

COĞRAFI VERİ YAPILARI

Hakan SARBANOĞLU

ÖZET

Coğrafi Bilgi Sistemleri, coğrafi varlıklara ait grafik ve grafik olmayan bilgilerin toplanması, depolanması, işlenmesi, analizi ve gösterimi fonksiyonlarını bütünleşik olarak yerine getiren donanım ve yazılım bileşenlerinden oluşur. Coğrafi Veri Tabanı ise Coğrafi Bilgi Sistemlerinin yazılım bileşenlerinin çekirdeğini oluşturur. Coğrafi Veri Tabanlarında depolanan grafik ve grafik olmayan bilgilerin Coğrafi Bilgi Sisteminden beklenen fonksiyonları etkin bir şekilde yerine getirecek şekilde yapılandırılmaları gerekir. Yazıda coğrafi bilgi ile ilgili temel kavramlar sunulduktan sonra çeşitli coğrafi veri yapılarının özellikleri açıklanmaktadır.

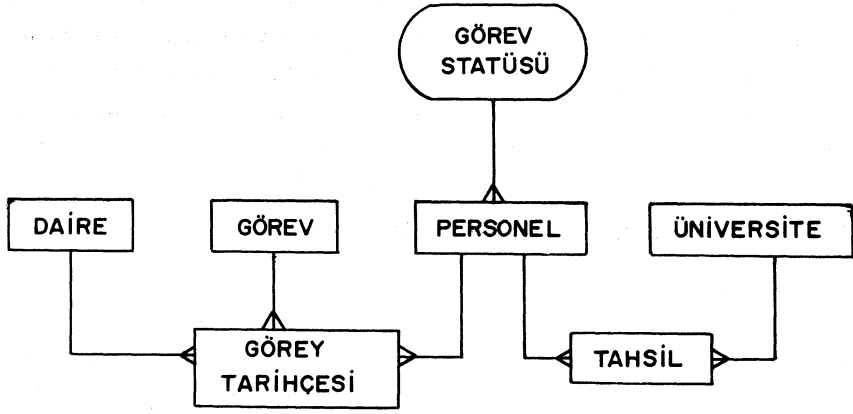
1. GİRİŞ : (COĞRAFI VERİ TABANI FARKLILIĞI)

Normal bir veri tabanında, belirli varlıklara (entities) ait bilgiler tutulur. Belli tekniklerle yapılan sistem analiz ve tasarım çalışması sonucunda; veri tabanında hangi varlıklara ait hangi bilgilerin bulundurulacağı tek tek belirlenir. Öznitelik adı verilen bu bilgilerin, "ait olduğu varlığa özgü" olmalarına dikkat edilir. Tasarım sırasında ayrıca özniteliklerin her biri için veri özellikleri de (veri tipi, uzunluğu, geçerlilik koşulları vb.) saptanır.

Örnek :

Varlık	Personel			
Öznitelikler:	<u>Öznitelik adı</u>	<u>Veri tipi</u>	<u>Uzunluğu</u>	<u>Geçerlilik Koşulu</u>
	personel-no	karakter	6	
	adı	karakter	10	
	soyadı	karakter	14	
	cinsiyet-kodu	karakter	1	E veya K
	adres	karakter	50	
	şehir	karakter	20	
	posta kodu	karakter	5	
	doğum tarihi	karakter	11	

Nasıl ki gerçek yaşamda varlıklar arasında çeşitli ilişkiler varsa, veri tabanında da varlıklar arasında gerekli ilişkiler kurulabilir. Hangi varlıklar arasında hangi ilişkilerin kurulacağı, sorgulama gereksinimlerine göre, yine sistem analiz ve tasarım çalışması sırasında saptanır. Varlıklar arasındaki ilişkiler sorgulamalardaki erişim yollarını belirler. Varlıklar ve varlıklar arasındaki ilişkiler "varlık-ilişki diagramı (E-R diagramı)" veya "mantıksal veri yapısı (LDS)" adı verilen şemalarla gösterilirler. Şekil-1 de örnek olarak bir personel veri tabanının basit mantıksal yapısı LDS şeması ile gösterilmektedir.



————— : Bire bir ilişki

—————< : Bire çok ilişki

>—————< : Çoka çok ilişki

Şekil-1 : Personel veri tabanı mantıksal yapısı (örnek)

Bu şemada herbir kutu "hakkında bilgi tutulan bir varlığı" temsil etmektedir. Bu varlıklar somut, soyut veya türetilmiş olabilir. Kutular arasındaki çizgiler varlıklar arasındaki ilişkileri göstermektedir. Örneğin "bir personel zaman içinde birçok dairede birçok görevlere atanabilir." İki varlık ara-

sındaki bire bir ilişki birinci varlığa ait bir kaydın, ikinci varlığa ait sadece bir kayıtle ve ikinci varlığa ait bir kaydın birinci varlığa ait yine sadece bir kayıtle ilişkili olduğunu gösterir. Bire çok ilişki ise birinci varlığa ait bir kaydın ikinci varlığa ait birçok kayıtle; ancak ikinci varlığa ait bir kaydın birinci varlığa ait sadece bir kayıtle ilişkili olduğu anlamına gelir.

Veri tabanlarının mantıksal yapıları belli biçimlerde modellendirilmektedir. Veri modeli, fiziksel depolama ve erişim yolları başta olmak üzere veri tabanının birçok özelliklerini biçimlendirmektedir ve buna göre her veri tabanı yönetim sistemi yazılımı belli bir modeli esas almaktadır.

En çok kullanılan veri tabanı modelleri

- Hiyerarşik model,
- Ağ modeli (CODASYL),
- Ters çevrilmiş listeler modeli ve
- İlişkisel model'dir.

Çok kısaca özetlenen bu yapısı ile veri tabanı teknolojisi günümüzde artık tatminkâr bir düzeye erişmiştir ve bilgi sistemlerinin temel taşlarını oluşturmaktadır.

Coğrafi veri tabanının klasik bir veri tabanından en önemli farkı varlıkların diğer bilgilerine ek olarak konuma bağlı ("spatial" veya "positional") bilgilerinde veri tabanında tutulmasıdır. Bu nedenle bir coğrafi veri tabanındaki bilgiler konuma bağlı olan ve olmayan şeklinde ("spatial" ve "non-spatial") veya grafik ve grafik olmayan ("graphical" ve non-graphical") şeklinde gruplandırılmaktadır.

Bir varlığın konuma bağlı bilgileri ;

- Varlığın belli bir referans sistemine göre konumunu
- ve
- Varlığın biçimini ifade ederler.

Böylesi varlıklara da "coğrafi varlıklar" denir. Son yıllarda yaygınlaşan nesneye yönelik sistem (object oriented sytem) yaklaşımının da etkisiyle coğrafi veri tabanlarında varlık yerine nesne (object) terimi daha çok kullanılır olmuştur.

2. DEĞİŞEN KAVRAMLAR, YENİ TANIMLAR

Bilgisayar destekli sayısallaştırma ve çizim sistemleri (CADD: Computer Aided Digitizing and Drawing) ve ardından Coğrafi Bilgi Sistemleri (GIS : Geographical Information Systems) teknolojileri birçok yeni kavramı da beraberlerinde getirmiş ve haritacılık terminolojisinde yeni tanımlar ortaya çıkmıştır. Aşağıda bu kavramlardan coğrafi veri yapılarına altlık teşkil eden bir bölümü özetlenmiştir.

a. Varlıklar veya Nesneler

Haritacılık terminolojisindeki "detay" kavramına karşılık veri tabanı terminolojisinde "varlık" veya "nesne" kelimeleri kullanılmaktadır.

b. Üç Boyutlu Veri Tabanı

Bütün detaylar gerçekte üç boyutlu nesnelerdir. Üç boyutlu veri tabanı kavramı değişik biçimlerde yorumlanabilir. Bir yoruma göre üç boyutlu veri tabanı "detayların biçiminin" üç boyutlu olarak temsil edildiği bir veri tabanıdır. Diğer bir yoruma göre üç boyutlu veri tabanı "detayların konumunun" üç boyutlu (x, y, z veya ϕ, λ, h) saklandığı veri tabanlarıdır. Bugünkü donanım ve özellikle yazılım sınırlamaları nedeniyle büyük bir alandaki detayların biçiminin üç boyutlu olarak saklanması pratik olarak mümkün değildir. Detayların konumlarının üç boyutlu tutulması ise mümkündür. Bu işlem ya her detay noktası için üç koordinat saklamak suretiyle veya detayın sadece yatay koordinatlarının saklanması ve istendiğinde yükseklik bilgisinin bölgeye ait sayısal arazi modelinden enterpole edilmesi suretiyle gerçekleştirilmektedir.

c. Ölçek

Detayların biçimi bugün için üç boyutlu sayısallaştırılmadığından, bu detayların iki boyutlu bir referans yüzeyindeki izdüşümlerinin çerçevesi sayısallaştırılır. Bu sayısallaştırmada bir incelik söz konusudur. Örneğin bir su deposu her ne kadar üç boyutlu bir cisimse ve her ne kadar yatay izdüşümü daire veya dörtgen bir şekilse de genellikle tek bir nokta olarak sayısallaştırılır.

Yeryüzü üzerinde sayısallaştırılacak detayların sonsuz sayıda olduğu söylenebilir. Dolayısı ile bir "özetleme" söz konusu olur. "Genelleştirme" programları için tasarlanan algoritmalar henüz yetersiz durumdadır. Bu nedenle "Ölçekten Bağımsız Veri Tabanları" bugün için mümkün değildir. Başka bir deyişle detaylar daha önce herhangi bir amaca yönelik olarak özetlenmekte veya seçim yapılmakta ve daha sonra sayısallaştırılmaktadır. Gerek "detayların

biçimlerinin temsil edilmesinde", gerekse "detay sıklığında" belli bir ayrıntı düzeyi" söz konusudur. Buna göre grafik bir kavram olmamakla beraber coğrafi veri tabanlarının "ölçeklerinden" söz edilebilir. Ancak bu ölçek kavramı klasik tanımından oldukça farklıdır ve veri tabanındaki detayların sıklığının, özetleme derecesinin ve şekillerinin temsil biçimlerinin bir ifadesidir.

d. Raster ve Vektör Teknikleri

Şekilleri tanımada insan gözü oldukça yeteneklidir. Oysa bir bilgisayarda şekilleri temsil etmek üzere çok daha başka yaklaşımlar gereklidir. Bilgisayarda şekilleri temsil etmek üzere birbirinden tamamen farklı ancak birbirini tamamlayan iki değişik yol izlenmektedir.

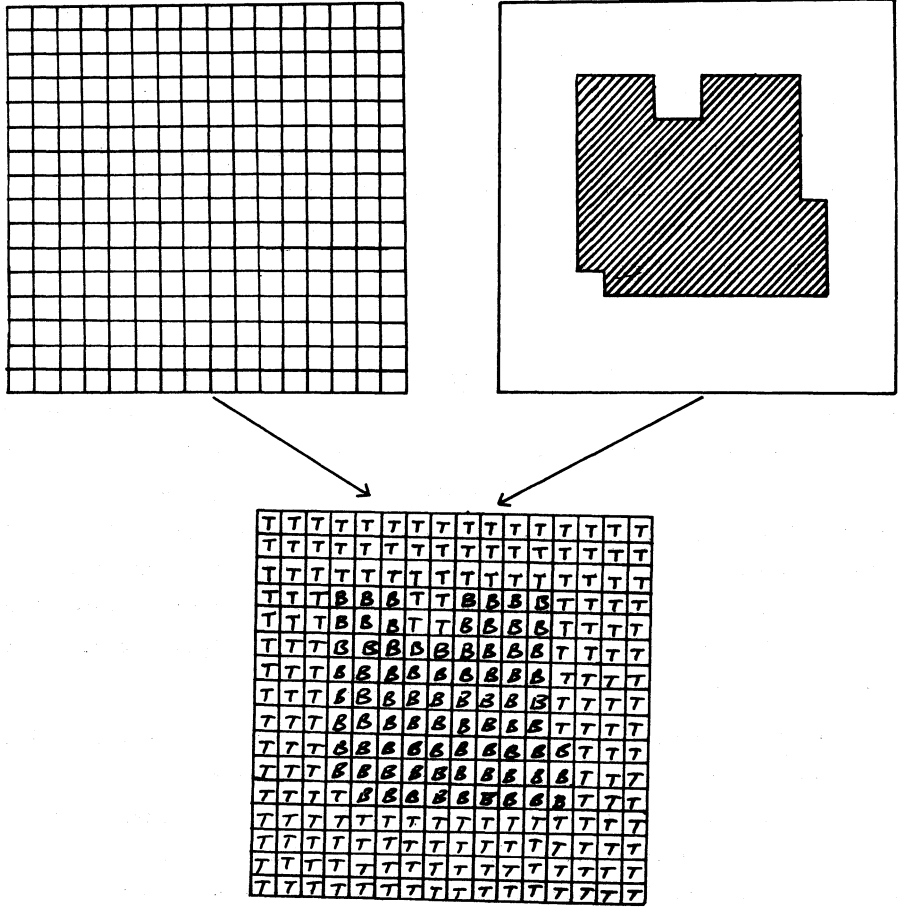
(1) Raster Teknik

İki boyutlu bir şeklin bilgisayarda raster teknikle temsil edilmesinde, şekli içeren grafik alan üzerine bir grid ağı oturtulduğu düşünülür. Grid ağını oluşturan herbir kare "pixel" (benek) olarak adlandırılır. Herbir kare içinde neyin bulunduğu bilgisayarda belli kodlarla (sayılar/harfler) kaydedilir. Başka bir deyişle grid ağı bir matris olarak düşünülür, öyle ki bu matrisin elemanları pixellerdir ve herbir pixelde neyin olduğu kodlanmıştır. Örneğin toprak bir alan üzerindeki bir binanın raster gösteriminde (Şekil-2) toprak için T, bina için B, kodları kullanıldığında bilgisayarda, eleman değerleri ya T ya da B olan bir matris depolanır.

