

Bir bitmap dosyasından sembol görüntülemek için kullanılır. Sembol şu şekilde belirtilir:

SYMBOL(doya adı, renk, büyüklük, stil)

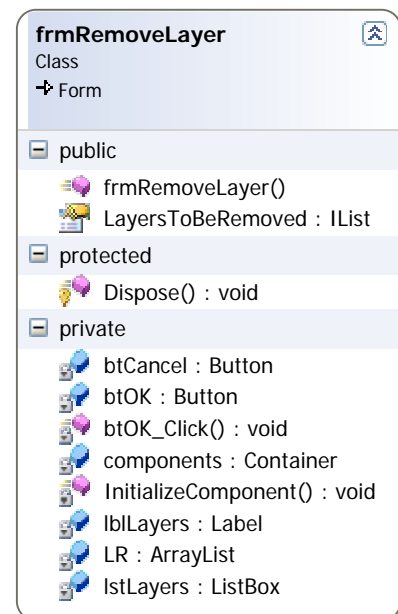
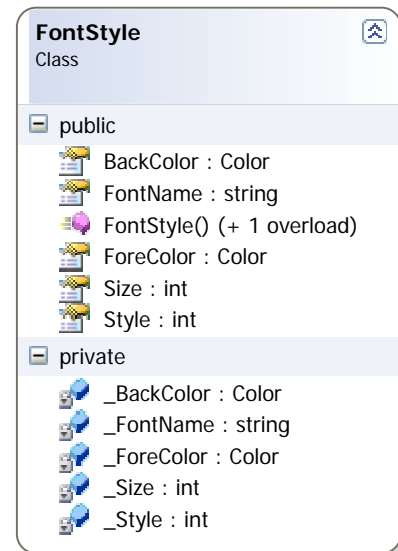
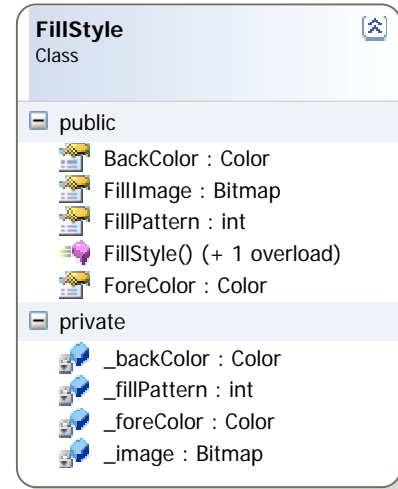
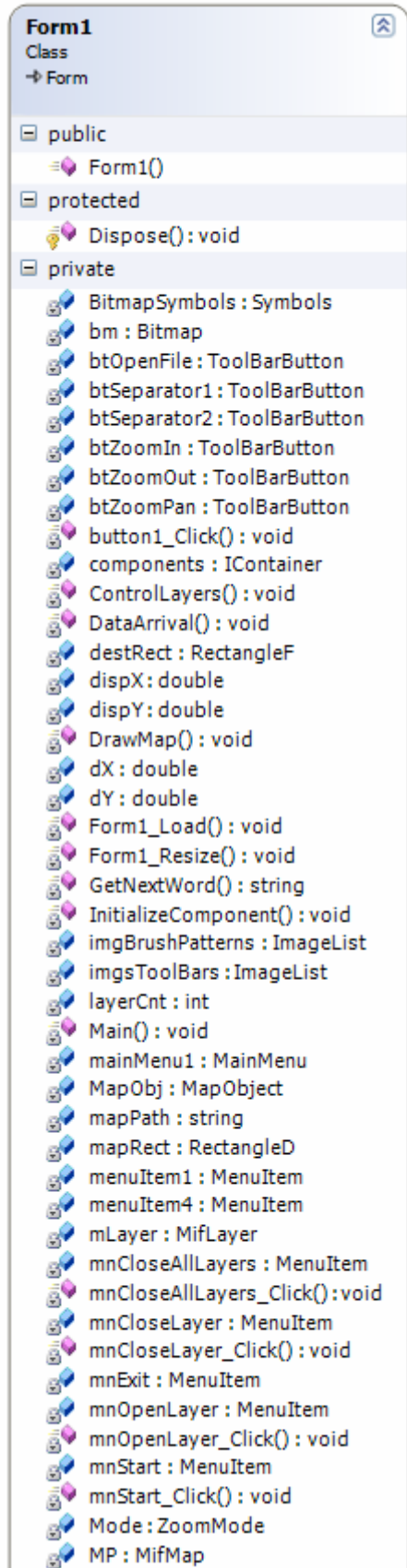
Renk ve büyüklük öncekiler gibidir.

Dosya adı sembolün bulunduğu dosyanın adını belirten metindir.

Stil ise sembolün art alan ve renk özelliklerini belirleyen bir tamsayıdır. Tabloda stil değerlerinin sembolün görünüşüne olan etkisi verilmektedir.

