T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MODEL GENELLEŞTİRMESİNDE
ALAN ÇİZGİ GEOMETRİK DÖNÜŞÜMÜ

Hüseyin Zahit SELVİ

YÜKSEK LİSANS TEZİ

JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ ANABİLİM DALI

KONYA, 2006

T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MODEL GENELLEŞTİRMESİNDE ALAN ÇİZGİ GEOMETRİK DÖNÜŞÜMÜ

HÜSEYİN ZAHİT SELVİ

YÜKSEK LİSANS TEZİ

JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ ANABİLİM DALI

ÖZET

Yüksek Lisans Tezi

MODEL GENELLEŞTİRMESİNDE ALAN ÇİZGİ GEOMETRİK DÖNÜŞÜMÜ

Hüseyin Zahit SELVİ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Jeodezi ve Fotogrametri Mühendisliği Anabilim Dalı

Danışman: Prof.Dr. Mehmet YERCİ

2006, 66 Sayfa

Jüri: Prof.Dr. Mehmet YERCİ

Prof.Dr. Cevat İNAL

Doç.Dr. İ. Öztuğ BİLDİRİCİ

Genelleştirme, kavram olarak, ayrıntı açısından zengin olan gerçek dünya tanımlamasının içeriğinin azaltılmasını gerektirir. Bu azalma objelerin hem geometrisini hem de özniteliklerini etkiler. Model genelleştirmesi sonucu alan-çizgi, alan-nokta, çizgi-nokta gibi geometri değişimleri ortaya çıkar. Bu tez kapsamında bu dönüşümlerden yaygın kullanım alanına sahip olan alan-çizgi dönüşümü üzerinde durulacaktır. Alan objeler birbirlerinden çok farklı geometrik özelliklere sahip olduklarından, her alan obje için geçerli olan bir alan-çizgi dönüşüm yöntemi henüz üretilememiştir. Bundan dolayı günümüzde çeşitli alan obje tipleri ve ihtiyaçlara uygun farklı modeller geliştirilmiştir. Bazı yöntemler uygulama zamanının kısalığına, bazıları doğruluğa, bazıları da bilgisayar belleğinin optimal kullanılmasına öncelik vermişlerdir. Bu çalışmada alan-çizgi dönüşüm yöntemlerinden beş tanesi ve çift çizgi tek çizgi dönüşüm yöntemi ayrıntılı olarak incelenecektir. Uygulama bölümünde ise üçgenleme yöntemi esasına dayalı geliştirilen alan çizgi dönüşüm yazılımı detaylıca anlatılacaktır.

Anahtar Kelimeler: model genelleştirmesi, geometrik dönüşüm, topoloji, coğrafi bilgi sistemi

ABSTRACT

MSc Thesis

AREA-LINE GEOMETRY CHANGES IN MODEL GENERALIZATION

Hüseyin Zahit SELVİ

Selcuk University

Graduate School of Natural and Applied Sciences

Department of Geodesy and Photogrammetry Engineering

Supervisor: Prof.Dr. Mehmet YERCİ

2006, 66 Page

Jüri: Prof.Dr. Mehmet YERCİ

Prof.Dr. Cevat İNAL

Assoc.Prof.Dr. İ.Öztuğ BİLDİRİCİ

Reducing the real world reality to map scale makes generalization necessary. This reduction influences geometry and attributes of objects. In model generalization,area-line, area-point and line-point changes occur. In this study, area-line changes is handled. Since area objects have different geometric properties, there is no universal solution that is applicable to all kinds of area objects. Therefore there are several methods applicable to certain objects under certain circumstances. Priorities of methods are accuracy, processing time and optimal use of computer memory. We discuss five area – line geometry change methods and double line – single line change method in detail. Finally an algorithm that is developed by using triangulation method is explained in detail.

Keywords: model generalization, geometric changing, topology, geographic information systems.

ÖNSÖZ

Çalışmalarım boyunca bana desteğini eksik etmeyen danışmanım Prof.Dr. Mehmet YERCİ'ye, tez çalışmam süresince bana her türlü desteği veren saygıdeğer hocam Doç. Dr. İ. Öztuğ BİLDİRİCİ'ye, yazılım aşamasında fikirleriyle destek olan Sayın Serdar AK'a, çalışmam sırasında bana tahammül eden oda arkadaşım Arş.Gör. R. Alpay ABBAK'a teşekkürü bir borç bilirim.

Hüseyin Zahit SELVİ

İÇİNDEKİLER

ÖZET i
ABSTRACT ii
ÖNSÖZ iii
İÇİNDEKİLER iv
ŞEKİL LİSTESİ vi
ÇİZELGE LİSTESİ viii
1. GİRİŞ 1
2. MODEL GENELLEŞTİRMESİ VE GEOMETRİ DEĞİŞİMLERİ 3
2.1. Model Genelleştirmesinde Geometri Değişimi 4
3. CBS AÇISINDAN TOPOLOJİ VE ALAN ÇİZGİ DÖNÜŞÜMÜ KAVRAMI 6
3.1. Cbs Açısından Topoloji Kavramı 6
3.1.1. Çizgi – düğüm (arc - node) topolojik veri yapısı 7
3.2. Cbs Açısından Alan – Çizgi Dönüşümü Kavramı 8
4. ALAN – ÇİZGİ DÖNÜŞÜM YÖNTEMLERİ 10
4.1. Üçgenleme Yöntemi 10
4.2. Su Çizgileri Yöntemi 15
4.3. En Büyük Daire Yöntemi 20
4.3.1. Vektör formatta en büyük daire yönteminin uygulanması 21
4.3.2. Raster formatta en büyük daire yönteminin uygulanması 22
4.4. İnceltme (Thinning) Yöntemi 23
4.4.1. RLE (Run Length Encoding) yöntemi 25
4.4.2. Alanların bölümlenmesi 27
4.4.3. Alanların iskeletinin çıkarılması 27
4.4.4. İskeletleşmiş elemanların vektörizasyonu ve topoloji elde edilmesi 31
4.5. Basit İskeletleştirme Ve Çatı Yöntemi 33
4.5.1. Basit iskeletleştirme yöntemi 33
4.5.2. Çatı yöntemi 38
5. ÇİFT ÇİZGİ – TEK ÇİZGİ DÖNÜŞÜM YÖNTEMLERİ 42
5.1. Yapısallaştırılmış Vektör Formatında Eksen Belirleme 42
6. UYGULAMA:ÜÇGENLEME YÖNTEMİYLE EKSEN BELİRLEME 44

6.1. Türkiye'deki Mevcut Veri Yapısı 44
6.2. NETCAD ve NCMACRO 45
6.2.1. Netcad 45
6.2.2. Ncmacro 46
6.3. Uygulama 49
6.4. Sonuç 57
7. SONUÇ VE ÖNERİLER 59
KAYNAKLAR 62

ŞEKİL LİSTESİ

Şekil 4-1 : Voronoi diyagramı ve Delaunay üçgenlemesi 10
Şekil 4-2 : Delaunay üçgenlemesinde daire kriteri 11
Şekil 4-3 : Delaunay üçgenlemesinde kenara nokta eklenmesi 13
Şekil 4-4 : Üçgenleme yönteminde orta eksen yaklaşımları 14
Şekil 4-5 : Su çizgileri yöntemiyle eksen oluşturulması 17
Şekil 4-6 : Su çizgileri yöntemiyle oluşturulan bir yol ağı ve kesişim bölgelerinde oluşan yaylar 20
Şekil 4-7: En büyük dairenin özellikleri 21
Şekil 4-8: En büyük daire yöntemiyle oluşturulan eksen örnekleri 22
Şekil 4-9 : Piksellerin 4 veya 8 komşulu olarak etiketlenmesi 26
Şekil 4-10: Bozuk şekillerin tamponla çevrilmesi 26
Şekil 4-11: Piksel ve komşuları (N(P)) 28
Şekil 4-12: Pikselin beyaz köşe komşuya sahip olma durumu 30
Şekil 4-13: Paralel inceltme algoritmasında silinme şartlarını sağlayan iki piksel örneği 30
Şekil 4-14: Sıralı inceltme algoritmasında silinme şartlarını sağlayan piksel örneği 31
Şekil 4-15: İnceltme yönteminin aşamaları a) Orijinal görüntü b)Bölgelerin etiketlenmesi (her farklı bölge ayrı gri tonda c) Görüntünün ters çevrilmesi (bölgelerin arka alana alınması) d) İskeletin elde edilmesi e) Kesişim bölgelerinin genişletilmesi f) Son iskelet ağı 32
Şekil 4-16 : Kenar olayı (A) ve Ayrılma olayı (B) 34
Şekil 4-17 : Poligonal büzülme ve basit iskelet yapısı 34
Şekil 4-18 : Üçgen yuva oluşma durumu 36
Şekil 4-19 : Konveks poligonlar için algoritmanın akış aşamaları 37
Şekil 4-20 : Ayrılma noktası B'nin bulunması 38
Şekil 4-21 : Açıortay grafik ve çatı modeli 40
Şekil 5-1 : Eksenin orta paralel olma durumu 42
Şekil 5-2 : Eksenin açıortay olma durumu 43
Şekil 5-3 : Boşlukların ara parça elemanlarla (2,4) doldurulması 43
Şekil 6-1 : Türkiye'de büyük ölçekli haritalarda veri yapısı 45