Harita üzerinde bulunabilecek farklı nesne (toprak, bina.....) sayısı ne kadarsa o kadar sayıda farklı koda gereksinim vardır. Dolayısıyla bu farklı kodları depolamak üzere bilgisayar ana belleğinde herbir pixele karşılık yeterli kadar bit ayrılması gerekir. Örneğin herbir pixele karşılık bilgisayar ana belleğinde 8 bit ayrıldığında en çok $2^8 = 256$ farklı nesnenin raster teknikle kodlanması mümkün olabilir.

Şekillerin raster teknikle temsil edilmesi sadece bilgisayar ana belleği için veya bu matrisin depolandığı teyp/disk kütükleri için söz konusu değildir.

Raster teknik aynı zamanda bir grafik veri toplama ve grafik veri gösterim yöntemidir. Raster teknikle grafik veri toplamada harita üzerinde herbir pixele karşılık gelen noktada hangi rengin veya hangi gri tonunun bulunduğu bir algılayıcı ile (CDD: Charge Coupled Device) tesbit edilir ve bunlar sayılar olarak kaydedilir. Bu teknikle veri toplayan cihazlara raster tarayıcı (raster scanner) denmektedir. Aslında uydu sensörleri de benzer prensiple çalışmaktadır.

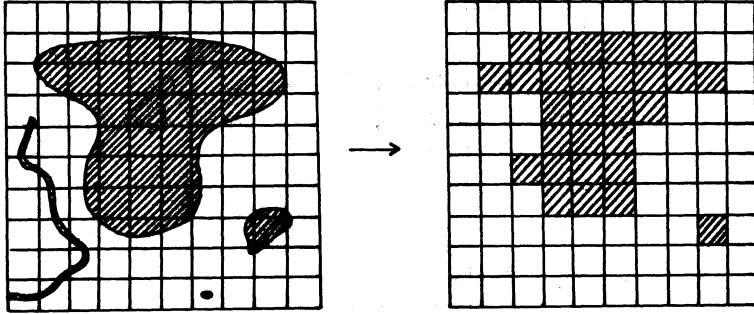


Şekil-2 : Raster Gösterim

Harita üzerindeki detayların ne olduğunu (bina, toprak, su vb.) sayısallaştırma esnasında ayırtedebilen akıllı raster tarayıcılar henüz yoktur. Bu konuda çalışmalar (artificial intelligence, patern recognition) sürmektedir.

Raster teknikle grafik veri gösteriminde ekranda hangi pixelin hangi renkle gösterileceğini veya çizim materyali üzerinde hangi noktaya ne renk nokta konacağını belirleyen bir düzen uygulanır. Raster teknikle grafik veri gösterim cihazlarına örnek olarak renkli grafik ekranlar (raster refresh screens), matris yazıcılar (matrix/dot printers), ink-jet çiziciler (ink-jet plotters) ve elektrostatik çiziciler (electrostatic plotters) sayılabilir.

Raster teknikte bir grid karesinin (pixelin) her yerinde aynı detayın olduğu varsayılır. Buna göre gösterimin inceliği pixel boyutu ile ters orantılıdır. Pixel boyutu büyüdükçe çok küçük detayların ve eğri sınırların temsil edilmesinde sorunlarla karşılaşılır. (Şekil-3)



Şekil-3 Raster gösteriminde pixel boyutuna bağlı incelik

Buna göre özellikle çizgisel haritaların bilgisayarda raster teknikle temsil edilmesinde pixel boyutunun en ince çizgi kalınlığından da küçük tutulması kaçınılmazdır.

(2) Vektör Teknik

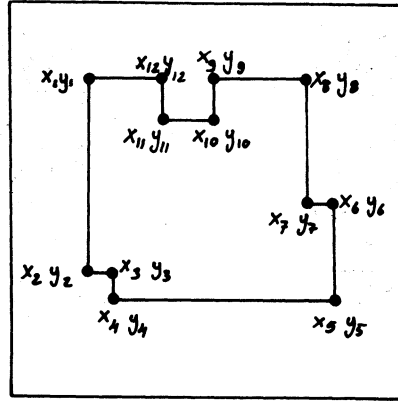
Vektör teknikte detaylar; nokta, çizgi ve alan (poligon) detaylar olarak ele alınırlar. Bazı detaylar veri tabanı ölçeğine göre öyle küçüktürler ki tek bir nokta ile ifade edilirler. Böyle detaylara "nokta detay" denir. Hatta bazen nokta detaylar öyle sıktır ki bunların tek bir nokta detay olarak özetlenmesi gerekebilir. (Birbirine çok yakın kuleler veya enerji nakil hattı direkleri gibi.)

Bazen detayların geometrik biçimleri, yine esas alınan ölçek gereği, çizgi olarak gösterilmelerini gerektirir. Bu gibi detaylara örnek olarak yollar, akarsular, uzun köprüler gösterilebilir. Böylesi detaylara "çizgi detay" denir.

Nokta detaylarda söz konusu nokta, detayın geometrik merkezine karşılık gelirken; çizgi detaylarda sözü edilen çizgi, detayın orta eksenini (yol orta çizgisi, nehir ortası vb.) temsil eder.

Ölçeğe göre nokta olarak alınacak kadar küçük olmayan veya tek çizgi olarak alınacak kadar ince olmayan diğer tüm detaylar ise alan detay olarak ele alınırlar.

Buna göre bilgisayarda bir nokta detay için bir koordinat çifti, bir çizgi detay için bir koordinat çifti dizisi (vektörü) ve bir alan detay için ise bu alanın çevresini oluşturan kapalı poligon yine bir koordinat çifti dizisi (vektörü) olarak temsil edilir. (Şekil-4)



Şekil-4 : Vektör Gösterim

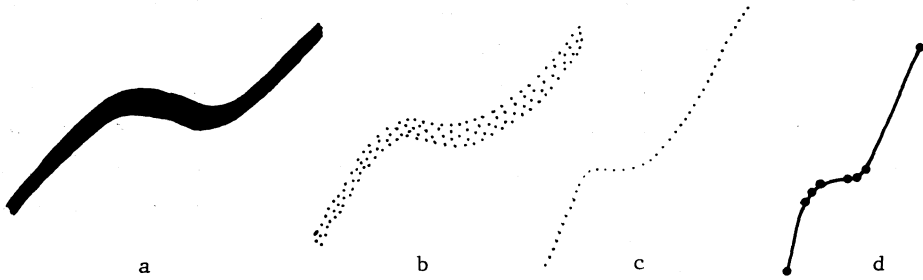
Raster teknikte olduğu gibi vektör teknik de sadece bilgisayar ana belleğinde temsil edilen grafik şekiller veya teyp/disk üzerindeki grafik kütükler için söz konusu değildir. Vektör teknik aynı zamanda bir grafik veri toplama ve bir grafik veri gösterim yöntemidir.

Vektör teknikle grafik veri toplamada en çok manuel sayısallaştırıcılar (manuel digitizers) ve stereo sayısallaştırıcılar (stereo plotters) kullanılır. Manuel sayısallaştırıcılarda özde sayısallaştırma masası üzerindeki bir koordinat sistemine göre sayısallaştırma cursor'ının bulunduğu noktanın (x,y) masa koordinatları bilgisayarda sayısallaştırma masasını süren bir program ile oku-

nur ve disk üzerindeki bir kütüğe yazılır. Böylece sayısallaştırma sonucunda disk üzerinde vektör yapılı bir grafik kütük elde edilmiş olur. Burada sayısallaştırma masasının ayırma gücü (resolution) raster teknikte olduğu gibi veri kaybına neden olmaz, ancak kaydedilen koordinatların duyarlılığını etkiler.

Stereo sayısallaştırıcılar ise, fotogrametride kullanılan sayısal çıkışlı analog veya analitik aletlerden oluşmaktadır. Bunlarda veri toplama yöntemi genel olarak manuel sayısallaştırıcılarda olduğu gibidir.

Vektör türündeki veriler, raster verilerin vektöre dönüştürülmesiyle de elde edilebilmektedir. Raster veriden vektöre dönüşüm iki aşamalıdır. Birinci aşamada, pixel koordinatları atılarak inceltilme (Thinning) işlemi gerçekleştirilir. (Şekil-5.....a,b,c)



Şekil-5 : Raster-Vektör Dönüşüm

İnceltilmiş bu raster verilerin sıklığı, detayın çizgi ile belirtilmesinde yine de fazladır. Bu nedenle ikinci bir algoritma ile detayı en uygun belirleyecek koordinatlar alınarak diğerleri atılır. (Şekil-5.d) Buna ayıklama (Weeding) adı verilir.

Vektör teknikle grafik veri gösteriminde en yaygın uygulama kalem tipli çizicilerdir (pen plotters). Bu tip çizicilerde iki eksen üzerinde hareket etmesi sağlanan çizim başlığındaki kalem, çizim masası üzerinde kurulu bir koordinat sisteminde belli bir (x,y) noktasına gitme, aşağı inme, yukarı kalkma gibi hareket olanaklarına sahiptir. Buna göre kalem aşağıda iken (masa üzerindeki kağıda değerken) çizim başlığı başka bir noktaya doğru gittiğinde, çizgi çizilmiş olur. Çizim başlığı her iki ekseninde hareket edebilen çiziciler (flat-bed pen plotters) olduğu gibi çizim başlığının tek eksen üzerinde hareket edebildiği çiziciler (drum plotters) de mevcuttur. Diğer bir vektör teknikli grafik veri gösterim cihazı tek renkli vektör ekranlardır. Burada çizim masasının yerini fosforlu bir ekran, kalemin yerini de bu ekran üzerinde iz bırakabilen ışın hüzmesi almıştır.

Sayısal coğrafya bilgisi üretmek ve kullanmak üzere bir konfigürasyon belirlerken raster veya vektör tekniklerinin hangilerinin kullanılacağı dikkatli bir analiz ile belirlenmelidir. Ancak tüm konfigürasyonda tek bir teknik kullanma zorunluluğu yoktur. Önemli olan amaca uygun bir kombinasyon teşkil etmektir. Örneğin disk üzerindeki grafik veri kütükleri vektör yapıda iken çizici olarak raster yapıli elektrostatik plotterler kullanılabilir veya grafik veri toplama birimleri olarak vektör yapıli manuel sayısallaştırıcıların yanısıra raster tarayıcılar da kullanılabilir. Aradaki dönüşümler yazılımlarla gerçekleştirilir. Bu nedenle böylesi kombine konfigürasyonlarda rasterdan vektöre ve vektörden rastera dönüşüm programları yer alır.

e. Öznitelikler

Coğrafi veri tabanlarında nesnelerin konum ve biçimlerini gösteren vektörel veya raster yapıdaki grafik bilgilerin yanısıra, grafik olmayan bilgilerde bulundurulur. Münhaniler veya nokta yükseklikler (tepe noktaları, çukur gibi vb.) için bu bilgi sadece yükseklik değerinden oluşurken detaylar için birçok başka bilgilerinde veri tabanında bulundurulması gerekir. Bu bilgiler detayın "öznitelikleri" olarak adlandırılır. Hangi cins detay için hangi özniteliklerin tutulacağı bir tasarım sorunudur. Örneğin bir ormanlık alan detayı için bu alandaki ağaç cinsi, ağaç sıklığı, ortalama ağaç yaşı, ortalama yıllık verimi onun öznitelikleridir. Veya bir yol detayı için yapı malzemesi, genişliği, şerit sayısı özniteliklerdir.

Bu noktada önemli bir prensip ortaya çıkmaktadır. "Detayın grafik olarak ifade edilemeyen özellikleri mutlaka öznitelik olarak kaydedilmelidir". Örneğin bir bina ölçek gereği nokta detay olarak sayısallaştırılmışsa, bu binanın geometrik bilgileri (eni, boyu, yüksekliği ve yönü) öznitelik olarak kaydedilmelidir. Eğer bina büyükse ve alan detay olarak sayısallaştırılmışsa, sadece yüksekliğinin (tabi diğer özniteliklerin yanısıra) kaydedilmesi yeterlidir. Aynı şekilde bir yol veya uzun bir köprü çizgi detay olarak alınmışsa "genişlik" öznitelik olarak kaydedilir.

f. Coğrafi Hücre

Bir grafik kütük belli bir çerçeve içindeki münhanileri, detayları içerir ve bu hali ile bir sayısal pafta biçimindedir. Ancak kenarlaştırma ve birleştirme yazılımları ile komşu sayısal paftaları birleştirmek ve pafta sınırlarını ortadan kaldırarak sürekli bir coğrafi veri tabanı elde etmek mümkündür. Fakat böylesi bir veri tabanında bile veri yönetiminde ve özellikle veriye erişimde kolaylık sağlamak üzere coğrafi veri tabanı bölümlere ayrılır. Genellikle

enlem boylam çizgileri ile ayrılan bu bölümlere "coğrafi dörtgen" veya "coğrafi hücre" denir.

g. Detay tipi, detay kodu, detay numarası

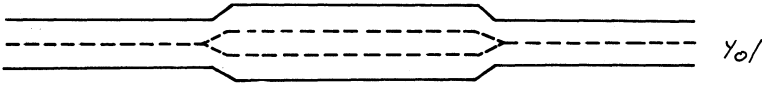
Vektörel bir coğrafi veri tabanında detay tipi detayın nokta, çizgi veya alan (poligon) detay olduğunu belirtir.

Detay kodu bir detayın "ne" olduğunu gösteren koddur. Her coğrafi veri tabanı için bir detay kodlama katalogu gereklidir.