Stil Değeri	Etkisi
0	Sembol dosyada olduğu gibi fakat beyaz pikseller saydam olacak şekilde görüntülenir.
1	Sembol dosyada olduğu gibi görüntülenir.
2	Dosyadaki beyaz olmayan pikseller belirtilen renk ile değiştirilerek, beyaz pikseller ise saydam olacak şekilde görüntülenir.
3	Sadece, beyaz olmayan pikseller belirtilen renk ile değiştirilerek görüntülenir.

EK-6 Yazılımda kullanılan sınıfların sınıf diyagramları



```

OpenMifLayer() : void
panel1 : Panel
pnlMap : mPanel
pnlMap_MouseDown() : void
pnlMap_MouseMove() : void
pnlMap_MouseUp() : void
pnlMap_Paint() : void
pnlToolBars : Panel
pnlX : StatusBarPanel
pnlY : StatusBarPanel
pX : int
pY : int
radius : double
ReadData() : string
ReadGeoSet() : void
ReadGeosetProperties() : void
ReadMifLayer() : void
ReadObjectStyles() : void
ReadTable() : void
RefreshMap() : void
RefreshMapView() : void
SetMapZoom() : void
SetTransform() : void
ShiftMap() : void
ShiftTransform() : void
srcRect : RectangleF
statusBar1 : StatusBar
tlbFile : ToolBar
tlbFile_ButtonClick() : void
tlbZoom : ToolBar
tlbZoom_ButtonClick() : void
udpListener : Listener
vehicleTrack : ArrayList

```

Nested Types

LabelProperties

Class

```

public
Duplicate : bool
LabelFont : FontStyle
LabelMax : int
LabelProperties()
LabelZoomMax : double
LabelZoomMin : double
LineType : int
Offset : int
Overlap : bool
Parallel : bool
PartialSegments : bool
Position : int
Visible : bool
ZoomLabel : bool

private
_duplicate : bool
_labelMax : int
_labelZoomMax : double
_labelZoomMin : double
_lblFont : FontStyle
_lineType : int
_offset : int
_overlap : bool
_parallel : bool
_partialSegments : bool
_position : int
_visible : bool
_zoomLabel : bool

```

LineStyle

Class

```

public
LineColor : Color
LinePattern : int
LineStyle() (+ 1 overload)
LineWidth : float

private
_lineColor : Color
_linePattern : int
_lineWidth : float

```

Listener

Class

```

public
allDone : ManualResetEvent
data : string
DataReceived : ReceivedHandler
Listener()
StartServer() : void

private
Receive() : void
recThread : Thread
sb : StringBuilder

```

MapObject
Class

- public
 - Bounds : RectangleD
 - Caption : string
 - Center : PointD
 - Line : LineStyle
 - MapFont : FontStyle
 - MapObject() (+ 1 overload)
 - MapSymbol : SymbolStyle
 - Name : string
 - Points : IList
 - RegionFillStyle : FillStyle
 - Type : int
- private
 - _Bounds : RectangleD
 - _Caption : string
 - _Center : PointD
 - _Line : LineStyle
 - _mapFont : FontStyle
 - _mapSymbol : SymbolStyle
 - _Name : string
 - _regionFillStyle : FillStyle
 - _Type : int
 - PL : ArrayList

MifLayer
Class

- public
 - AddObject() : void
 - AutoLabel : bool
 - Bounds : RectangleD
 - File : string
 - LabelProps : LabelProperties
 - Line : LineStyle
 - MapObjects : IList
 - maxX : double
 - maxY : double
 - MifLayer()
 - minX : double
 - minY : double
 - Name : string
 - OverrideStyle : bool
 - RegionBorder : LineStyle
 - RegionFillStyle : FillStyle
 - Texts : IList
 - Visible : bool
 - ZoomLayer : bool
 - ZoomMax : double
 - ZoomMin : double
- private
 - _AutoLabel : bool
 - _Bounds : RectangleD
 - _file : string
 - _lbp : LabelProperties
 - _Line : LineStyle
 - _name : string
 - _OverrideStyle : bool
 - _regionBorder : LineStyle
 - _regionFillStyle : FillStyle
 - _Visible : bool
 - _ZoomLayer : bool
 - _ZoomMax : double
 - _ZoomMin : double
 - LO : ArrayList
 - LT : ArrayList
- Nested Types

mPanel
Class
↳ UserControl

- public
 - mPanel()
- protected
 - Dispose() : void
- private
 - components : Container
 - InitializeComponent() : void

ReceivedEventArgs
Class
↳ EventArgs

- public
 - Data : string
 - ReceivedEventArgs()
 - RemoteHost : IPEndPoint

PointD
Class

- public
 - PointD()
 - X : double
 - Y : double
- private
 - _X : double
 - _Y : double