Şekil 6-2 : Nokta sıklaştırması için kullanıcıdan gerekli değerlerin alınması 49
Şekil 6-3 : Adalar üzerinde nokta sıklaştırması yapılması..... 50
Şekil 6-4 : Nokta sıklaştırması yapıldıktan sonra üçgenlerin oluşturulması..... 51
Şekil 6-5 : Eksen oluşturmada kullanılacak yol üçgenleri..... 52
Şekil 6-6 : Kavşak bölgelerindeki üçgenlerin farklı tabakaya alınması..... 53
Şekil 6-7 : Eksen noktalarının oluşturulması 53
Şekil 6-8 : Eksen başlangıç ve bitiş noktalarının belirlenmesi 54
Şekil 6-9 : Kavşak noktalarının belirlenmesi..... 55
Şekil 6-10 : Eksenlerin oluşması..... 56
Şekil 6-11 : Yazılım başlangıç ekranı 57
Şekil 6-12 : Kavşaklar dikkate alınmadan eksenlerin oluşturulması 58

ÇİZELGE LİSTESİ

Tablo 2-1 : Küçültücü geometri tipi değişimleri (Uçar ve ark. 2003) 5

1. GİRİŞ

Yapım yöntemlerine göre haritalar “temel harita” ve “türetme harita” olarak iki grupta ele alınır. Temel harita, orijinal arazi ölçmelerinden ya da fotogrametrik değerlendirme yoluyla elde edilir. Türetme harita, daha büyük ölçekli haritalardan ve bölgeye ait başka bilgi kaynaklarından yararlanılarak üretilen haritadır. Türetme haritaya temel altlık oluşturan harita ise “kaynak harita” olarak adlandırılır. Kaynak haritalar olarak temel haritalar kullanılabileceği gibi, daha büyük ölçekli başka kaynak haritalardan elde edilmiş türetme haritalar da kullanılabilir.

Orijinal (yersel ya da fotogrametrik yöntemlerle) ölçmeler temel alınarak harita üretimi, büyük ölçeklerde (1: 1000, 1: 5 000) ya da Türkiye’de uygulanmış olduğu gibi orta ölçeklerin en alt kesiminde (1: 25 000) mümkün olabilir. Harita kullanımının söz konusu olduğu sayısız alanda çok değişik ölçekte ve içerikte haritaya gereksinim duyulduğu açıktır. Bu haritaların üretimi için kartografya, tümü “kartografik genelleştirme” başlığı altında toplanan kendine özgü yöntemler geliştirmiştir. Kartografik genelleştirme günümüzde de kartografyanın en önemli araştırma konularından biridir.

Klasik anlamda kartografik genelleştirme, türetme haritaların elde edilmesi sırasında ortaya çıkan bilgi karmaşıklığının azaltılması, önemsiz bilgilerin atılması, harita objeleri arasındaki belirgin mantıksal ilişkilerin ve estetik kalitenin korunması işlemlerinin bileşkesi olarak tanımlanabilir. Genelleştirmede amaç grafik okunaklılığı çok iyi olan haritalar üreterek haritanın görünümünün ve aktarılmak istenen bilginin kolayca anlaşılmasını sağlamaktır (Bildirici ve Uçar 2003).

Bilgisayar teknolojisinde yaşanan büyük çaplı gelişmeler, birçok alanda olduğu gibi kartografik genelleştirme alanında da etkisini göstermiştir. Yıllarca genelleştirme kavramı, kartografların tecrübe ve becerileriyle sınırlanmış ve kesin kurallara bağlanamamıştır. Bilgisayar teknolojisinin gelişmesi sayısal harita kavramını gündeme getirmiş ve haritaların bilgisayar ortamında yapılması yaygınlaşmıştır.

Bunun sonucu olarak genelleştirme kavramını da subjektif olmaktan çıkarmak ve bilgisayar ortamında modellemek için çalışmalar hızlanmıştır.

Kartografik genelleştirmenin modellenmesi, kartografik objelerin çok çeşitli olması ve birbirlerinden çok farklı yapıya sahip olmaları nedeniyle oldukça karmaşık bir iştir. Kartografik objelerin modellenmesindeki zorluklar ve ihtiyaçların çeşitliliği sayısal ortamda her ihtiyaca cevap verebilen kartografik genelleştirme yazılımlarının üretilmesini bugüne kadar engellemiştir. Bunun yerine değişik ihtiyaçlara öncelik veren, belirli obje tiplerinde iyi sonuçlar veren çok sayıda yazılım geliştirilmiştir.Bu nedenle kartografik objeler alan, çizgi ve nokta obje olarak sınıflandırılarak, her bir obje türü için genelleştirme modelleri üretilmeye çalışılmıştır. Fakat kartografik genelleştirmenin doğası gereği ölçeğe bağlı olması ve günümüzde bir ihtiyaç haline gelen bazı coğrafi bilgi sistemi (CBS) analizlerinin yapılabilmesi, kartografik objeler arasında geometrik dönüşümler yapılmasını zorunlu hale getirmiştir.

Bu çalışma kapsamında model genelleştirmesi içerisinde alan – çizgi geometrik dönüşümü üzerinde durulacaktır. İkinci bölümde model genelleştirmesi kavramı üzerinde durulacak ve bu kavram içerisinde geometri değişimlerinin yeri, geometrik dönüşüm türleri anlatılacaktır. Üçüncü bölümde CBS açısından alan – çizgi geometrik dönüşümü incelenecek ve bu dönüşümle yapılabilecek CBS analizleri hakkında bilgi verilecektir. Dördüncü bölümde ise alan – çizgi dönüşüm yöntemlerinden üçgenleme, su çizgileri, en büyük daire, inceltme ve çatı yöntemleri incelenecek, bu yöntemlerin üstün ve zayıf yönleri üzerinde durulacak, yöntemlerin birbirleriyle karşılaştırması yapılacaktır. Beşinci bölümde çift çizgi tek çizgi dönüşüm yöntemi ve bu yöntemin kullanılabilirliği üzerinde durulacaktır. Altıncı bölüm tezin uygulama kısmı olup, geliştirilen üçgenleme yöntemiyle eksen oluşturma algoritması bu bölümde ayrıntılı olarak sunulacaktır. Son bölümde ise çalışmadan çıkarılan sonuçlar ve ilerisi için yapılan öneriler yer almaktadır.

2. MODEL GENELLEŞTİRMESİ VE GEOMETRİ DEĞİŞİMLERİ

Gerçek dünyanın birebir modellenmesi mümkün olmadığından ilk olarak mekansal verilerin yapılandırılması ve genelleştirilmesiyle birincil model olarak nitelendirilebilecek sayısal mekan modeli (SMM) elde edilir. Birincil model, görselleştirme göz önüne alınmadan amaca uygun çözünürlükte topografik arazi objelerini kapsar. Birincil modellerde ölçek değil çözünürlük kavramı kullanılmaktadır. Model genelleştirmesi, orijinal araziden elde edilen modellerde uygulandığı gibi, bir SMM'den daha düşük çözünürlükte bir başka SMM elde edilmesinde de uygulanır. Model genelleştirmesinde görselleştirilerek kullanıcıya sunulacak ikincil modellere (kartografik model) uygun yoğunlukta ve yapıda verilerin hazırlanması temel hedeftir (Uçar ve ark. 2003).

Model genelleştirmesi yüksek çözünürlüklü SMM verilerinin düşük çözünürlüklü SMM verilerine dönüştürülmesi için gerekli işlev ve araçları sunmaktadır. Bu bağlamda SMM verileri geometrik, semantik ve modelleme tekniği açısından değişime uğratılır. Bu nedenle, geometri, semantik modelleme ve veri modellemesi model genelleştirmesinin değişkenleri olarak düşünülebilir.

Semantik kavramı içerik olarak tanımlamayı ifade etmektedir. Semantik (anlamsal) değişkeninin değişimi model genelleştirmesinde obje sınıflarında, objelere ya da objelere ait özniteliklerde uygulanan sınıflandırma ve seçme (elemine etme) temel işlemlerine neden olur (Uçar ve ark. 2003).

Geometri kavramı, geometrik tanımlamayı ifade eder. Model genelleştirmesinde geometri değişkeninin değişimi geometrik yapının basitleştirilmesini gerektirir. Geometrik yapının değişmesi iki şekilde olur:

- Geometri değişimi (alandan çizgiye, alandan noktaya vb.),
- Konum doğruluğunun azalmasına da neden olan noktaların azaltılması, geometrik yapının basitleştirilmesi işlemleri (örneğin çizgi basitleştirilmesi).