Detay numarası ise detayın "hangi" detay olduğunu gösterir. Bir detayı aynı veya farklı cins diğer detaylardan ayırd etmeye yarayan detay numarası her coğrafi hücre içinde tek anlamlıdır.

h. Detay Bileşeni

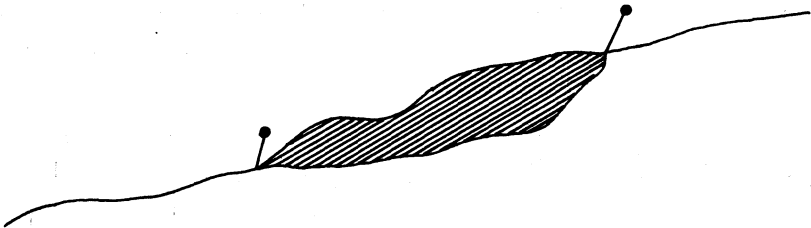
Özellikle çizgi ve alan detaylar için kaydedilen öznelik değerleri detayın tümü için geçerli olmalıdır. Ancak örneğin bir yol detayını ele alırsak ve diyelim 2 şeritli olan bu yol belli bir bölümde genişliyor ve 3 şeritli oluyorsa, bu durumda her parçayı ayrı ayrı düşünmek gerekir. (Şekil-6)



detay bileşeni - 1 • *detay bileşeni - 2* • *detay bileşeni - 3* — *Çizgi detay*

Şekil-6 : Detay Bileşenleri

Bu durumda daha önce yapılan tanımları biraz geliştirerek "detay"ın tematik bir kavram olduğunu ve "detay bileşenlerinin" (grafik bileşenlerin) birleşiminden oluştuğunu söyleyebiliriz. Bir detaya ait detay bileşenlerinin aynı detay tipinde olması da gerekli değildir. Örneğin arada küçük bir göl meydana getiren bir nehir için çizgi ve alan detay bileşenlerinin birleşiminden oluşuyor diyebiliriz. (Şekil-7)



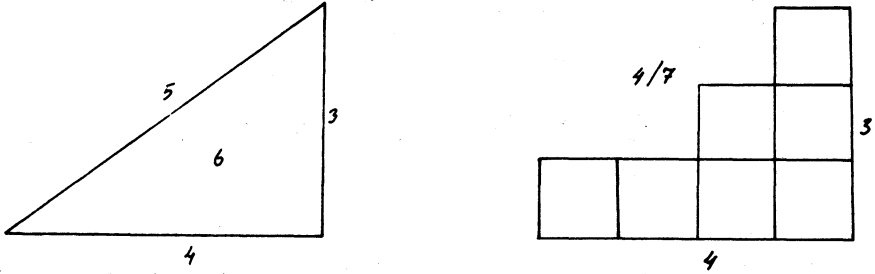
Şekil-7 : Farklı tipte detay bileşenleri

Bilgisayar Destekli Kartografya ve Coğrafi Bilgi Sistemlerindeki gelişmelerin beraberlerinde getirdikleri şüphesiz pek çok yeni kavram daha vardır. Ancak bu yazıda 3 ve 4 ncü bölümde açıklanan coğrafi veri yapılarına altlık teşkil edecek olanlarla yetinilmiştir.

3. RASTER VERİ YAPILARI

2,d. paragrafında temel prensipleri kısaca açıklanan raster teknikte en basit veri yapısı, satır ve sütunlardan oluşan matris yaklaşımıdır. Matris veri yapısı, özellikle FORTRAN gibi bir dille bilgisayarda işlenmesi çok uygun bir yapıdır. Ancak matris biçimli bir veri yapısının getirdiği bazı sınırlamalar vardır.

* Coğrafi verilerin üzerinde yayıldıkları iki boyutlu yüzey sürekli değil fakat sayılabilir niteliktedir. Bu nedenle yüzey ve alan hesaplarında pixel boyutuna da bağlı olan sapmalar olmaktadır. Örneğin (Şekil-8) daki dik üçgenin raster gösteriminde hipotenüs değeri pixel kenar sayısı olarak 7, pixel sayısı olarak 4 hesaplanmaktadır. Aynı şekilde 6 pixel kare olması gereken üçgen alanı 7 pixel kare olmaktadır.

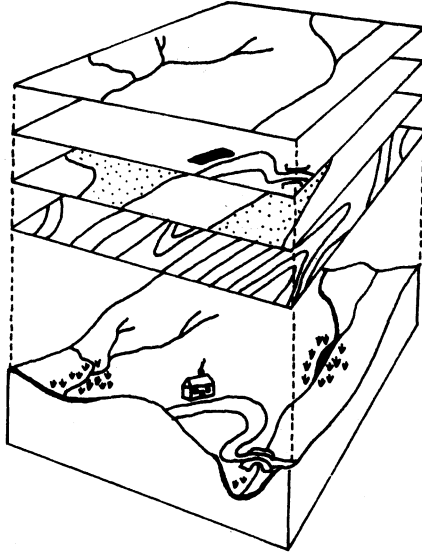


Şekil-8 : Raster kodlamada pixel saymaya dayalı uzunluk ve alan hesabında yanlıgılar.

Bu sakıncayı gidermenin tek yolu pixel boyutunu mümkün olduğunca küçük tutmak ve böylece gerek hesaplamalarda gerekse görüntülerde belirli sınırlar içinde gerçeğe yakın sonuçlara ulaşmaktır. Günümüzde ekran ve çizici teknolojisi bu bakımdan yeterli düzeye ulaşmıştır.

* Her matris elemanına (piksel veya grid karesine); ancak tek bir sayı atanabilmektedir. Bu nedenle örneğin toprak cinsi, yeraltı su seviyesi ve toprak üstü bitki örtüsü cinsi gibi aynı noktaya ait birden fazla coğrafi verinin tek bir matris elemanı ile gösterimi mümkün değildir.

Bu sakıncayı gidermek için de "harita katmanları" kavramı geliştirilmiştir. Buna göre bir harita paftası üst üste dizili matrislerle temsil edilir. Herbir matris üzerinde belli bir sınıfa ait detaylar yer alır. Buna göre bir harita örneğinin topografya, hidrografiya, ulaşım, yerleşim ve tesisler, enerji, iletişim, toprak cinsi vb. katmanlar biçiminde ele alınır. (Şekil-9)



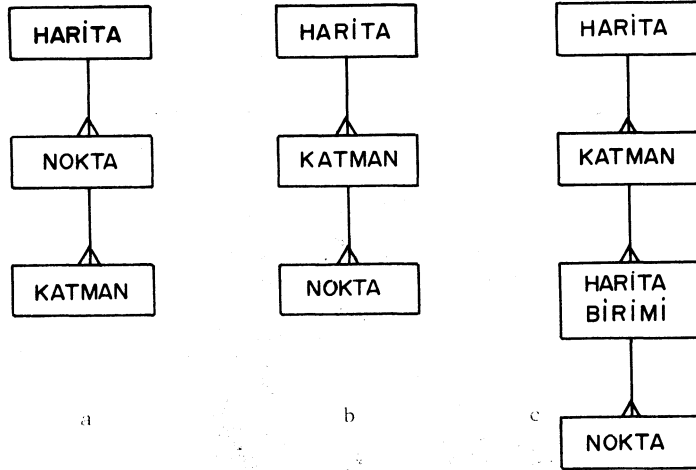
Şekil-9 : Harita Katmanları Kavramı

Bu katmanların herbiri bilgisayarda ayrı birer matris olarak temsil edilir. Ancak bu durumda yeni bir sorunla, bellek sorunu ile karşı karşıya kalınır. 50x70 cm. ebadındaki bir harita paftasının bilgisayarda 0.1 x 0.1 mm.lik pixellerle temsil edildiğini varsayarsak tek bir katmandaki matriste 35 milyon eleman bulunacaktır. Her bir eleman için 1 bit (detay var: 1, detay yok:0) kullanıldığında bir katman için 4.5 MByte, 8 bit kullanıldığında (256 detay kodu) bir katman için 35 MByte bellek gereksinimi olacaktır. Bu rakamlar bir de katman sayısı ile çarpıldığında büyük bir bellek sorunu ile karşı karşıya kalınacaktır.

Detaylara erişimi optimize etmek ve bellek sorununu çözmek üzere sırasıyla "raster mantıksal veri yapıları" ve " raster veri depolama yöntemleri" geliştirilmiştir.

a. Raster Temsilde Mantıksal Veri Yapıları

Raster teknikli bir coğrafi veri tabanındaki verilere erişimi optimize etmek üzere veriler çeşitli biçimlerde organize edilebilirler.



Şekil-10 : Raster Temsilde mantıksal veri yapıları

Mantıksal veri yapısı şekil-10.a'da gösterilen çözümde her bir harita kü-tüğünde birden çok nokta kaydı vardır. Her noktada ise birden çok katmana i-lişkin bir dizi bulunduğu düşünülür. Buna göre harita üzerinde coğrafi veri bulunan her nokta için (x,y, katman no. ,z) biçiminde kayıtlar tutulur.

İkinci çözümde ise bir harita kü-tüğü önce katmanlara ayrılmış olarak dü-şünülür ve her bir katmanda coğrafi veri bulunan noktalar kaydedilir.(Şekil-10.b) Böylece yine her bir coğrafi veri için (katman no. x,y,z) biçiminde ka-yıtlar tutulur.

Diğer bir çözüm ise "Harita birimi" kavramına dayanır. Harita birimi aynı coğrafi veri değerine sahip noktalar kümesidir.(Şekil-10.c) Buna göre veri ta-banında bu sefer dört değişik tip kayıt tutulur, Harita kayıtlarında haritaya özel bilgiler (sayısallaştırma tarihi vb.) kaydedilir. Katman kayıtlarında han-gi haritaya ait olduğu, katman ismi, ölçeği vb. bilgiler tutulur. Son olarak da her bir harita biriminin hangi noktalarda yer aldığı kayıtlıdır.

Başka bir anlatımla ilk iki veri yapısında "hangi noktalarda nelerin bulun-duğu" kayıtlı iken üçüncü çözümde "nelerin hangi noktalarda bulunduğu" (örneğin meşe ormanının hangi noktalarda olduğu) kayıtlıdır.

b. Raster Veri Depolama Yöntemleri

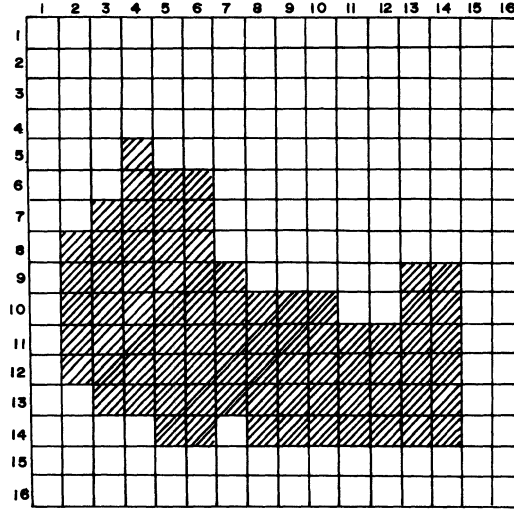
Herhangi bir katmanda aynı öznelik değerine sahip alanları en az bellekle temsil edebilmek üzere karmaşık veri depolama yöntemleri geliştirilmiştir. Kullanılan tekniklerin başlıcaları aşağıda açıklanmıştır.

(1) Zincir kodları (Chain Codes) Yöntemi

Zincir kodları yönteminde bilgisayarda kodlanacak olan bölgenin sınırları belli bir orijinden başlanarak ana yönlerdeki birim vektörler dizisi halinde saat yönünde tanımlanır. Yönler (doğu=0, kuzey=1, batı=2, güney=3) biçiminde kodlanır. Tekrarlı yönler üs olarak belirtilir. Örneğin şekil-11 deki taralı bölge 8 nci satır, 2 nci sütun orijinininden başlanarak :

$0,1,0,1^2,0,3,0^2,3^3,0,3,0^3,3,0^2,1^2,0^2,3^6,2^7,1,2,3,2^2,1,2^2,1,2,1^5$

olarak kodlanır.



Şekil-11 : Zincir Kodları

Ana yön kodu için 2 bit, üs için 3 bit kullanıldığında düşünülürse bu şeklin bilgisayarda temsili için $26 \times 5 = 130$ bit = 17 byte yeterli olacaktır. Oysa matris biçimli raster kodlamada her eleman için 2 bit harcanacağından $16 \times 16 \times 2 / 8 = 64$ byte gerektirdi.

Zincir kodlama yöntemi bellek tasarrufu açısından çok iyi sonuçlar vermekte ve ayrıca çevre ve alan hesabı işlemlerini kolaylaştırmaktadır. Ancak kesim ve birleşim işlemleri için matris biçimli kodlamaya dönülmesi gerekir. Diğer bir dezavantaj da komşu bölgelerin kodlanmasında ortak sınırların iki kez kodlanmasıdır.

(2) Eş-Tarama Uzunluğu Kodları (Run-length Codes) Yöntemi

Eş tarama uzunluğu yönteminde kodlanacak bölge (harita birimi) satırlar halinde ele alınır. Herbir satırda aynı değere sahip (tarama sonunda aynı değeri yansıtan) pixellerin hangi sütunda başlayıp hangi sütunda bittikleri kaydedilir.

Buna göre Şekil-11 deki taralı bölge :

Satır	5	4,4	
Satır	6	4,6	
Satır	7	3,6	
Satır	8	2,6	
Satır	9	2,7	13,14
Satır	10	2,10	13,14
Satır	11	2,14	
Satır	12	2,14	
Satır	13	3,14	
Satır	14	5,6	8,14

Biçiminde kodlanır. Satır veya sütun numaralarının herbiri için 4 bit kullanıldığı düşünülürse $36 \times 4 / 8 = 144$ bir=18 byte yeterli olacaktır.