RectangleD
Class

public

- Contains() : bool
- Expand() : void
- Intersects() : bool
- operator +() : RectangleD
- P1 : PointD
- P2 : PointD
- RectangleD() (+ 1 overload)
- Scale() : void
- Shift() : void

private

- _X1 : double
- _X2 : double
- _Y1 : double
- _Y2 : double

SymbolStyle
Class

public

- CustomStyle : int
- FileName : string
- FontName : string
- FontStyle : int
- Rotation : float
- Shape : int
- Size : int
- SymbolColor : Color
- SymbolStyle() (+ 1 overloa...)
- SymbolType : int

private

- _fontStyle : int
- _Name : string
- _Rotation : float
- _Size : int
- _style : int
- _symbolColor : Color
- _symbolType : int

Symbols
Class
↳ DictionaryBase

public

- Add() : void
- Bitmaps : ICollection
- Contains() : bool
- Keys : ICollection
- Remove() : void
- Symbols()
- this : Bitmap

KAYNAKLAR

- Blanchette, J., Summerfield, M. (2006) C++ GUI Programming with Qt 4, **Prentice Hall**, ,560s
- Chrisman, N. (1984) On Storage of Coordinates in Geographic Information Systems, *Geoprocessing*, 2:259-270
- Chrisman, N. (2002) Exploring Geographic Information Systems, **John Wiley and Sons**, 305s
- Dana, P. H. (1999) Coordinate Systems Overview, <http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys.html>
- Dana, P. H. (2000) Map Projection Overview, <http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj.html>
- Dueker, K., J., Kjerne, D. (1989) Multipurpose cadastre: Terms and definitions, **ASPRS and ACSM**, Falls Church, s.7-8
- Escobar, F., Hunter, G., Bishop, I., Zerger, A. (2001) Introduction to GIS, <http://www.sli.unimelb.edu.au/gisweb/>
- EUROCONTROL and IfEN (1998) WGS 84 Implementation Manuel
- Lee, W. (2003) Windows XP Unwired, **O'Reilly**, , 278s
- MapInfo Corporation (2002) MapX Developer's Guide, **MapInfo Corporation**
- MapInfo Corporation (2004) MapInfo Professional v7.8 User Guide, **MapInfo Corporation**, New York, 680s
- McHarg, I., L. (1969) Design With Nature, **Natural History Press**, New York, 197s
- Microsoft Corporation (2002) Microsoft .NET Framework SDK Documentation, **Microsoft Corporation**
- Mitchell, T. (2005) Web Mapping Illustrated, **O'Reilly**, , 368s
- O'Sullivan D., Igoe T. (2004) Physical Computing: Sensing and Controlling the Physical World with Computers, Course Technology, , 494s
- Rodriguez, A., Gatrell, J., Karas, J., Pescheke, R. (2001) TCP/IP Tutorial and Technical Overview, IBM Corporation, , 957s

Stevens, W., R. (1993) TCP/IP Illustrated, Volume 1: The Protocols, **Addison Wesley Professional**, 600s

U.S. Coast Guard Navigation Center (1996) Navstar GPS User Equipment Introduction, <http://www.gps.gov/pub/gps/gpsuser/gpsuser.pdf>

Wigley, A., Wheelwright, S. (2003) Microsoft® .NET Compact Framework (Core Reference), **Microsoft Press**, 896s

WEB_1 (1997) Coordinate Systems – A Deeper Understanding, http://www.posc.org/Epicentre.2_2/DataModel/ExamplesofUsage/eu_cs32.html (10.10.2006)

ÖZGEÇMİŞ

Önder ÇİVRİL 7 Eylül 1968 tarihinde Denizli’de doğmuştur.

Eylül 1980 – Haziran 1983 yılları arasında İzmir Fevzi Çakmak Ortaokulu’nda orta öğrenimini tamamlamış, 1983 yılında TÜBİTAK ortaokullar arası matematik yarışmasında Ege Bölgesi Üçüncülük ödülü kazanmıştır. Eylül 1983 – Haziran 1986 yılları arasında İzmir Fen Lisesi’nde lise öğrenimine devam etmiştir. Ekim 1987’de Ortadoğu Teknik Üniversitesi Mühendislik Fakültesi Elektrik – Elektronik Mühendisliği Bölümü’nde lisans eğitimine başlamış ve Şubat 1996’da bu bölümden mezun olmuştur.

Mayıs 1996’da Ege Gübre Sanayi A.Ş. bilgisayar programcısı olarak iş hayatına başlamış ve Kasım 2002’den bu yana Pamukkale Üniversitesi Denizli Meslek Yüksekokulu Bilgisayar Teknolojisi ve Programlama Programı’nda öğretim görevlisi olarak çalışmaktadır.