Model genelleştirmesi değişkeni olarak veri modelleme kavramı, semantik ve geometrik değişkenlerin kapsamında olmayan modelleme tekniği ile ilgili tüm değişimleri ifade eder.

Bu tez kapsamında geometrik değişimler inceleneceğinden bu aşamadan sonra model genelleştirmesinde geometri değişimleri üzerinde durulacaktır.

2.1. Model Genelleştirmesinde Geometri Değişimi

Model genelleştirmesinde ayrıntı açısından zengin olan gerçek dünya tanımlamalarının azaltılması, objelerin özniteliklerini etkilediği gibi geometrik tanımlamalarını da etkiler. Geometrik genelleştirme, yüksek çözünürlüklü SMM geometrisini düşük çözünürlüklü SMM gereksinimlerine göre azaltır. Bu bağlamda genelleştirmenin amacı model çözünürlüğüne uygun olarak nokta yoğunluğunun ve geometrik doğruluğun azaltılması ve obje yapısının basitleştirilmesidir. Bir objeye ait geometrik bilgi, geometri tipi ve geometriyi tanımlayan noktalar dizisi olmak üzere iki bileşenden oluşur. Geometri tipi alansal, çizgisel ve noktasal olmak üzere üç şekilde modellenir (Uçar ve ark. 2003).

Her obje için değişik çözünürlük düzeylerinde değişik geometrik modellemeler (alansal, noktasal ve çizgisel) öngörülmektedir. Yüksek ve düşük çözünürlüklü SMM'lerde obje sınıflarının geometrik modellemeleri uyuşuyorsa, geometri yüksek çözünürlüklü modelden düşük çözünürlüklü modele kopyalanarak aktarılabilir. Aksi halde düşük çözünürlüklü modelde geometrinin yeniden oluşturulması gerekecektir. Bu problem geometri tipi değişimi olarak adlandırılır. Geometri tipi değişimleri büyütücü değişimler ve küçültücü değişimler olmak üzere iki gruba ayrılır (Uçar ve ark. 2003).

Büyütücü değişimlerde düşük çözünürlüklü SMM'de geometri boyutu yüksek çözünürlüklü SMM'den fazladır. Bu tür dönüşümlerin gerçekleşmesi için ya ek bilgilere ya da keyfi olarak belirlenen değerlere ihtiyaç vardır. Keyfi kabuller kurallara dayalı genelleştirmenin doğasına aykırıdır. Ek bilgiler ise genel olarak

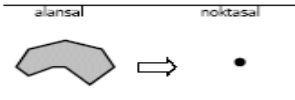
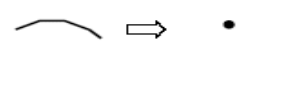
yüksek çözünürlüklü SMM'den elde edilemez. Bu nedenlerle büyütücü değişimlerden model genelleştirmesinde kaçınılmalıdır.

Küçültücü değişimlerde ise yüksek çözünürlüklü SMM'de geometri boyutu düşük çözünürlükteki SMM'deki geometri boyutundan fazladır. Düşük çözünürlükte geometri boyutunun azalması geometrinin basitleşmesine yol açar ki bu da model genelleştirmesinde istenen bir durumdur.

Geometri değişiminin gerekli olup olmadığı yüksek ve düşük çözünürlükteki SMM'lerin obje sınıfları için öngörülen geometri tipleri incelenerek belirlenir. Yeni geometrinin belirlenmesi ise öngörülen modellemeye ve objelerin aralarındaki ilişkilere bağlıdır. Geometri değişimine uğrayacak obje hiçbir obje ile ilişkili değilse geometri tipi değişimi bağımsız olarak düşünülebilir. Obje başka objelerle ilişkili ise bu ilişkilerin, yeni geometrinin belirlenmesinde dikkate alınması gerekir. Geometri tipi değişimi probleminin üçüncü bileşeni komşu objelerin değişen geometrik duruma uygun hale getirilmesidir. Burada topolojik olarak bir bozulma olup olmadığı kontrol edilmeli ve topolojik ilişkiler değişmişse eski haline getirilmelidir.

Schürer (2002) tarafından yapılan bir kaynak araştırması sonucu belirlenen küçültücü geometri değişimlerinde vektör ve raster formatlarda uygulanabilecek algoritmalar Tablo 2-1 'de görülmektedir.

Tablo 2-1 : Küçültücü geometri tipi değişimleri (Uçar ve ark. 2003)

Küçültücü geometri değişimi	Algoritmalar
	raster veriler - MONTANARI - Orta çizgi dönüşümü (OGNIEWICZ) - Voronoi diyagramları vektör veriler - MENKE (yol ağı) - tüm noktalardan orta çizgi hesaplama - su çizgileri yöntemi (CHRISTENSEN)
	dolaylı : a-lan>çizgi>nokta direkt: tüm koordinatların ortalamasının alınması
	1. geometriye ait bir noktanın seçilmesi 2. Nokta belirleme a. çizginin ortası b. ekstrem değerlerle c. tüm koordinatlarla d. alan ağırlığı ile

3. CBS AÇISINDAN TOPOLOJİ VE ALAN ÇİZGİ DÖNÜŞÜMÜ KAVRAMI

3.1.CBS Açısından Topoloji Kavramı

Coğrafi bilgi sistemlerinde nokta, alan, ve ağ topolojisi olmak üzere üç tür topolojiden söz edilir. Topoloji, kelime anlamından da anlaşılacağı üzere coğrafi objelerin birbirleriyle mekansal komşuluk ilişkilerini tanımlar. Temel topoloji bileşenleri, merkez, düğüm noktası ve bağlantı olmak üzere üç tanedir. Nokta topolojisi düğüm noktalarından, ağ topolojisi düğüm noktaları ve bağlantılardan, alan topolojisi ise düğüm noktaları, merkezler ve bağlantı çizgisi objelerinden oluşur (Bildirici 2003).

Bağlantı çizgileri bir düğüm noktasından başlar bir başka düğüm noktasında sona ererler. Merkez noktaları ise kapalı alan içerisinde herhangi bir yerde bulunurlar. Bunların şeklin ağırlık merkezi gibi özel bir noktada olmaları gerekmez. Tek koşul merkez noktaların kapalı şeklin içinde yer almasıdır.

Temel topoloji bileşenlerinden düğüm noktalarına ya tek bağlantı objesi ya da ikiden çok bağlantı objesi bağlanmalıdır. Bir düğüm noktasından yalnızca iki bağlantı elemanı çıkıyorsa, o düğüm noktası topolojik olarak anlamlı değildir. Bu tür düğüm noktaları “pseudonode” olarak adlandırılır. Bu noktalar özel bir anlamı yoksa iki bağlantı çizgisi birleştirilip silinmelidir. Bu noktaların ağ topolojisinde özel anlamları olabilir. Bir yol ağı ele alınırsa, en kısa yol analizi gibi analizler düğüm noktalarına dayalı olarak yapılırlar. Bu durumda en kısa yol araştırması yapılan düğümlerden biri, yol üzerinde bulunan hastane veya polis karakolu olabilir. Bu durumda bu düğüm “pseudonode” olsa dahi silinmemelidir (Bildirici 2003).

Topoloji mantığı, coğrafi bilgi sistemleri için geliştirilen yazılımların önemli bir fonksiyonudur. Bir CBS yazılımında topolojiye olan gereksinimin en büyük sebebi; topoloji sayesinde bir ağ kapsamındaki bağlantıları, yönleri, düğüm noktalarına göre modelleme, benzer özellikteki poligonların komşuluk ilişkilerini, coğrafi özelliklerini

karşılaştırma gibi konum analizlerinin koordinat bilgisine ihtiyaç duymadan yapılabilmesidir (Yomralıoğlu 2000, s.75).

Bir coğrafi bilgi sisteminde, temel topolojik fonksiyonları yerine getirmek için de üç temel topolojik veri yapısı mevcuttur. Bunlar çizgi – düğüm (arc –node) topolojik veri yapısı, poligon – çizgi (polygon – arc) topolojik veri yapısı, sol – sağ (left – right) topolojik veri yapısıdır. Bu tez kapsamında daha çok çizgi – düğüm topolojik veri yapısıyla ilgilenileceğinden diğer yapılar üzerinde durulmayacaktır. Diğer yapılarla ilgili ayrıntılı bilgiler Yomralıoğlu'da (2000) bulunabilir.

3.1.1. Çizgi – düğüm (arc - node) topolojik veri yapısı

Bağlantı yapısını (connectivity) bilgisayarda saklayan topolojik veri yapısıdır. Bağlantı, herhangi bir güzergah üzerindeki hatların birbirini nasıl takip ettiği ve bu hatların bağlantı noktaları hakkında bilgi verir. Örneğin cadde ve sokak bağlantıları gibi. Bağlantı yapısı, çizgi – düğüm topolojik veri yapısıyla ifade edilir. Her bir çizgi iki düğüm noktasından oluşur. Çizginin başlangıç ve bitiş durumuna göre, düğümden düğüme şeklinde çizginin bir anlamda akış yönünü de gösteren liste oluşturulur. Çizgiler kesinlikle bir düğüm noktasında birleşirler. Bu düğüm noktaları farklı çizgiler arasında ilişki kurulmasını sağlar (Yomralıoğlu 2000, s.80).