Önemli miktarda bellek tasarrufu sağlayan bu yöntem özellikle veri hacminin sınırlı olduğu durumlarda ve küçük bilgisayarlarda avantajlıdır. Ancak böylesi çok fazla veri sıkıştırma kartografik işlemler sırasında işlem zamanı gerektirir. Ayrıca şekiller küçük olduğunda bellek gereksinimi artar. Bu yöntem özellikle raster tarama sırasında raster kütüğe giris için kullanılmaktadır.

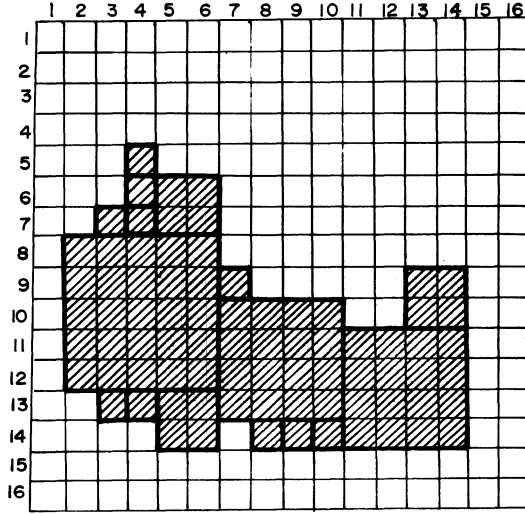
(3) Blok Kodları (Block Codes) Yöntemi

Eş-tarama uzunluğu düşüncesi iki boyuta yayılarak eşit değerde pixel-lerden oluşan kare bloklarının kodlanması biçiminde bir yönteme dönüşmektedir. Bu veri yapısında her bir kare blok üç sayı ile (karenin orijini yani sol üst köşenin satır-sütun numarası ve kenar pixel sayısı) kodlanmaktadır. Karelerin mümkün olduğunca büyük olması bellek gereksinimini azaltmaktadır.

Bu yönteme göre aynı şekil 1 adet 5 li, 2 adet 4 lü, 3 adet 2 li ve 10 adet 1 li kare ile temsil edilebilmektedir. Her sayı için 4 bit kullanıldığında $16 \times 3 \times 4 / 8 = 24$ byte gerekli olacaktır.

Blok kodları özellikle büyük şekillerin kodlanmasında bellekte önemli ölçüde tasarruf sağlamaktadır. Bu arada birleşim ve kesişim işlemleri için belli

avantajlar sunmaktadır. Ancak yine de bir çok işlem için orijinal matris yapısına dönmek zorunlu olmaktadır.



Şekil-12 : Blok Kodları

(4) Dörtlü Ağaç Yapı (Quad-tree)

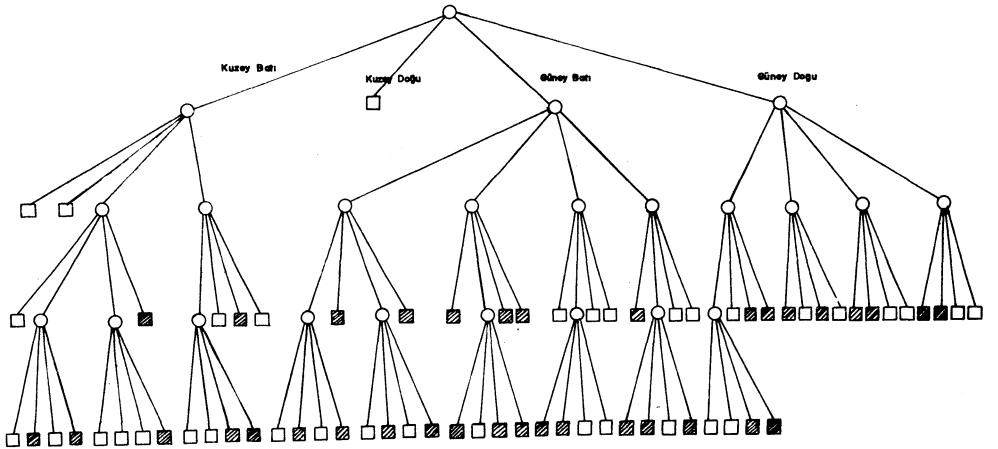
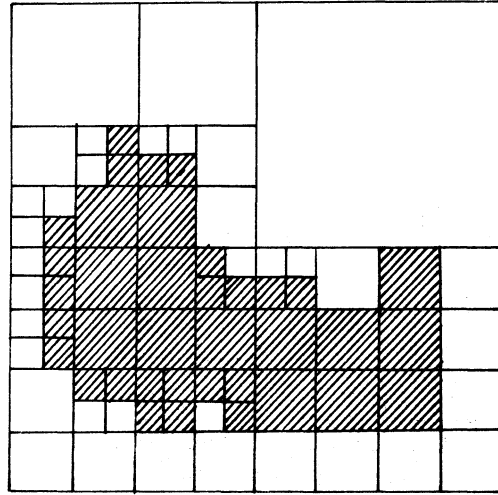
Bu veri yapısında 2x2 boyutlu dizi aşamalı olarak dördüllere ayrılır. Adım adım dördüllere ayırma işlemi bölünecek alanın tamamen dolu veya tamamen boş kalmasına kadar sürdürülür. Dörde bölme işlemi en çok pixel boyutuna kadar inebilir. Böylece her düğümde 4 dalı olan bir "dörtlü ağaç yapı" elde edilmiş olur. Herbir düğümün temsili için 2 bit yeterlidir. (11:dolu, 00:boş, 10:bölünme devam, 01:bölünme bitti.)

Bu yöntemle kodlamada aynı şekil için sadece $23 \times 4 / 8 = 12$ byte yeterli olmaktadır.

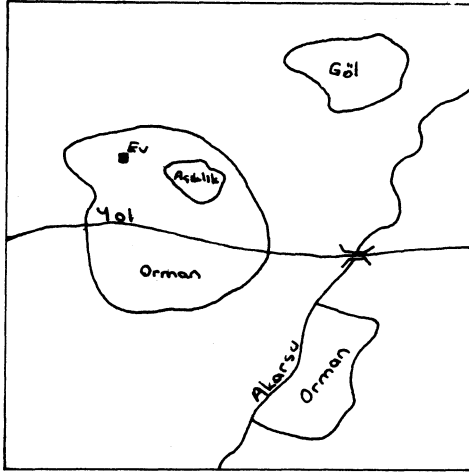
Dörtlü ağaç yapı birçok algorithmaya uygunluğu nedeniyle çokca kullanılmaktadır. Bölgeye göre değişken ayırma gücü önemli bir avantajdır.

4. VEKTÖR VERİ YAPILARI

Bu bölümde, detaylar için vektörel coğrafi veri yapıları basitten gelişmişe doğru beş ayrı aşama halinde açıklanacaktır. Bugünkü vektörel coğrafi veri tabanlarında kullanılan veri yapılarının küçük bazı farklılıklarla bu beş yapı modelinden, özellikle 2,3,4 ve 5 ncisinden birisine uyduğu söylenebilir. Açıklamalarda örnek olarak Şekil-14 de görülen sembolik harita parçası kullanılacaktır.



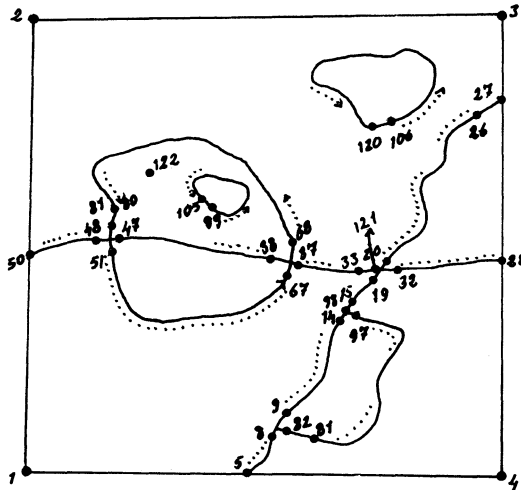
Şekil-13 : Dörtlü ağaç yapısı



Şekil-14 : Örnek Çizgisel Harita

a. Yapısallaştırılmamış Vektör Veri

Bir manuel sayısallaştırıcımızın olduğunu düşünelim. Ancak sayısallaştırma yazılımımızın veri yapısallaştırma özelliği olmasın, sadece yeni bir detaya (çizgiye) başladığımızı belirtebiliyoruz. Ancak detay kodlarını veremiyoruz. Programı çalıştırıp kütüğü ilk açtığımızda (sayısallaştırmaya başladığımızda) program ilk detaya 1 numarasını veriyor ve sonraki detaylar için bu numarayı sırayla artırıyor. Bu program ile örnek haritamız Şekil-15 deki gibi nokta-nokta sayısallaştırıldığında Şekil-16 da görülen kütük elde edilir.

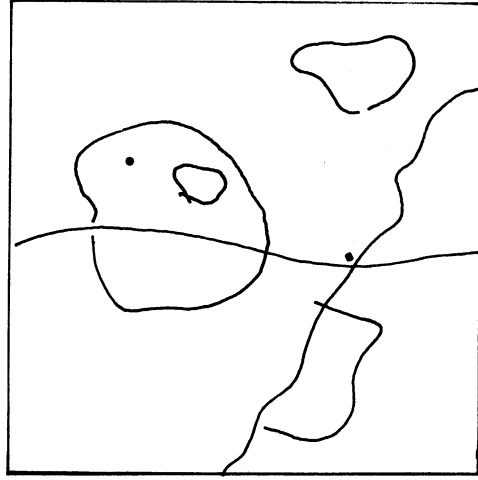


Şekil-15 : Manuel Sayısallaştırma

1	1	
2	x_1	y_1
3	x_2	y_2
4	x_3	y_3
5	x_4	y_4
6	2	
7	x_5	y_5
.	.	.
.	.	.
29	x_{27}	y_{27}
30	3	
31	x_{28}	y_{28}
.	.	.
.	.	.
53	x_{50}	y_{50}
54	4	
.	.	.
.	.	.
126	x_{119}	y_{119}
127	x_{120}	y_{120}
128	8	
129	x_{121}	y_{121}
130	9	
131	x_{122}	y_{122}

Şekil-16 : Yapisallaşmamış coğrafi veri kütüğü

Hemen söylenebilir ki sonuçta elde edilen 131 kayıtlık bu kütük olduğu gibi çizimden başka hiçbir işe yaramaz. Çizim için kullanılabilecek program ancak ve ancak her detayı ilk noktasından son noktasına kadar bir çizgi (veya nokta) halinde çizecektir. Kütükte detaylara ait koordinatlar (grafik veriler) ve sadece detay numaraları (grafik olmayan veriler) bulunduğundan, detay kodları bulunmadığından; çizimde farklı renkler veya semboller kullanmak, detayların alan mı, çizgi mi olduklarını ayırmak mümkün değildir. Bu nedenle, elde edilecek tek renkli çizim Şekil-17 deki gibi olacaktır.



Şekil-17 : Yapısallaştırılmamış veri kütüğünden
alınan çizim

Sorunlar :

- * Alan detaylar kapanmamaktadır. (Orman, açıklık, göl)
- * Temas noktaları çakışmamaktadır. (Yol-çerçeve, akarsu-çerçeve, akarsu-orman)
- * Kesişme noktalarındaki nokta detaylar çakışmamaktadır. (Akarsu-yol-köprü)
- * Detayların ne olduğu belli değildir.

Aslında bu veri yapısı modeli-bilindiği kadarıyla-bugün hiç kullanılmamaktadır. Ancak sorunları açıkça göstermesi bakımından burada yer verilmiştir.

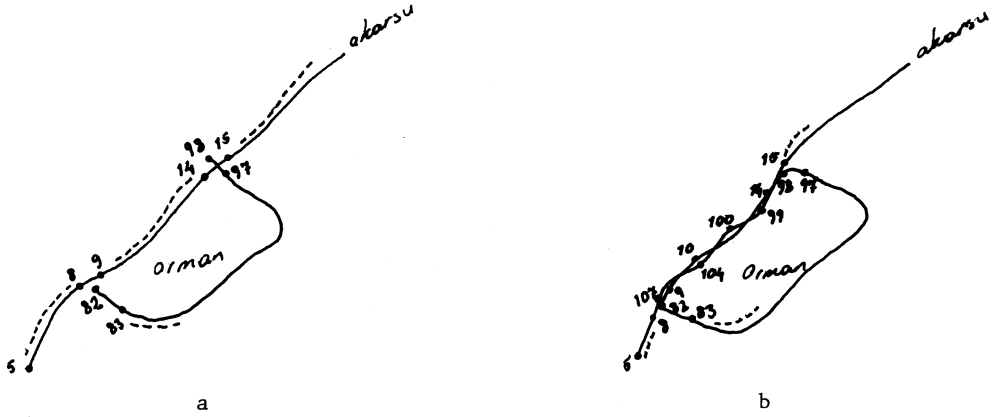
Geometrik sorunlar olarak adlandırılabilir kaplanmama, kesişmeme, çakışmama sorunlarının giderilmesi bu veri yapısında mümkün değildir.

b. Sphagetti Veri Yapısı

Sphagetti veri yapısı, yapısallaşmamış veriye göre bazı iyileştirmeler içermektedir. En önemli iyileştirme "detay" kavramının eklenmesidir. Yapısallaşmamış

veride grafik verilere bir çizgi ve nokta topluluğu gözüyle bakılırken, sphagetti veri yapısındaki grafik veri nokta, çizgi ve poligon detaylar kümesi biçiminde ele alınır. Her detay kendi içinde bir bütündür ve detaylar birbirinden bağımsızdır. Komşu veya çakışık detaylar ayrı ayrı sayısallaştırılır. Buna göre; alan-alan, alan-çizgi, alan-nokta, çizgi-çizgi, çizgi-nokta, nokta-nokta çakışıklıklarında ve komşuluklarında ortak geometrik eleman birden fazla sayısallaştırılır.