Şebeke yapısına sahip, çizgi tabanlı coğrafi objelerin bağlantı şekillerinden, karar vermeye yönelik sonuç çıkarmaya yarayan konum analizlerine ağ analizleri denir (Yomralıoğlu 2000, s.225). Ancak düğüm – çizgi topolojisi oluşturulmuş coğrafi veriler ağ analizlerinin gerçekleşmesine olanak sağlar. Autodesk'de (1999, s.413) yaygın olarak kullanılan ağ analizleri hakkında aşağıdaki bilgiler verilmiştir.

- **En kısa yol analizi:** Verilen iki nokta arasındaki en uygun güzergahı bulma analizidir.
- **Overlay (örtme) analizi:** Var olan iki topolojiden yararlanılarak oluşturulan üçüncü topoloji yardımıyla yapılan analizler. Örneğin bir bölgedeki kirli akarsular ve kirlilik kaynaklarından yararlanarak yerleşim açısından uygun olmayan yerlerin belirlenmesi gibi.

- **Buffer analizi:** Var olan bir topolojik yapıya belirli bir uzaklıktaki objelerin incelenmesi. Örneğin kamulaştırma analizleri gibi.

Bir sonraki konuda bu analizler için alan – çizgi dönüşümünün gerekliliği üzerinde durulacaktır.

3.2.CBS Açısından Alan – Çizgi Dönüşümü Kavramı

Yol, akarsu vb. çizgisel kartografik objeler, gözün ayırt etme hassasiyeti dikkate alındığında en çok 1:5000 ölçeğinde orijinal büyüklüklerinde ve çift çizgi olarak sınır çizgileriyle gösterilebilir. Bu ölçekte 1 mm. 5 m.'ye karşılık gelmektedir ve çift çizgili gösterim dikkate alındığında 5m'den daha az genişliğe sahip objelerin bu ölçekte gösterilmesi mümkün değildir. Kısacası 1:5000'den daha küçük ölçeklerde yol, akarsu vb. objeler eksenleri ile gösterilmek zorundadır. Özellikle büyük boyutlu CBS analizlerinin küçük ölçekli haritalarda yapılma zorunluluğu, alan - çizgi dönüşümünü zorunlu kılmaktadır.

Yukarıda bahsedilen geometrik zorunluluğun yanı sıra 3.1. bölümünde anlatılan ağ analizlerinin yapılabilmesi için de kullanılan coğrafi veriler mutlaka çizgi tabanlı vektörel yapıda olmalıdır. Teorik olarak CBS analizlerinde kullanılabilecek topolojiyi koruyan bir ağın oluşması için düğüm noktalarına ve bu noktaları birbirine bağlayan çizgilere ihtiyaç vardır. Bunun yanısıra nokta ve çizgilerin öznitelik bilgilerinin varlığı da ağ analizleri için bir zorunluluktur. Çünkü ağ analizleri bir anlamda grafik olmayan bilgiden grafik bilginin sorulması olup, birbirini izleyen çizgilerin öznitelik bilgilerinin karşılaştırılması şeklindedir (Yomralıoğlu 2000,s.225). Başlangıç noktasından itibaren gelen ilk düğüm noktasına bağlı olan çizgiler, istenen özellikleri bakımından irdelenir ve istenen özellikteki çizgi seçilerek bir sonraki düğüm noktasına ulaşılır. Ancak bu çalışma biçimine uygun oluşturulmuş bir veri yapısı CBS analizlerine imkan sağlar.

Sonuç olarak CBS açısından düşünüldüğünde hem geometrik zorunluluklar nedeniyle, hem de ağ analizlerinin yapılabilmesi için alan – çizgi dönüşümü zorunludur. Yol, akarsu vb. geometrik objelerin alan – çizgi dönüşümü sonucu

oluşturulan eksenleri, geometrik olarak bu objeleri temsil etmesinin yanısıra ağ analizleri için gerekli öznitelik bilgilerini de barındırmalı ve topolojiyi korumalıdır.

4. ALAN – ÇİZGİ DÖNÜŞÜM YÖNTEMLERİ

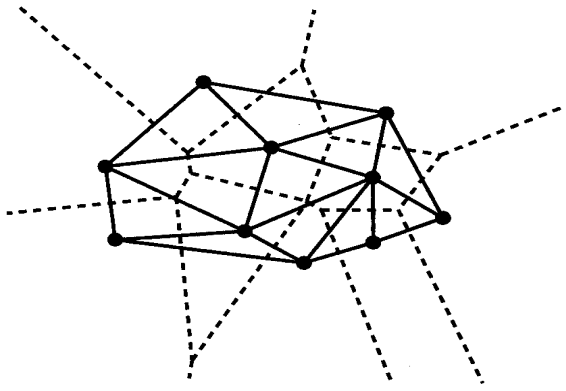
4.1.Üçgenleme Yöntemi

İngiliz uzmanlar arasında yoğun ilgi gören bu yöntem, Voronoi Diyagramı ve Delaunay Üçgenlemesi üzerine kurulmuştur. Yöntemin daha iyi anlaşılabilmesi için öncelikle bu iki kavramın tanımlanması gerekir. Aşağıda bu iki kavram üzerinde durulacaktır.

- **Voronoi Diyagramı ve Delaunay Üçgenlemesi**

Düzlemde yer alan sonlu nokta kümesine ait herhangi bir noktaya, kümedeki diğer noktalardan daha yakın konumda bulunan düzlem noktalarının geometrik yerine, o noktanın "Voronoi Çokgeni" denilmektedir. Kümedeki tüm noktaların Voronoi çokgenlerinin birleşimi, o kümenin Voronoi diyagramını oluşturur (Yanalak 1997).

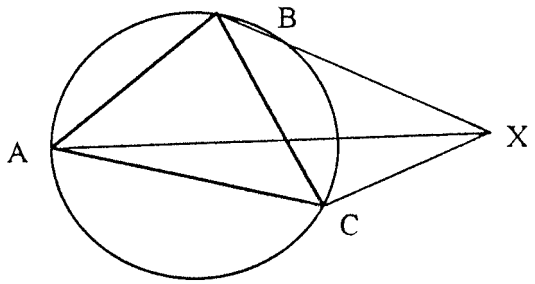
Şekil 4-1'de bir veri kümesi ve ona ait Voronoi diyagramı görülmektedir. Bu diyagram en yakın nokta problemleri için kullanılan kesin bir yapıdır. Bir noktanın Voronoi çokgeni o noktayı, komşu noktalar denen, o noktaya en yakın konumdaki noktalardan ayırmaktadır. Çokgenin kenarları, nokta ile komşu noktaları birleştiren doğru parçalarının kenar orta dikmelerinden oluşmakta, her nokta kendisine ait komşu noktalar ile birleştirildiğinde "Delaunay Üçgenlemesi" elde edilmektedir. Şekil 4-1'de Voronoi diyagramı verilmiş olan kümenin Delaunay üçgenleri görülmektedir.



Şekil 4-1 : Voronoi diyagramı (kesik çizgiler) ve Delaunay üçgenlemesi (kalın çizgiler) (McAllister ve Snoeyink, 2000)

Voronoi çokgenlerinin köşe noktaları, üçgenlerin kenar orta dikmelerinin kesişimiyle bulunan noktalardır. Dolayısıyla bu noktalar üçgenlerin çevrel çemberlerinin merkezleridir. Bu merkezler ait oldukları üçgenin içinde olabileceği gibi dışında da olabilir. Delaunay üçgenlemesinin algoritması kurulurken bir çok araştırmacı tarafından kullanılan en büyük özellik, üçgenlerin çevrel çemberi içerisinde bir başka nokta yer almamasıdır. Uygulamada buna daire kriteri adı verilmektedir. Üçgenlemeyi çekici kılan bir başka nokta, oluşan üçgenlerin eş kenar üçgene en yakın üçgenler olmasıdır. Çok dar açılı üçgenlerin oluşumu, dolayısıyla, birbirlerine uzak olan ve direkt ilişkisi bulunmayan noktalar arasında doğrusal bir ilişki kurulması engellenmektedir.

Uygulamada daire kriteri şu şekilde sağlanır.Üçgenin üçüncü köşesi olma olasılığı bulunan noktalardan birisi (Şekil 4-2 X noktası) sistematik olarak seçilir. Bu üç noktadan geçen çevrel çemberin içerisinde bir başka nokta olup olmadığı kontrol edilir. Çemberin içerisinde bir başka nokta (Şekil 4-2 B noktası) olduğu tespit edilirse, belirlenen bu nokta yeni oluşacak üçgenin üçüncü noktası olmaya aday olur. Yeni aday üçgenin çevrel çemberi içerisinde bir başka nokta olup olmadığı kontrol edilir. Bu işlem çevrel çemberin içerisinde nokta kalmayana kadar devam eder ve üçgen oluşur **(**Şekil 4-2**)**.



Şekil 4-2 : Delaunay üçgenlemesinde daire kriteri (Yanalak, 1997)

- **Yöntem :**

Bir poligonal çizgi veya bir çoklu doğru, köşe adı verilen nokta dizisi ile tanımlanır ve bu köşeleri birleştiren doğru parçaları kenar olarak isimlendirilir. Poligon kapalı bir dizidir, son köşesiyle ilk köşesi aynı olan çoklu doğru, poligon tanımlar. Herhangi

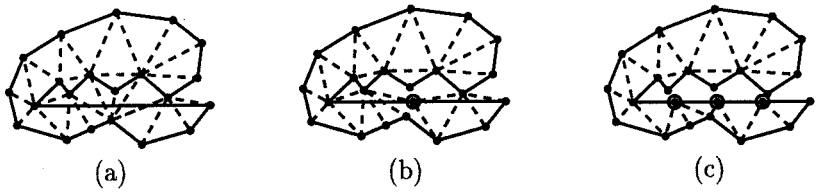
bir poligonun orta ekseni, poligonun içinde kalan iki veya daha fazla poligon kenarına teğet olan daire merkezleri kümesidir. Her daire için, daire merkezi, değdiği iki kenara eşit uzaklıktadır (McAllister ve Snoeyink 2000).

Orta eksenin belirlenebilmesi için Voronoi diyagramı ve Delaunay üçgenlemesi çok önemlidir. Matematiksel olarak Voronoi diyagramı ve Delaunay üçgenlemesi bir çifttir. T Delaunay üçgenini tanımlayan 3 Voronoi bölgesi, aynı zamanda T'nin çevrel çemberinin merkezindeki Voronoi köşesini de tanımlarlar **(**Şekil 4-1**)**.

Bir poligonun orta ekseni, köşeleri poligon köşesi olan ve kenarları açık olan bölgelerden oluşan Voronoi diyagramının bir alt kümesidir. Yani, Voronoi diyagramının poligonun içine uzanan parçası, poligonun orta eksenine yakınsar. Bu nedenle orta eksen Voronoi diyagramından türetilebilir.

Poligon sınırlarındaki Voronoi diyagramından, poligonun iç bölgeleri için bir eşleşmeli Delaunay üçgenler grubu üretilir ve bu üçgenleme orta eksene ulaşmada yol gösterir.

Poligon sınırına daha fazla nokta eklemek bozulmaları ve hesaplama zamanını artırır. Eğer poligon konveks değilse, üçgenleme sırasında istenmeyen durumlar ortaya çıkacağından sınıra noktalar eklemek gerekebilir. Örneğin şekil konveks olmadığı için karşılıklı olmayan kenarlar, karşılıklı kenarmış gibi üçgenlenebilir **(**Şekil 4-3**)**. Bunu önlemek için poligon sınırındaki kırık noktalarda Voronoi diyagramı hesaplandıktan sonra, sınırdaki Voronoi bölge çiftine ait her bir Delaunay üçgenlemesi birbiriyle karşılaştırılır. Eğer poligon sınırları herhangi bir Delaunay üçgenini kesmişse, sınır kenarı orta noktasından ikiye bölünür ve kenara bir orta nokta eklenir. Daha sonra poligon sınırı için Voronoi diyagramı tekrar hesaplanır. Poligon sınırı Delaunay üçgenlerini kesmeyinceye kadar bu işlem devam eder **(**Şekil 4-3**)**. Bütün bu işlemlerden sonra poligon iç bölgesi Delaunay üçgenleriyle kaplanmış olur ve uygun Voronoi kenarları poligon içerisinde yaklaşık bir eksen oluşturur.



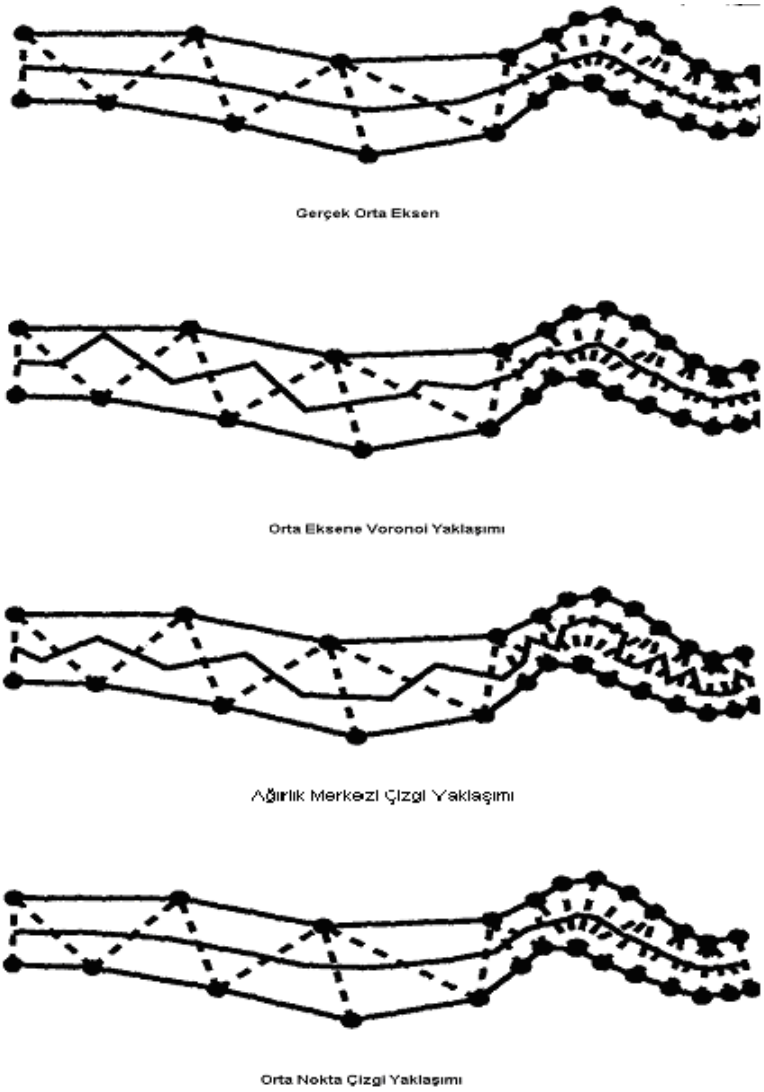
Şekil 4-3 : Delaunay üçgenlemesinde kenara nokta eklenmesi (McAllister ve Snoeyink 2000)

Voronoi kenarlarının eksen olarak kullanılmasında McAllister ve Snoeyink (2000) 3 farklı yaklaşımdan bahsetmişlerdir **(**Şekil 4-4**)**.

İlk yaklaşımda, belirlenen Delaunay üçgenlerine uygun Voronoi köşeleri birleştirilerek eksen oluşturulur. Yani üçgenlerin çevrel çemberlerinin merkezleri birleştirilerek eksen oluşturulur **(**Şekil 4-4**)**. Buna "orta eksene Voronoi yaklaşımı" adı verilir. Fakat bu yöntem sınır kenarlarındaki kırık noktalar arası mesafeler orantısızsa, yani iki nokta arası mesafe kısa iken ardından gelen iki nokta arası uzaklık uzunsa, oluşan orta eksen umulan yumuşak orta eksen yerine oldukça sert köşelere sahiptir.

İkinci yaklaşım ise "ağırlık merkezi çizgi yaklaşımı"dır. Bu yaklaşımda Voronoi kenarları, Delaunay üçgenlerinin ağırlık merkezlerinde birleştirilirler **(**Şekil 4-4**)**. a, b, c köşeleri ile verilen bir üçgende, ağırlık merkezi (a+b+c)/3, daima üçgenin içine düşer. Ağırlık merkezi üçgeni temsil eden bir nokta olmasına rağmen, ağırlık merkezlerinden geçen eksen de yumuşak değildir. Üçgenin bir kenarı diğer iki kenarından çok küçükse yine eksen zigzaglar oluşturur **(**Şekil 4-4**)**.

Üçüncü yaklaşım ise "orta nokta çizgi yaklaşımı"dır. Oluşturulan Delaunay üçgenlerinden bir kenarı sınır çizgisine ait olanlar izlenerek eksenler üretilir **(**Şekil 4-4**)**. Bu izleme sırasında sınır çizgisine ait olmayan üçgen kenarlarının orta noktaları ekseni oluşturur (Bildirici 2000,s.40).