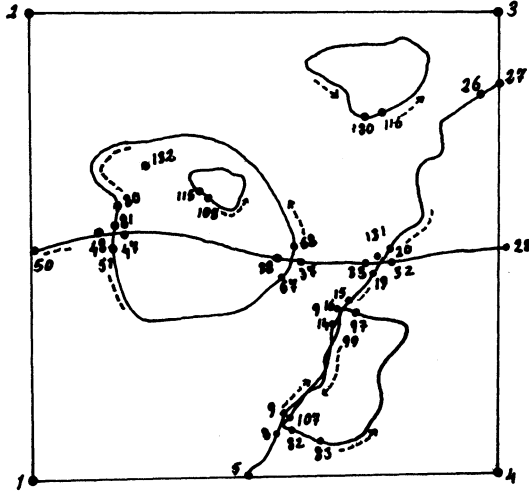
Örneğin Şekil-14 deki örnek haritada yer alan akarsu ve ormanlık alan birbirine komşudur. Ormanlık alanın sınırının bir bölümü aynı zamanda akarsu çizgisinin de bir bölümünü oluşturmaktadır. Yapisallaştırılmamış veri yapısında bu bölüm Şekil-18.a daki gibi sayısallaştırılırken, sphagetti veri yapısında Şekil-18.b deki gibi sayısallaştırılmaktadır.



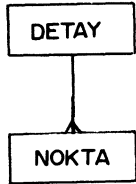
Şekil-18 : Yapisallaşmamış veri ve sphagetti veri yapısında alan-çizgi komşuluğu

Örnek haritamız sphagetti veri yapısında Şekil-19 da görülen noktalarla sayısallaştırıldığında Şekil-20 deki gibi bir kütük elde edilir.

Şekil-21 den de görüldüğü gibi sphagetti veri yapılı bir grafik kütükte iki değişik tip kayıt vardır. Bunlardan birisi detay başlık kayıdır. Her detay için bir tane detay başlık kaydı bulundurulur. Detay başlık kaydında detay numarası, detay kodu, detay tipi (Alan, Çizgi, Nokta) gibi bilgiler bulunur. Bu bilgiler etkileşimli sayısallaştırma sırasında girilebileceği gibi, sonradan etiketleme sırasında da girilebilir.



Şekil-19 : Sphagetti Veri Yapısında
Sayısallaştırma



Şekil-21 : Sphagetti veri yapısı
varlık-ilişki diagramı

1	CRÇ	A
x_1	y_1	
x_2	y_2	
x_3	y_3	
x_4	y_4	
-999	-999	
2	AKR	Ç
x_5	y_5	
:	:	
x_{27}	y_{27}	
-999	-999	
3	Yol	Ç
x_{28}	y_{28}	
:	:	
x_{50}	y_{50}	
-999	-999	

x_{116}	y_{116}	
:	:	
x_{130}	y_{130}	
-999	-999	
8	KPR	N
x_{131}	y_{131}	
9	EV	N
x_{132}	y_{132}	

Şekil-20 : Sphagetti veri yapılı
grafik kütük

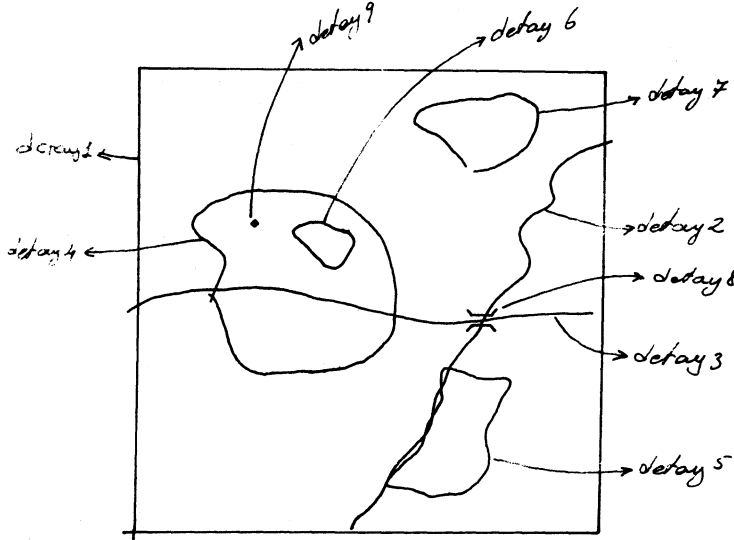
Çözümler :

Yapısallaştırılmamış coğrafi verilere göre sphagetti veri yapısında bir takım sorunlar çözülmüştür.

* Detayların ne oldukları (detay kodları) bellidir. Buna göre gösterimlerde ve çizimlerde detayların cinsine göre farklı renkler veya semboller kullanılabilir.

* Detay tipleri (alan/çizgi/nokta) bellidir.

Buna göre sphagetti veri yapılı bir kütükten alınacak çizim Şekil-22 deki gibi olacaktır.



Şekil-22 : Sphagetti veri yapısından
alınan çizim

Sorunlar :

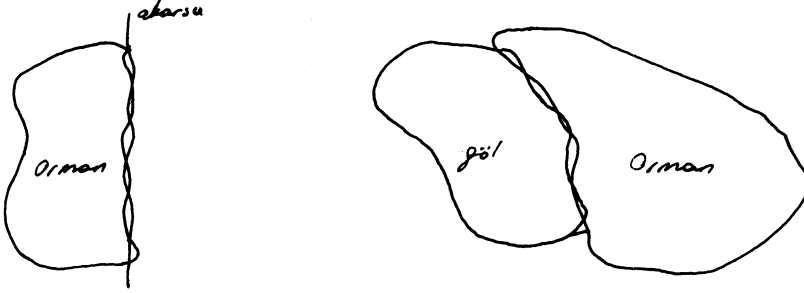
(1) Geometrik Sorunlar :

(a) Alan detaylardan kapanma sorunları vardır. (Şekil-23)



Şekil-23 : Alan Detaylarda Kapanma Sorunu

(b) Her detay ayrı ayrı sayısallaştırıldığından ortak geometrik elemanlar birden fazla yerde farklı noktalarla kayıtlıdır. Buna göre alan-çizgi alan-alan komşuluklarında binme ve boşluklar vardır. (Şekil-24)



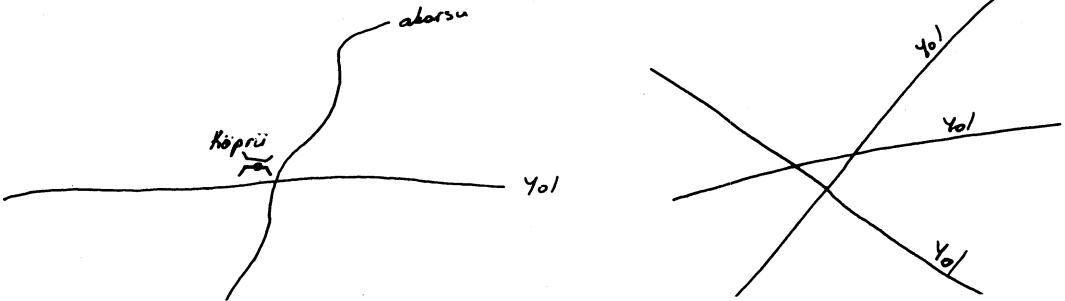
Şekil-24 : Binme ve Boşluklar

(c) Temas noktalarında (yol-çerçeve, akarsu-çerçeve vb.) kopukluk ve taşmalar vardır. (Şekil-25)



Şekil-25 : Kopukluk ve Taşmalar

(d) Kavşak noktasındaki çizgi ve nokta detaylar aynı noktada birleşmemektedir. Örneğimizde yol ve akarsu kavşağında olması gereken köprü ayrıca bir nokta olarak sayısallaştırıldığından sayısallaştırmada yapılan kaçınılmaz hata kadar farklı yerdedir. (Şekil-26)

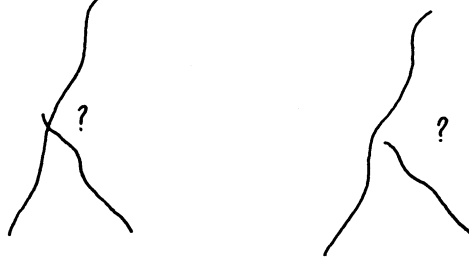


Şekil-26 : Kavşak Noktası Hataları

(2) Diğer Sorunlar

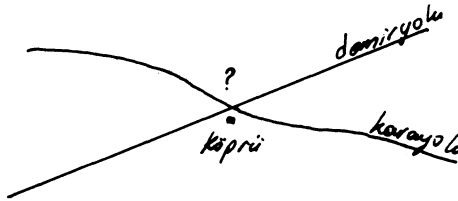
Geometrik sorunların yanısıra veri yapısının "akıllı" olmayışından kaynaklanan diğer bazı sorunlar daha vardır:

(a) Bir akarsüda kolların birleştiği noktayı tanımlamak mümkün değildir. (Şekil-27) Çünkü kavşak ve temas noktaları ayrıca tanımlı değildir.



Şekil-27 : Tanımlanmamış kesişim veya temas
Noktaları

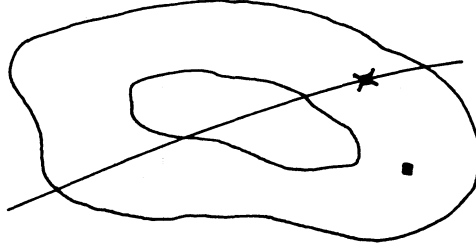
(b) Üç boyutla ilişkileri belirlemek mümkün değildir. Örneğin bir karayolu ile bir demiryolunun kesiştiği yerde hangisinin üstte hangisinin altta olduğu belirlenememektedir. (Şekil-28)



Şekil-28 : Üç Boyutta Belirlenemeyen İlişkiler

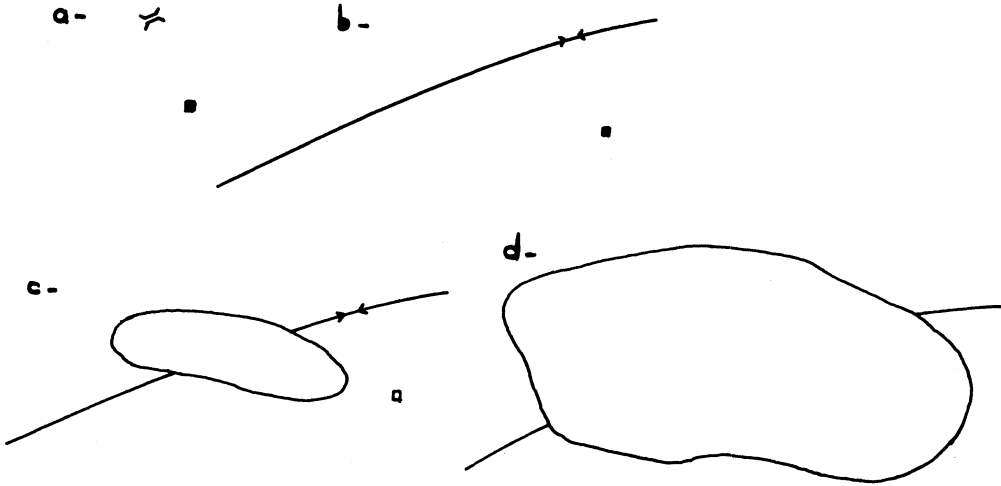
(c) Örtme ve içerme sorunu : Alan detaylar çevreleri ile doğrusal olarak tanımlanmıştır. Bu durumda "detay sayısallaştırma yönüne göre hangi taraf-tadır ?" , "bu alan içinde başka (alan/çizgi/nokta) detaylar da var mıdır?" gibi sorular gündeme gelir.

Bu sorulardan ilkinin gidermek üzere alan detaylar sayısallaştırılırken sayısallaştırma yönü detaylar hep aynı tarafta kalacak şekilde seçilir. Buna göre uygulamada alan detaylar daima solda kalacak şekilde sayısallaştırılmaktadır. Ancak sorulardan ikincisi (içerme) çizimlerde veya ekranda gösterimlerde detayların birbirini örtmesine neden olur. Örnek olarak bir orman alan içinde açıklık olduğunu ve bu orman içinden bir yolun geçtiğini düşünelim. (Şekil-29) Yol orman içinde küçük bir köprüden geçiyor ve ağaçlar arasında bir de ev var.



Şekil-29 : Alan Detay İçinde Diğer Detaylar

Sayısallaştırmanın köprü, ev, yol, açıklık ve orman sırasında yapıldığını düşünelim. Bu durumda sphagetti yapıdaki grafik kütükdeki detaylara ait kayıtlar da aynı sırayı takip edecektir. Çizim veya ekranda görüntüleme programı da bu kütükdeki kayıtları aynı sırada okuyarak çizicide çizecek veya ekranda görüntüleyecektir. Önce köprü ve ev çizilecek ancak yol çizildiğinde köprü örtülecektir. Daha sonra açıklık alan çizilip içi kahverengi (toprak) ile doldurulduğunda yolun bir kısmı kaybolacaktır. Son olarak da orman alan sınırı çizilip içi yeşille doldurulduğunda orman içindeki herşey gözden kaybolacaktır.(Şekil-30)



Şekil-30 : Detayların Çiziminde veya ekranda Gösteriminde Birbirini Örtmesi

Örtme olarak bilinen bu sorunu gidermek üzere detaylar arasında alan-detay-çizgi detay-nokta-detay biçiminde bir hiyerarşi uygulanır ve sayısallaştırma bu sırada yapılır. Alan detaylar kendi içinde dıştan içe olarak sıralanır. Detay numaralamada da bu sıraya uyulur. Böylece numara sırası ile çizimde örtme sorunu olmaz. Ancak detay sayısallaştırmada uyulması gereken bu sıra sayısallaştırma operatörüne ek bir yük getirmektedir.