Şekil 4-4 : Üçgenleme yönteminde orta eksen yaklaşımları

Eğer sınırdaki kırık nokta dağılımı uygunsa özellikle üçüncü yaklaşımla eksene çok iyi bir yakınsama sağlanır. Fakat eksen kesişimleri ve poligon kenarlarındaki ani çıkıntılar ayrıca değerlendirilmelidir. Eksen kesişimlerinde, kesişim bölgelerinde her poligona ait eksen son noktasının ortalaması alınarak tek bir orta nokta kesişim noktası olarak belirlenir. Poligon kenarlarında ani girinti çıkıntılar olması durumunda ise bu girinti çıkıntıların ihmal edilerek üçgenlemenin yapılması daha iyi sonuçlar verir.

Bu yöntemin dezavantajları ise poligonun karşılıklı kenarlarında uygun noktalar belirlemenin zorluğu ve çok girintili çıkıntılı poligonlarda iyi sonuç vermemesidir. Ayrıca yöntemin bir eksiği de ada vb. poligon içerisindeki iç alanların otomatik olarak belirlenememesi ve bu gibi yerlerde elle müdahale gerektirmesidir. Bu da pratik bir çözüm değildir.

4.2.Su Çizgileri Yöntemi

Günümüzde halen eksenlerin kısmen veya tam otomatik olarak tanımlanabildiği ticari yazılımlar yoktur. Bu nedenle yardımcı yazılımlarla eksene en yakın sonuçlar elde edilmeye çalışılmaktadır. Burada amaç sınır çizgilerini oluşturan parçalardan yararlanılarak eksenin tanımlanmasıdır. Bu ise özellikle sınırlar çok girintili çıkıntılı ve alan büyükse oldukça zahmetlidir. Kesişim bölgelerinde eksenlerin birleşmesi ayrıca çok farklı bir çözüm düşüncesi ve emek gerektirir. İşte bu yöntem tüm bu sıkıntılara olabildiğince çözüm getirmek üzere geliştirilmiştir.

Su çizgileri yöntemiyle (waterlining method) elde edilen eksen, "sağ ve sol tarafındaki parametrik bölümlere eşit uzaklıktaki noktaların yerini gösteren bilgisayar yapısıdır" diye tanımlanabilir (Christensen 1996). Eğer orijinal şekil çok girintili çıkıntılı değilse, bu yöntemle elde edilen eksenden yararlanılarak, orijinal şekil, su çizgileri arasındaki mesafe veri tabanında kaydedildiğinden dolayı herhangi bir ek veriye ihtiyaç duyulmaksızın tekrar elde edilebilir. Ayrıca bu yöntemde elde edilen eksen, yumuşamadan dolayı orijinal şekildeki aşırı girinti ve çıkıntılardan fazla etkilenmez, böylece istenmeyen kollar oluşmaz.

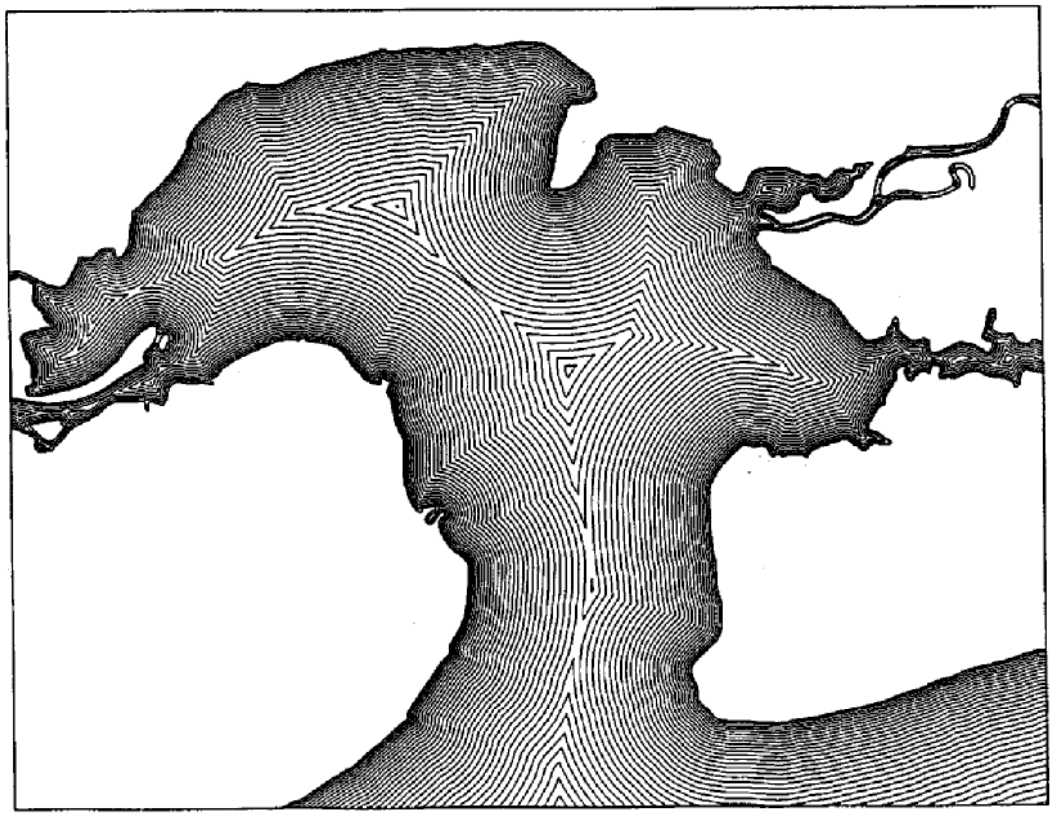
Su çizgileri yöntemi, üçgenleme yönteminin kenarlar üzerinde çok fazla nokta tanımlamaya gereksinim duyması, yani kenarları doğru parçalarıyla tanımlamasının ve kenarların çok girintili çıkıntılı olması durumunda iyi sonuçlar vermemesinin bir eksiklik olduğu düşünülerek Christensen tarafından geliştirilmiştir (Christensen 1996).

Orta eksen uygulamalarındaki ana sorun genelde aynıdır. Tüm şekli bilgisayar nasıl olup da görecek? Tüm şekli bütün parçalarıyla bilgisayara tanıtan bir yazılım bugüne kadar geliştirilememiştir. En uygun çözüm, veriyi küçük ve daha kolay ifade edilebilen parçalara bölmektir. Fakat üçgenleme yöntemindeki gibi kenarları doğru parçalarıyla ifade etmek çok yoğun bir emek gerektirir. Bu nedenle Christensen, kenarların çoklu doğru parçaları olarak tanımlanmasının uygun olduğunu düşünmüş ve GIS yazılımlarındaki "buffer" işlemi yardımıyla tanımladığı su çizgileri sayesinde eksenlerin belirlenebileceğini göstermiştir.

Bu algoritma şekli aşama aşama sorularla öğrenmekte, böylece düzenli olarak parametre boyutu azalmakta ve şekil basitleşmektedir. Orijinal şekle paralel iç içe geçmiş su çizgileriyle hem eksen oluşturulmakta hem de doğal bir basitleştirme sağlanmaktadır **(**Şekil 4-5**)**.

Bir şeklin su çizgilerinin oluşturulması orta eksen oluşturulmasındaki aşamalı yaklaşımın ilk adımıdır. Bir önceki su çizgisinden (veya şeklin kenarından) tek bir su çizgisi elde etme yöntemi, birçok GIS yazılımının içinde bulunan buffer (tampon) komutunun çalışmasına benzer. Burada ilk özellik, herhangi bir su çizgisi kenara sabit uzaklıktadır. Bir su çizgisinden diğerine olan uzaklık ise önceki iki su çizgisi arasındaki uzaklığa göre değişkendir. Şekil 4-5'de de görüldüğü üzere su çizgileri, karşı kenardan gelen su çizgileriyle çakışıncaya kadar kenara paralel olarak devam ederler. Çakışan su çizgileri, çakışma noktalarındaki ani yön değişikliklerini de gösterecek şekilde poligon içerisinde birleştirilir. Çakışma noktaları birleştirilerek, kenarlara yaklaşık eşit uzaklıkta, orta eksene güzel bir yakınsama elde edilir.

Şekil 4-5'deki gibi düzensiz şekillerin yorumlanması çok zordur. Bu gibi şekillerde eksen su çizgilerinin çakıştığı noktalar ve su çizgilerinin karakteristik noktalarını birleştiren doğru parçalarından oluşur **(**Şekil 4-5**)**. Diğer yönden yol ağları gibi kolaylıkla karakterize edilebilen şekillerde orta eksen sadece su çizgilerinin çakıştığı noktalarla tanımlanır. Burada belirtilmesi gereken bir unsur da, bu yöntemle elde edilen eksen, gerçek orta eksen değildir, fakat GIS ve CAD uygulamaları için gerçek orta eksene yeterli yaklaşıklıktadır.