Bütün bu sorunlarına rağmen sphagetti veri yapısı günümüze kadar (ve hatta günümüzde) sayısallaştırmalarda en çok kullanılan bir veri yapısıdır. Intergraph'ın IGDS, Laser-Scan'in IFF, Synercom'un INFORMAP veri yapıları örnek olarak verilebilir.

c. Kenar-Düğüm Veri Yapısı

Kaynaklarda kenar-düğüm veya bağ-düğüm (edge-node, link-node, arc-node) gibi değişik isimler alan bu veri yapısı sphagetti veri yapısında karşılaşılan sorunları gidermek üzere tasarlanmış daha iyi bir yapıdır.

Düğüm bir noktadır ve bir koordinat çifti ile ifade edilir. Buna göre bir nokta detay, bir çizgi detayın baş ve son noktaları, kesişim (kavşak) noktaları, temas noktaları birer düğüm olarak tanımlanır.

Kenar ise iki düğüm arasında uzanan nokta (koordinat çifti) katarıdır. Baş ve son noktaları kenara dahil değildir. Bir kenar mutlaka iki düğüm arasında uzanır. Bu tanımlamalara göre kenar-düğüm yapısında :

Nokta detay = düğüm

Çizgi detay = zincir = düğüm + kenar + düğüm + kenar
..... + düğüm + kenar + düğüm

alan detay = zincir

Veya

alan detay = dış zincir + iç zincir

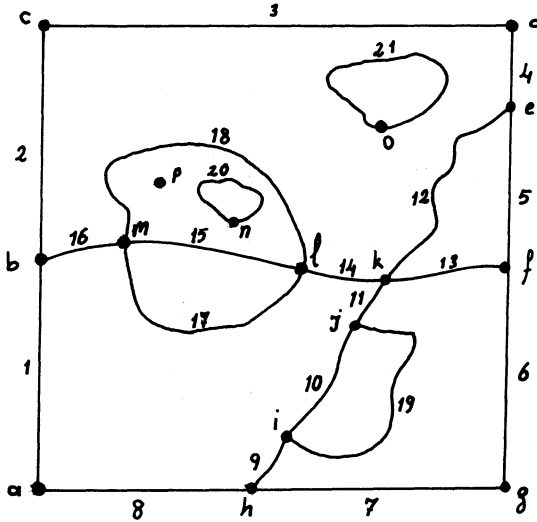
olarak gösterilir.

Kenar-düğüm yapılı bir grafik kütük elde etmek üzere başlıca iki yol izlenir. İlkinde sayısallaştırma sphagetti yapıda yapılır ve elde edilen kütük sonradan batch (yığın) bir programla kenar-düğüm yapıya çevrilir. İkinci yolda ise daha sayısallaştırma sırasında program düğümleri belirleyerek kenar-düğüm yapılı kütüğü doğrudan oluşturur. Her iki yolda da detayların belirlenmesi sonradan etkileşimli editleme ve etiketleme ile yapılır.

Kenar-düğüm veri yapılı bir kütükte üç ayrı tip kayıt vardır. Düğüm kayıtları sabit uzunluklu, kenar kayıtları değişken uzunlukludur. Ayrıca bir de de-

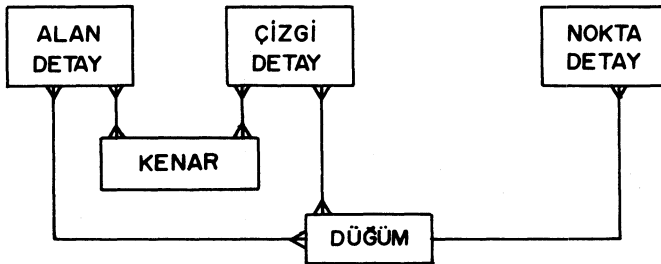
tay kayıtlarına gerek vardır ve değişken uzunluklu bu kayıtlarda detayın hangi düğüm ve kenarlardan oluştuğu kaydedilir.

Örnek haritamızın kenar-düğüm veri yapısındaki görünümü (Şekil-31) dedir.



Şekil-31 : Kenar-Düğüm Veri Yapısı

Bu haritayı bilgisayarda temsil eden grafik kütük (Şekil-32) deki gibi olacaktır. Bu veri yapısındaki varlıklar ve ilişkileri gösteren mantıksal veri yapısı şeması ile (Şekil-33) dedir.

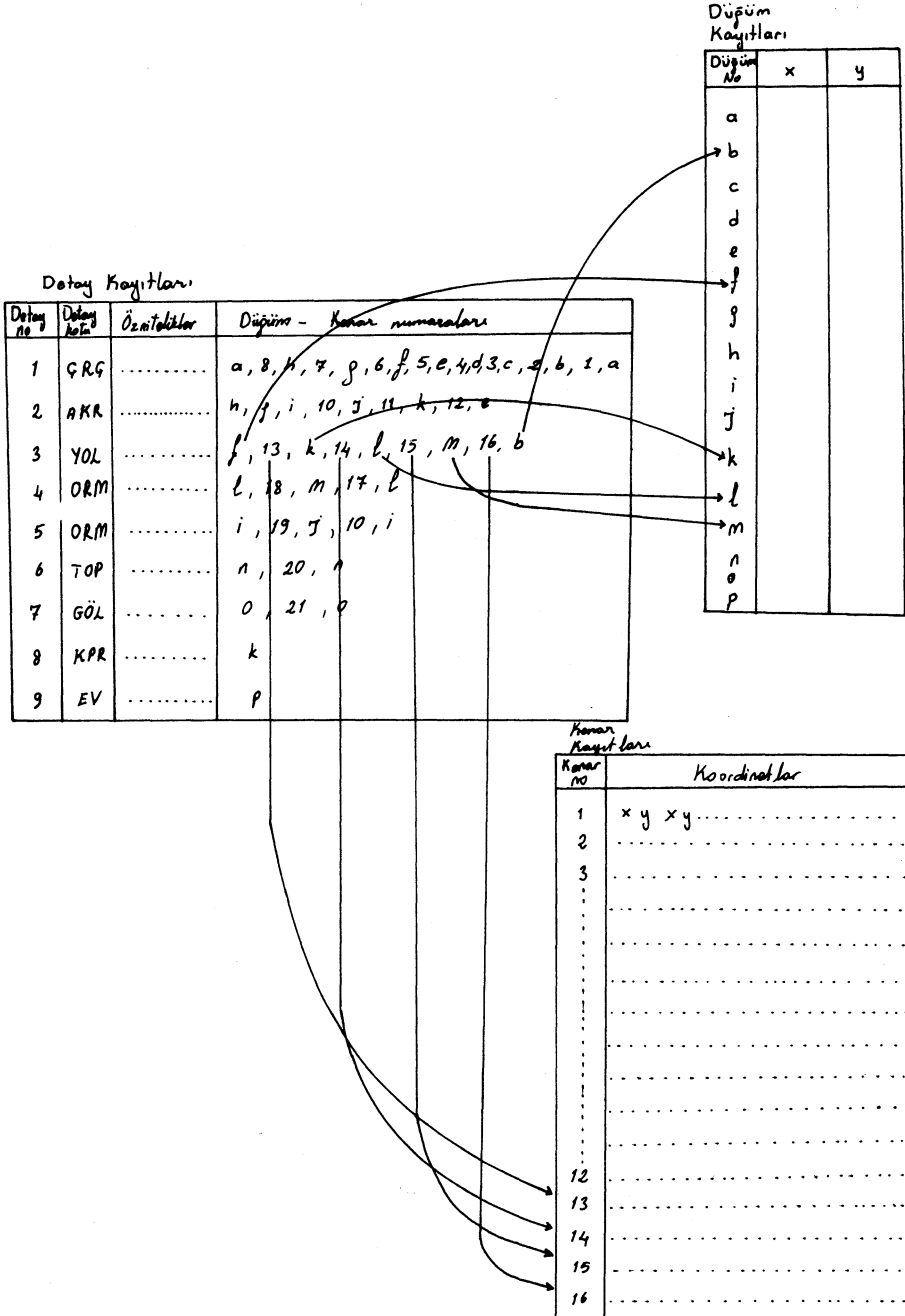


Şekil-33 : Kenar-Düğüm Mantıksal Veri Yapısı
Şeması

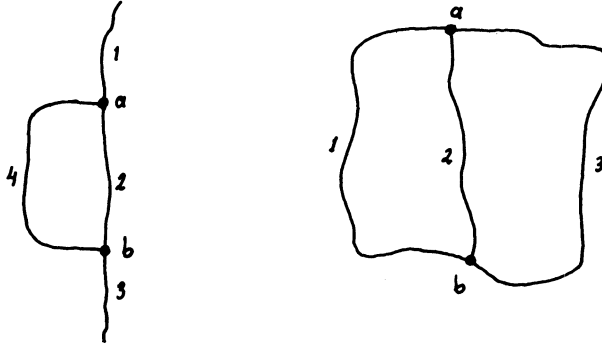
Çözümler :

Kenar-düğüm veri yapısı sphagetti veri yapısına göre özellikle geometrik sorunlar için birçok çözüm getirmektedir.

* Çift sayısallaştırma yoktur. Dolayısıyla binmeler ve boşluklar ortadan kalkmıştır (Şekil-34).

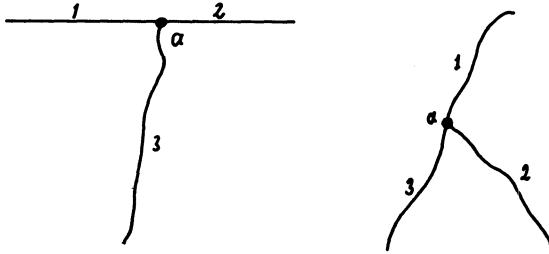


Şekil-32 : Kenar-düğüm veri yapılı kütük



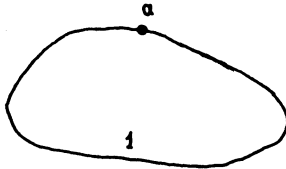
Şekil-34 : Kenar-Düğüm Veri Yapısında Alan/Çizgi ve Alan/Alan Komşulukları

* Temas noktaları ve kesişim noktalarındaki kopukluk ve taşmalar ortadan kalkmıştır (Şekil-35).



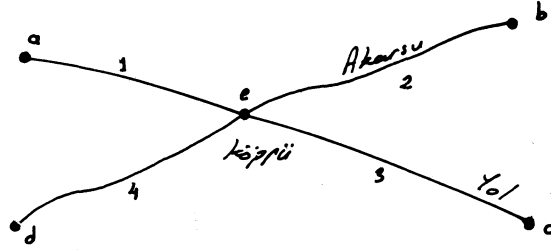
Şekil-35 : Kenar Düğüm Veri Yapısında Temas ve Kavşak Noktaları

* Alan detaylar sorunsuz kapanmaktadır (Şekil-36).



Şekil-36 : Kenar Düğüm Veri Yapısında Alan Detaylar

* Çizgi üstünde veya çizgilerin kesişim noktasında olması gerekli noktalar tam yerlerindedir (Şekil-37).



Şekil-37 : Kenar-Düğüm Veri Yapısında Kesişim Noktasındaki
Nokta Detaylar

SORUNLAR :

Bütün bu çözümlere rağmen hâlâ bazı sorunlar vardır ve veri yapısı hâlâ yeteri kadar "akıllı" değildir.

* Yön kavramı yoktur. (Akarsu hangi yöne akıyor ?)

* Bir kavşak noktasına gelindiğinde bağlantı seçenekleri açık değildir, navigasyon zordur. (Akşehir'den Bayburt'a giden yolların belirlenmesi ve bunlardan en kısa olanının seçimi.)

* Komşuluk ilişkileri açık değildir. Belli yönde ilerlerken sağda ve solda neler var, belli değildir. (İstanbul'dan İzmit'e giderken yolun sağ tarafındaki benzin istasyonları hangileridir ?)

* Alan detayların içindeki diğer detaylar belli değildir. Ancak uzun hesaplamalardan sonda bulunabilir. (Beynam ormanı içindeki binalar ve köprüler hangileridir ?)

Şimdiye kadar sözünü etmediğimiz sorunların, gerçek bir coğrafi bilgi sisteminde örnekleri yukarıda parantez içinde verilen sorulara yanıt vermek üzere çözümü gerekmektedir.

Kenar-düğüm veri yapısı bugün uygulama kurumlarında yapısal veri olarak adlandırılmaktadır. Bu adlandırma göreceli bir sezgiden kaynaklanmaktadır ve kenar-düğüm yanında sphagetti yapısı yapısal olmayan olarak nitelenmektedir. Bu terimlerin doğru oldukları söylenemez.

Kenar-düğüm veri yapısı otomatik düğüm türetme programı ile sayısallaştırma sırasında türetildiğinde daha güçlü işlemciler (processor) gerekmektedir.