Şekil 4-5 : Su çizgileri yöntemiyle eksen oluşturulması (Christensen, 1996)

Orta eksenin raster sistemlerdeki karşılığı iskelettir. Raster resmi incelterek oluşturulan iskeletin gerçek orta eksen olmamasını Christensen (1996) üç nedene bağlamıştır. İlki, genel olarak raster algoritmaları şekil kenarına olan sabit uzaklığı korumazlar. Yani iskelet karşılıklı kenarlara eşit uzaklıkta değildir. İkincisi, inceltme algoritmaları sonucu oluşan sonuç, piksellerin birleşmemiş topluluğudur ve kullanılabilir bir yapıda değildir. Üçüncüsü, inceltme algoritmaları kenardan iskelete olan uzaklığı kaydetmez. Eğer bu uzaklık gerekiyorsa sonraki aşamada ayrıca hesaplanmalıdır. Bu üç nedenden farklı olarak ayrıca inceltme yöntemi çözünürlük dikkate alındığında esnek değildir (yumuşatma, abartma gibi işlemler yapılamaz) ve eğer kullanılabilir bir yapıya ulaşılmak isteniyorsa, pikseller başarılı bir şekilde bağlantılı vektörlere dönüştürülmelidir. Bu ise inceltme yönteminin kolay olma özelliğini yok eder çünkü vektörizasyon oldukça zahmetli bir iştir.

Yukarıda sayılan nedenlerden dolayı su çizgileri yöntemi Christensen (1996) tarafından vektör esaslı olarak düşünülmüş ve vektör formatta uygulanmıştır. Bu yöntemin uygulamasının en kritik noktası parametre büyüklüklerinin seçilmesidir. Özellikle su çizgileri arasındaki uzaklığı belirlemek çok önemlidir çünkü eksenin konum hassasiyeti, bu parametreler için belirlenen değerlere bağlıdır.

Yöntemin işleyişi Christensen (1996) tarafından aşağıdaki gibi açıklanmıştır:

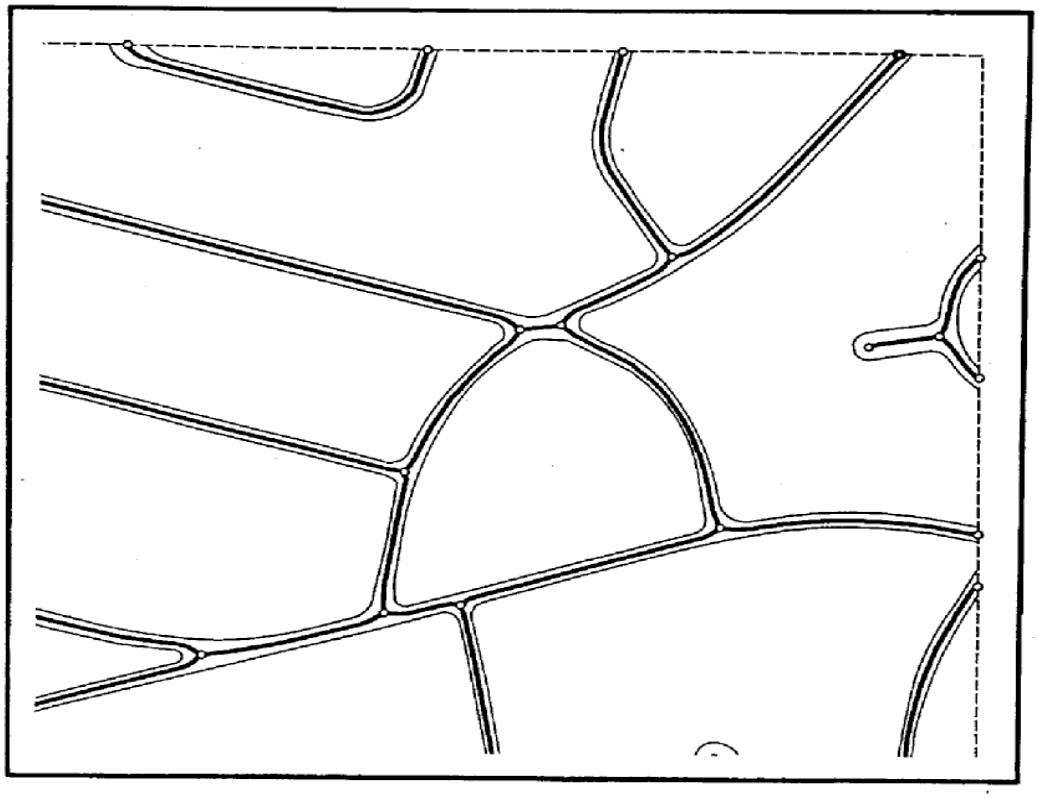
- Oluşturulan eksenin obje kenarlarına eşit uzaklıkta olmasını sağlamak amacıyla, obje kenarlarındaki sert çıkıntıların temizlenmesi. Eğer bu temizleme seçilmezse eksen şartı tam olarak sağlanmaz fakat uygulama hızlanır.
- Su çizgilerinin yoğun olduğu yerlerde orta eksen daha hassastır. Fakat yoğun su çizgileri daha fazla zaman gerektirir. Bu zaman kısıtlamasından dolayı orta eksende ideal hassasiyet elde edilemez. Bu durumda otomatik ayarlama (self calibration) çare olarak ortaya çıkar. Otomatik ayarlamada program, nerede ve ne zaman su çizgisi yoğunluğunun artması gerektiğine karar verir. Bu kararın verilmesi oldukça karmaşıktır. Ele alınan su çizgisi, köşe ve kenarlardaki eğrisel parçaların sayısı açısından bir önceki su çizgisiyle karşılaştırılır. Karşılaştırma sonrasında iki su çizgisi arasında köşe ve kenarlardaki eğrisel parça sayıları açısından fark belirli bir değerin üzerinde ise su çizgileri arasındaki mesafe azaltılır.
- Oldukça yoğun su çizgileri ile oluşturulan ağ derlenince, çok sayıda kol içerdiği görülür. Ağın manalı kolları dışındaki küçük kollar atılarak, ağ basit ve anlamlı hale getirilir.
- Su çizgilerinin ani sıçrama yaptığı bölgelerde, eksende saçma ve keskin köşeler oluşabilir. Bu düzensizlikler ağın genelleştirilmesinin son aşamasında otomatik olarak yumuşatılır. Diğer olabilecek düzensizlikler ise post process (sonradan değerlendirme) işlemiyle düzeltilir. Örneğin Şekil 4-6'daki eksenlerin birleştiği kesişim bölgelerinde eğriler görülmektedir. Bunun sebebi kenarlara olan eşit uzaklık şartının sağlanmak istenmesidir. Fakat bazı durumlarda kullanıcılar bu eğrilerin oluşmasını istemezler, bu nedenle bu eğri yaylarının düzeltilmesi post process ile gerçekleştirilir. Tabi bu durumda istenilen gerçekleşmiş olsa da gerçek eksenden uzaklaşılmıştır.

Yöntem giriş verilerinin düzenlenmemiş çoklu doğru olduğunu kabul eder. Çoklu doğrular su çizgisi oluşturma adımlarıyla otomatik olarak düzenlenirler. Bu adımların sonucunda her biri bir bölgeye (örneğin her bir yol ağına) ait dosyalar kümesi elde edilir. Bu dosyalar, taşınan geometriyi, su çizgilerinin obje kenarlarına olan uzaklığını, önceki ve sonraki su çizgilerine olan uzaklığı ve olabilecek diğer değerleri içerir.

Daha sonraki aşamada su çizgilerinin çakışma noktaları otomatik olarak seçilir. Çakışma noktaları birbiriyle ilişkilendirilir ve bu noktaları birleştiren eksen, su çizgilerine ait tüm özellikleri devralır. Örneğin kenarlara olan uzaklığı gibi. Bir sonraki aşamada ise ağın yaylarla ilişkisi kurulur. Ağ programının çıktısı dosyalar kümesidir.

1. Geometriyi içeren dosya
2. Kesişim bölgelerini (düğüm noktalarını) içeren dosya
3. Yayları içeren dosya
4. Tarih dosyası ve istatistikler

Bu yöntem eğer çok büyük veri gruplarına uygulanacaksa, bu veri grupları parçalanmalıdır. Çünkü çok büyük veri gruplarına bu yöntemin uygulanması büyük bilgisayar belleği gerektirir. Verilerin parçalanması durumunda, süreklilikten emin olmak amacıyla komşu paftalardan örtüşen bölümler belirlenir. Yöntemin uygulamasında eğrisel parçaların sonlarına düğüm noktaları eklenir ve bu düğüm noktaları ile komşu paftalarla ilişki kurulur **(**Şekil 4-6**)**. Yöntemin uygulamasında, en fazla su çizgilerinin oluşturulmasında zaman harcanır, diğer aşamalar buna göre daha kısa zamanda tamamlanır.



Şekil 4-6 : Su çizgileri yöntemiyle oluşturulan bir yol ağı ve kesişim bölgelerinde oluşan yaylar (Christensen 1996)

Su çizgileri yönteminin en büyük avantajı, sistemin esnek olmasıdır. Obje kenarına olacak herhangi bir ilave sadece eksenin o bölgesini etkiler, tüm ekseni etkilemez. Yine tüm veriler depolandığından istenildiğinde eksenden yararlanılarak orijinal şeklin elde edilmesi mümkündür.