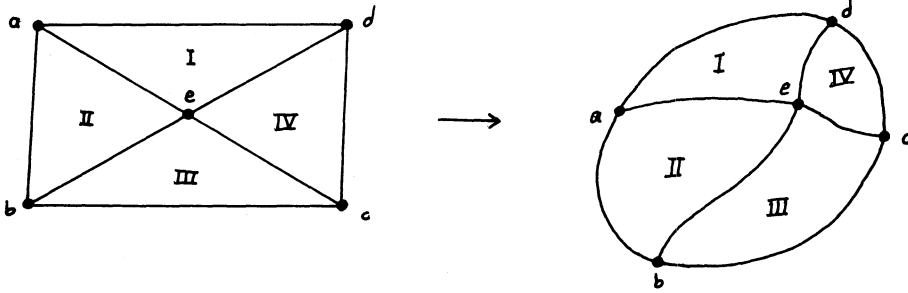
Kenar düğüm veri yapısı günümüzde sağladığı avantajları nedeniyle sayıları giderek artan yazılım sistemlerinde kullanılmaktadır. Bu yapının örnekleri olarak SIEMENS'in SICAD'i, SYSCAN'in GINIS'i gösterebilir.

d. Topolojik Kenar-Düğüm Veri Yapısı

4,c bölümünün sonunda sorunlar olarak sıralanan gereksinimlerin karşılanabilmesi için Topoloji biliminin katkısına gerek vardır. Bu nedenle burada topoloji biliminin konusu ve mantığı çok kısa (amaca yeter ayrıntıda) açıklanacaktır.

(1) Topoloji

Topoloji, geometrik şekillerin şekil bozulmasına rağmen değişmeden kalan özellikleri ile uğraşır. Örneğin Şekil-38,a daki geometrik şekil bozularak Şekil-38,b haline gelmiş olsun.



Şekil-38 : Geometrik Şekillerde Bozulma

Bu bozulma sırasında

- Koordinatlar,
- Uzunluklar,
- Alan değişmiştir.

Ancak

- Herbir noktada birleşen çizgi sayısı
- Noktalar arasındaki bağlantılar,
- Herbir kenarın iki tarafındaki alan detaylar değişmeden kalmıştır.

Bu durumda bu iki şeklin topolojik olarak eşdeğer olduğu söylenir. İşte topoloji şekillerin büyüklük ve biçim özellikleri ile değil, şekil bozulması karşısında değişmeden kalan özellikleri ile ilgilenir.

Gerçi kartografyada şekillerin bozulmasına (bazı transformasyonlar hariç) izin verilmez ama burada sözü edilen özellikler coğrafi verilerin yapılandırılmasında önemli katkılarda bulunabilir. Topolojinin bu katkıları şöylece özetlenebilir:

(a) Daha hızlı veri seçebilmek üzere ilişkilerin (çakışıklık, komşuluk vb.) kolayca tanımlanmasına yardım eder.

(b) Çakışıklık (detay tanımlarında aynı kenar veya aynı düğümün yer alması) bir kez tanımlandığında ortak çizginin bir yerde depolanması suretiyle fazla veriyi en aza indirir.

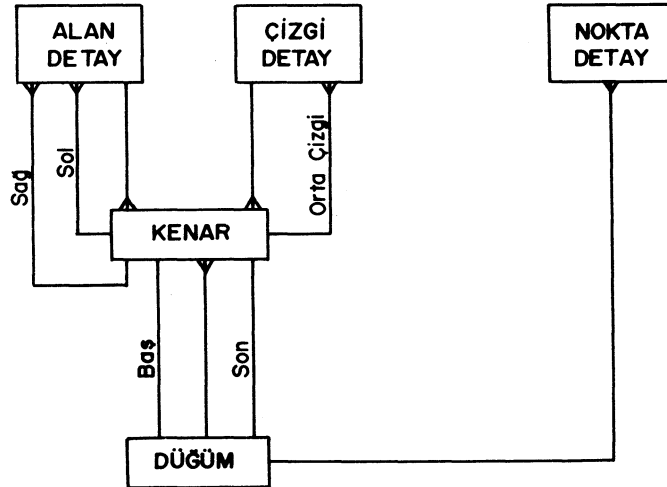
(c) Geometrik veri boyunca navigasyona yardımcı olur.

(d) Geometrik verinin kendi içinde tutarlı kalmasını sağlar.

İşte bu topolojik ilişki kavramları kenar-düğüm veri yapısına eklendiğinde daha ileri düzeyde veri yapıları elde edilmiş olur.

(2) Topolojik Kenar-Düğüm Veri Yapısı Genel Özellikleri

Bu yapıda normal kenar-düğüm veri yapısında olduğu gibi yine düğüm kayıtları (düğüm kütüğü), kenar kayıtları (kenar kütüğü) ve detay kayıtları (detay kütüğü) bulunmaktadır. Ancak kütüklerdeki kayıtlar arasında topolojik özellikleri (çakışıklık, komşuluk, bağlantı, yön vb.) temsil eden ilişkiler kurulmuştur. Bu veri yapısının mantıksal veri yapısı şeması Şekil-39 da görülmektedir. Şekilde de görüldüğü gibi hepsinden önemlisi kenar ve düğüm kayıtları arasında ilişki kurulmuştur. Herbir düğüm kaydında bu düğümün hangi kenarların başı olduğu herbir kenar kaydında da bu kenarın baş ve sonundaki düğümler yazılmıştır. Kenar kayıtlarına ayrıca herbir kenarın sağ ve solunda hangi alan detayların olduğu veya bu kenardan geçen çizgi detayların hangisi olduğu yazılmıştır.



Şekil-39 : Topolojik Kenar-Düğüm Mantıksal Veri Yapısı Şeması

Topolojik kenar düğüm veri yapısında kenar kayıtlarında baş ve son düğümleri kaydedildiğinden yön kavramı getirilmiş olur. Kural olarak sayısallaştırma yönü pozitifdir ve bir ok ile gösterilir.(Şekil-40)



Şekil-40 : Yönlü Kenar

Ayrıca yine kural olarak alan detaylar daima solda kalacak şekilde tanımlanır.

Topolojik kenar-düğüm veri yapısı bugünkü coğrafi bilgi sistemlerinin gereksinimlerini hemen hemen tümüyle karşılamakta ve bu arada uygun performans vermektedir.

Bu veri yapısını gerektiren yazılım sistemlerine örnek olarak ESRI'nin ARC/INFO'su GVC'nin Geo Vision'u ve DMA'nin SLF'i gösterilebilir.

(3) SLF (Standart Linear Format)

SLF, Topolojik kenar-düğüm veri yapısının özel bir örneği olmakla beraber bazı ufak farklılıklar içermektedir. DMA (Defence Mapping Agency) tarafından 1983-84 yıllarında geliştirilen bu veri yapısı mevcut sphagetti yapısından 1990 lı yıllarda kullanılmak üzere tasarlanan MC&G (Mapping Charting and Geodesy) isimli veri yapısına geçişte bir ara format (1984-1990) olarak düşünülmüştür ve halen kullanılmaktadır.

SLF teki en önemli değişiklik düğüm kaydı olmamasıdır. Kenarlar, şimdiye kadar tanımlanan kenarların aksine baş ve son nokta koordinatlarını da içermektedir. Bu tip kenara "kenar" yerine "segment" ismi verilmektedir.

Bu yapı içinde birden fazla segmentte yer alan baş ve son noktaları düğüm anlamına gelmektedir.

SLF teki diğer bir fark ise nokta, çizgi ve alan detayların aynı detay kaydında gösterilmesidir. SLF'in mantıksal veri yapısı Şekil-41 de görülmektedir.

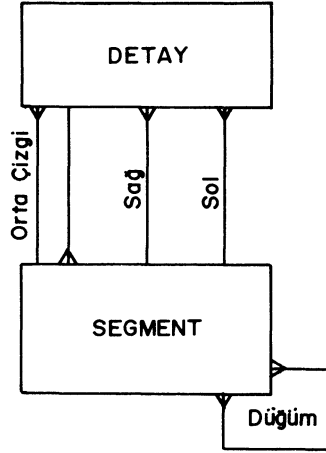
Detay kayıtlarında :

- detay no.,
- detay tipi (P = Nokta/L = Çizgi/A = Alan),
- öznitelikler,
- segment listesi : (yön, segment no.) ;

Segment kayıtlarında ise :

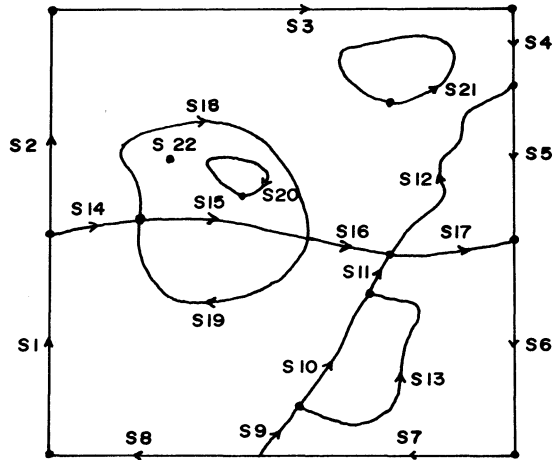
- Segment no.,
- koordinat listesi : (x,y.....x,y),
- detay listesi : (detay no, yön: C = ortada/L = solda/R = sağda)

bilgileri bulunur.



Şekil-41 : SLF Mantıksal Veri Yapısı Şeması

Buna göre örnek haritamızın bu yapıda sayısallaştırılması ile elde edilen segmentler Şekil-42 de görülmektedir.



Şekil-42 : Segmentler

Aynı örnek haritanın SLF yapılı grafik kütüğü ise Şekil-43 de görülmektedir.

Detay kütüğü

Det. no.	Det. tipi.	Öznitelikler	Segment listesi
1	A		-S8, -S7, -S6, -S5, -S4, -S3, -S2, -S1,
2	L		+S9, +S10, +S11, +S12
3	L		+S14, +S15, +S16, +S17,
4	A		-S19, -S18, +S20
5	A		+S13, -S10
6	A		-S20
7	A		+S21
8	P		+S17
9	P		+S22

Segment no.	Koordinat listesi	Detay listesi
1		det1-C
2		det1-C
3		det1-C
.		.
.		.
.		.
.		.
9		det2-C
10		det2-C, det5-R
11		det2-C
12		det2-C
13		det5-L
.		.
.		.
.		.
20		det6-R
21		det7-L
22		det9-C

Şekil-43: SLF kütüğü

Çözümler :

* Sağda ve solda ne var sorusu yön kavramı ile çözülmüştür.

* Aynı segmenti paylaşan detayların belirtilmesi ile komşuluk ilişkisi de çözülmüştür.

* Bağlantılar yardımı ile navigasyon (herhangi bir kavşağa gelindiğinde hangi yola sapılacağına belirlenmesi) kısmen çözülmüştür. Normal topolojik kenar düğüm veri yapısında düğüm kayıtları ayrıca var olduğundan navigasyon daha kolaydır.

* Topolojinin paylaşılması suretiyle dataylar arasındaki konuma bağlı ilişkiler mümkün kılınmıştır.

- aynı düğümde bulunan nokta detaylar,

- aynı kenarı paylaşan çizgi detaylar

- aynı alanı paylaşan alan detaylar (iç zincir:Şekil-41, detay 4, Segment 20) işlem yaparak da olsa tanımlanabilmektedir.

SORUNLAR :

* İki nokta arasında bağlantı olup olmadığı veya iki çizgi detayın kesişip kesişmediği aynı koordinata sahip uç nokta olup olmadığını aramak suretiyle olur. Başka bir deyişle yukarıdaki çözümlerden c ve d ancak işlem yaparak mümkündür.

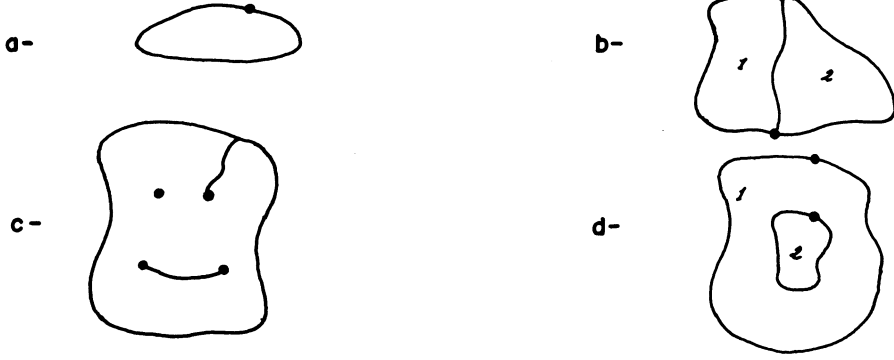
* İçerme (içine alma) sorununun çözümü çok zordur. Bir detayın başka detayları içine aldığını bulmak için bütün detayların bütün noktalarını taramak ve "poligon içinde nokta" hesabı yapmak gerekir.

e. Tam Topolojik Veri Yapısı

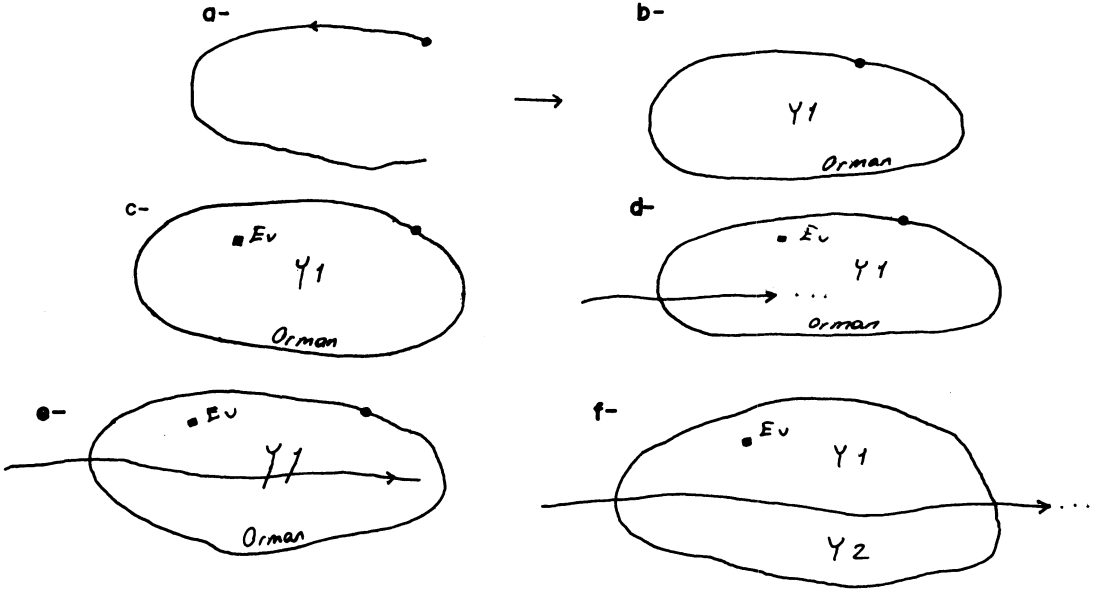
Yukarıda sayılan ve hâlâ daha çözümü zor olan sorunları gidermek üzere tam topolojik veri yapıları geliştirilmektedir. Tam topolojik veri yapısında sorunların çözümüne yardımcı olmak üzere kenar-düğüm yapısına ek olarak yüz (face) kavramı tanımlanmıştır.

Yüz, kenarlar ile sınırlanmış ve bir kenar ile daha fazla bölünmeyen iki boyutlu en büyük alandır. (Şekil-44)

Şekil-44 de a ve c de birer yüz, b ve d de ikişer yüz bulunmaktadır. Yüzlerin belirlenmesinde iki değişik yaklaşım vardır. Birinci yolda kenar-düğüm veri yapısındaki bir kütükte yığın (batch) bir programla kenarlarla bölünmüş bu kapalı poligonlar belirlenir. İkinci yolda ise yüzler daha sayısallaştırma sırasında, sayısallaştırma programı ile belirlenir. Bu işlem "yüz bölünmesi" olarak adlandırılır. Şekil-45 de orman alan içinden geçen yol sayısallaştırılırken yüzün bölünmesi görülmektedir.



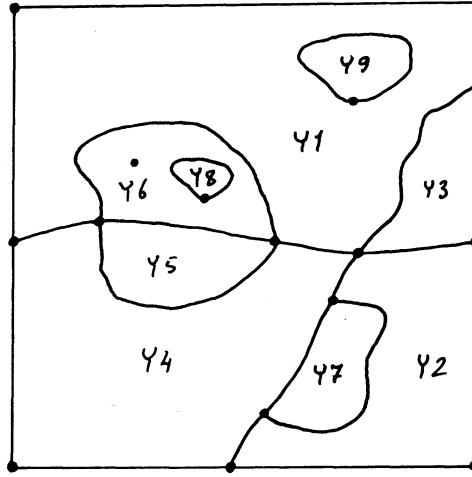
Şekil-44 : Yüzler



Şekil-45 : Yüz Bölünmesi

Bugün için tam topolojik özellikli veri yapısı örnekleri çok azdır. İngiltere'de DMS'te (Directorate of Military Survey) tasarımı yapılan MSDS (Military Survey Data Structure), ABD de DMA'de (Defense Mapping Agency) 1990 lı yıllar için geliştirilen MC&G (Mapping Charting and Ceodesy) isimli veri yapıları bu gruba girmektedir. Intergraph firmasınca bu veri yapısını gerçekleştirmek üzere TIGRIS isimli bir coğrafi veri tabanı yönetim sistemi geliştirilmiş ve 1988 yılı sonunda piyasaya sürülmüştür.

Süregelen örneğimizi MC&G veri yapısına göre sayısallaştırdığımızda Şekil-46 daki sonuç elde edilir.



Şekil-46 : MC&G Veri Yapısı

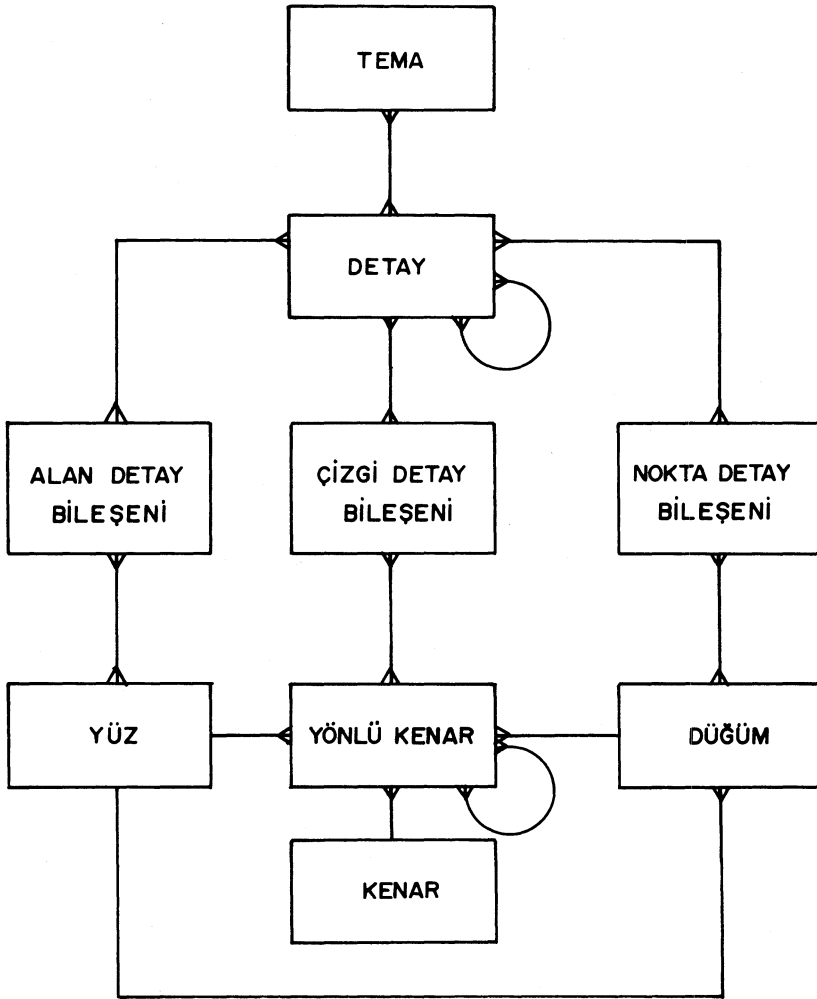
Buna göre bir harita paftasına çizgi ve nokta topluluğu olarak değil yüzlerin birleşimi gözüyle bakılmaktadır. Bu veri yapısına ait mantıksal veri yapısı şeması Şekil-47 de görülmektedir.

Bu yapıda en önemli özellik detay bilgileri (öznitelikleri vb.) ile geometrik eleman bilgilerinin (yüz, kenar, düğüm) birbirinden ayrı düşünülmesidir. Her detay birden fazla alan ve/veya çizgi ve/veya nokta detay bileşeninden oluşmaktadır. Alan detay bileşeni yüz olarak, çizgi detay bileşeni yönlü kenar olarak, nokta detay bileşeni düğüm olarak temsil edilmektedir. Her kenarın iki yönlü kenarı vardır. (+ ve -) Her yüz birden fazla yönlü kenar ve düğüm ile tanımlanır. Detaylar da kendi içinde ilişkili olabilir. Örneğin nehir sistemi detayı birçok akarsu detayından oluşur. Belli detaylar bir konuda grup oluştururlar. Bu gruplara tema denir.

Yapı oldukça ayrıntılı ve pekçok soruya cevap verebilecek şekilde düzenlenmiştir. Her nokta sadece bir kere kayıtlıdır, dolayısıyla bütünlük korunabilmektedir.

Tema-detay-bileşen şeklindeki detay hiyerarşisi zengin sorgulama olanaklarını sunmaktadır ve komşuluk, bağlantı, kesişim ve içerme sorunları çözülmüştür.

TIGRIS'te gerçekleştirilen bu veri yapısının aşağıda sıralanan avantajları sağladığı belirtilmektedir.



Şekil-47 : TIGRIS Mantıksal Veri Yapısı Şeması

- * Orta veri çizgileri açıkça bellidir.
- * Koordinat erişimleri minimumdur.
- * Her nokta ancak ve ancak bir topolojik elemana aittir.
- * Etkileşimli konuma bağlı analiz mümkündür.
- * Topolojinin güncelleştirmesi otomatiktir.
- * Detay hiyerarşisi sağlar.

Bununla beraber TIGRIS bazı dezavantajları da beraberinde getirmektedir.

- * Veri depolama hacmi artmıştır.
- * Yazılım karmaşıktır.

* Uygun performansa ulaşmak üzere çok daha güçlü çalışma istasyonları gerekmektedir. Bu da maliyetleri artırmaktadır.

5. SONUÇ

Coğrafi veri tabanları normal bir veri tabanından, konuma bağlı varlıklara ait grafik ve grafik olmayan bilgileri birlikte içermesi bakımından farklılaşmaktadır. Bir coğrafi veri tabanına dayalı olarak kurulan coğrafi bilgi sistemi ise konuma bağlı varlıklara (coğrafi varlıklara) ait grafik ve grafik olmayan bilgilerin toplanması, bir coğrafi veri tabanı biçiminde bütünleşik olarak organize edilmesi, işlenmesi (dönüşüm ve analiz), sorgulanması ve kullanıcılara çizimler, ekran görüntüleri veya raporlar biçiminde sunulmasını sağlayan yazılım/donanım sistemleridir. Coğrafi bilgi sisteminin kabiliyetleri büyük ölçüde dayandığı coğrafi veri tabanındaki veri yapısına bağlıdır. İleri düzeyde bir coğrafi veri yapısı konuma bağlı sorgulama, dönüşüm ve analiz olanaklarını büyük ölçüde artırmaktadır ve grafik bilgi sunuşunda geometrik sorunları ortadan kaldırmaktadır.

Grafik verilerin bilgisayarda sayısal olarak temsil edilmesi ile klasik haritacılıktaki bir çok kavram değişmiş ve yeni yeni kavramlar eklenmiştir. Yazıda bunlardan bir bölümüne yer verilmiştir.

Raster ve vektör teknikleri çeşitli avantaj ve dezavantajları ile oldukça değişik karakterlerdedir. Bu farklılık grafik veri toplayan ve sunan sayısal-laştırıcı, çizici, grafik ekran vb. donanım çeşitlerinde de kendini hissettirmektedir.

Raster veri yapılarında erişim yolunu kısaltmak ve bellek gereksinimini azaltmak ön planda iken vektör veri yapılarında geometrik kalite ve konuma bağlı sorgulamalara elverişlilik ön planda tutulmaktadır.

Yazıda raster ve vektör veri yapıları çeşitli kategoriler halinde gruplandırılarak açıklanmıştır. Ancak bu veri yapıları yazılım sistemleri ile gerçekleştirilirken ufak bazı farklılıklara uğradıkları bir gerçektir. Topoğrafyayı temsil etmek üzere kullanılan veri yapıları (vektörel, matris, TIN vb.) kapsam dışı bırakılmıştır.

Veri yapısının sunduğu olanaklar arttıkça performans sorunu da büyümekte ve bu durumda güçlü işlemciler gerekli olmaktadır.

Coğrafi veri yapıları üzerinde geliştirme çalışmaları sürmektedir. 1990 11 yıllarda iki boyutlu veri yapılarında tatminkâr bir düzeye ulaşılırken üç boyutlu veri yapılarının yoğun tartışmalara sahne olacağı tahmin edilmektedir.

KAYNAKLAR

- /1/ Armstrong M.A. : Basic Topology. Springer. 1983
- /2/ Blais J.A.R. : Theoretical Considerations for Land Information Systems. University of Calgary.1986
- /3/ Bruegger B., Frank A. : Hierarchies over Topological Data Structures. Proc. ACSM/ASPRS Vol.4 GIS, pp. 137-145, Baltimore. 1989
- /4/ Burrough P.A. : Principles of Geographical Information Systems for Land Resources Assessment. Clarendon Press, Oxford. 1986
- /5/ Claire R.W., Guphill S.C. : Spatial Operators for Selected Data Structures. Proc. Auto Carto 5, pp. 189-200 ASP/ACSM,Virginia. 1982
- /6/ Date C.J. : An Introduction to Data Base Systems Addison-Wesley, Reading Mass. 1986
- /7/ ESRI : ARC/INFO Geographic Information System Software. 1986
- /8/ Greasley I. : Data Structures to Organize Spatial Subdivisions. Proc. ACSM/ASPRS Vol. 5 GIS, pp. 139-148, St.Loie. 1988
- /9/ Hrowitz E.,Sahni S. : Fundamentals of Data Structures Pitman, Maryland. 1980
- /10/ Intergraph : TIGRIS An Object Oriented GIS Environment. 1989
- /11/ Kleiner A.,Brassel K.E. : Hierarchical Grid Structures for Static Geographic Data Bases.Proc. Auto Carto-8, pp. 485-496, London. 1986
- /12/ Mark D.M. : The use of Quadrees in Geographic Information Systems and Spatial Data Handling. Proc. Auto Carto-8, pp.485-496, London. 1986

- | | | |
|------|-------------------------------------|---|
| /13/ | Martin J. | : Computer Data Base Organisation. Prentice-Hall, New Jersey. 1977 |
| /14/ | Monmonier M.S. | : Computer Assisted Cartography Principles and Prospects. Prentice-Hall, N.J.. 1982 |
| /15/ | Sarbanoglu H. | : Coğrafi Bilgi Sistemleri Harita Dergisi, Sayı, 103, 1989 |
| /16/ | Tamminen M. | : Some Aspects of Defining Spatially Referenced Data. Proc. FIG XVI-Congress Comm. 3, pp. LII.1. 1981 |
| /17/ | U.K. Directorate of Military Survey | : Reports on CREST7 Project and MSDB. Feltham. 1987 |
| /18/ | Van Lamsweerde A. | : Spatial Datastructures for Land Information Systems. Proc. FIG XVI Congress Comm. 3., pp. 301.5. 1981 |