4.3.En Büyük Daire Yöntemi

Alan objeler için eksen tanımlarından birisi de, objelerin içerisine çizilebilecek en büyük dairelerin merkezlerini birleştiren doğru parçalarının obje eksenini tanımlamasıdır.. Burada en büyük daire, alan objesinin karşılıklı kenarlarına teğet olan dairedir **(**Şekil 4-7**)**.



Şekil 4-7: En büyük dairenin özellikleri (Sonka ve ark.1999, s.577)

Bu yöntemin verinin hem vektör hem de raster formatta olmasına göre uygulaması vardır. Ancak raster formattaki uygulaması hem matematiksel içerik hem de veri gereksinimi açısından daha kullanışlıdır. Aşağıda en büyük daire yönteminin vektör ve raster formatta uygulanması anlatılacaktır.

4.3.1. Vektör formatta en büyük daire yönteminin uygulanması

Vektör formatta en büyük daireyi tanımlamak oldukça zahmetlidir. Çünkü daire, merkezi, yarıçapı ve objeye teğet olduğu noktalarla tanımlanmak zorundadır. Burada merkez bileşeni eksen üzerinde olan, bizim aradığımız değerdir. Yarıçap değeri de uygulama da çoğu zaman sabit değildir, çünkü yarıçapın sabit olması, alan objesinin genişliğinin değişmediği durumlarda söz konusudur. Bu da en azından, vektör formatta en büyük daire tanımının, yarıçapın değişmediği aralıklar boyunca ayrı ayrı yapılması gerekliliğini ortaya çıkarır. Bu da alan objesinin çok girintili çıkıntılı olduğu bölgelerde imkansızdır. Bu gibi yerlerde kabul edilen bir doğruya olan uzaklık yarıçap olarak düşünülüp en büyük daire tanımlanabilir.

Vektör formatta en büyük daire tanımlamasında bir diğer sorun da dairenin objeye teğet olduğu en az iki teğet noktanın tanımlanma zorunluluğudur. Çünkü tek teğet nokta dairenin obje içerisinde kalmasını garanti etmez. Bu nedenle dairenin alan objenin her iki tarafında teğet olduğu karşılıklı noktalar tanımlanmalıdır. Bu da

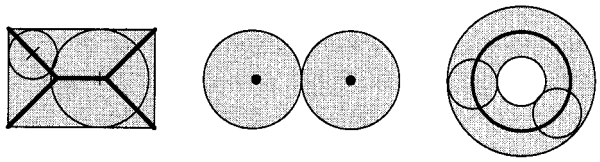
oldukça zahmetli ve ek uğraş gerektiren bir iştir. Ayrıca alan objenin karşılıklı kenarlarında karşılıklı noktalar belirlendikten sonra Delaunay üçgenlemesi çok rahat yapılabileceğinden üçgenleme yöntemi daha tercih edilen bir yöntem haline gelir.

Tüm bu zorluklara rağmen eğer alan obje içerisine yeterli sıklıkta en büyük daireler oluşturulabilirse, bu dairelerin merkezlerini birleştiren doğru parçaları ekseni tanımlar. Yukarıda anlatılan nedenlerle bu yöntem diğer yöntemlere göre çok rasyonel değildir. Bu nedenle uygulamada en büyük daire yöntemi genellikle raster formatta uygulanmıştır.

4.3.2. Raster formatta en büyük daire yönteminin uygulanması

Raster formatta, B(p,r) dairesi, p merkezli $r \geq 0$ yarıçaplı, merkezden $d \leq r$ uzaklığında bulunan noktalar kümesiyle tanımlıdır. Bu B dairesi eğer alan obje içerisinde ondan daha büyük bir daire tanımlanamıyorsa en büyük dairedir **(**Şekil 4-7**)** (Sonka ve ark. 1999, s.577).

Burada d uzaklığı, merkezden 6, 4 veya 8 farklı doğrultuda hesaplanabilir. Bu şekilde tanımlanan en büyük dairelerin merkezlerinin oluşturduğu küme, ekseni tanımlar. Eğer alan obje daire şeklindeyse iskelet, doğru parçası şeklinde değil de nokta şeklinde oluşur **(**Şekil 4-8**)**.



Şekil 4-8: En büyük daire yöntemiyle oluşturulan eksen örnekleri (Sonka ve ark. 1999, s.577)

Raster formatta da en büyük daire yöntemiyle iskelet oluşturmanın uygulamada iki dezavantajı vardır. Bunlardan ilki gerekli şekilde sürekliliğin korunamaması, ikincisi oluşturulan bazı iskeletlerin istenilen bir piksel kalınlığından daha kalın olmasıdır. Ayrıca Şekil 4-8'in ilk şeklinde görüldüğü gibi alan obje yeteri kadar en büyük daire

tanımlamaya elverişli değilse, eksen tanımlamak için ek küçük daireler tanımlamak gerekir. Bu durumda oluşan dairelerin merkezlerini birleştiren doğru parçası da eksen oluşturmaz, daireler kendi içerisinde tekrar değerlendirilerek eksen elde edilmelidir.

Raster formatta bu yöntemin uygulanmasının avantajlı tarafı ise alan objenin çok girintili çıkıntılı olması durumunda en büyük daire bu girinti ve çıkıntılardan etkilenmeyeceğinden, oluşan eksen gereksiz kollar içermez ve yüksek hassasiyete sahiptir.

4.4. İnceltme (Thinning) Yöntemi

Haritalar sık sık uydu görüntülerinden veya hava fotoğraflarından yararlanılarak elde edilir. Sadece elle sayısallaştırma değil, aynı zamanda uzaktan algılama ve yapay zeka teknikleri hava fotoğraflarından farklı bilgilerin ayrılmasında kullanılır. Bu bilgiler birçok bilgi sistemi çalışmasının temelini oluşturur.

Orta ölçekli (1:25000 ölçeği civarı) turistik şehir planları tarzındaki haritalar genellikle hava fotoğraflarının standart elle sayısallaştırılması ile elde edilir. Çünkü haritaların vektörel gösterimi kullanıcı için büyük avantajlar sağlar. Vektörel gösterimin depolama maliyeti oldukça az, ölçeği basit ve geometrik objelerinin değiştirilmesi kolaydır. Bununla birlikte tamamen farklı vektörel çizimlerden elde edilen sonuçlar aynı objeymiş gibi yorumlanabilir. Örneğin çok farklı bağımsız geometrik objeler, aynı grafik görüntüye sahip olabilir. Bir üçgen, üç kenarlı poligonal bir bölge ile veya üç kenarlı bir çoklu doğru ile veya uygun doğru parçalar kümesi ile gösterilebilir. Bütün bu gösterimler ekranda aynı görüntüyü verse de bilgisayar belleğinde taşıdığı anlamlar farklıdır. Bu genellikle sayısallaştırmada bir standardın olmamasından ve sayısallaştırmada yapılan hatalardan kaynaklanır. Raster gösterimdeki otomatik çizim algoritmalarının gerekliliğinin nedenlerinden birisi budur.

Elle sayısallaştırma ile elde edilen görüntü, orijinal vektör görüntünün doğru bir görüntüsü gibi görülse de, bu yaklaşık görüntü içerisinde umulmayan durumlar içerir. Bazı geometrik parçalar iki kere çizilebilir, elemanlar arasında kolaylıkla

görülemeyen kesişmeler oluşabilir veya kopukluklar meydana gelebilir. Bu veri otomasyon hataları Thomas (1998) tarafından 3 şekilde sınıflandırılmıştır.

1. **Kesiklik ve fazlalık:** Çizgilerin çiziminde gerekli hassasiyetin gösterilmemesinden kaynaklanır. İki çizgi bir noktada birleşecekken çizginin birisinin kısa kesilmesi (undershooting), veya fazla uzatılarak kesişme oluşturulması (overshooting). Bu problemin çözümü için geliştirilen yazılımlar çoğunlukla yeterli olmamakta ve elle düzeltmeler gerekmektedir.
2. **Çakışma :** Çakışma verinin iki kere çizilmesine neden olur. Bir geometrik obje, diğer bir geometrik objenin tamamının veya bir kısmının üzerinden geçerse buna "çakışma" denir. Genellikle çakışma gereksiz bilgilere neden olur, sonuçta çakışma içeren vektör gösterim basitleştirilerek (çoğunlukla çakışmaya neden olan obje silinerek) çakışma içermeyecek hale getirilir.
3. **Tabakanın yanlış belirlenmesi :** Geometrik objenin yanlış bir tabakaya tahsis edilmesiyle ortaya çıkar. Problem, bu yanlış tahsis edilen objenin orijinal şeklinin değişmesi durumunda ortaya çıkar ve objenin tanınmasını zorlaştırır.

Vektör haritalarda karşılaşılan bu hatalar nedeniyle, vektörleri doğrudan dönüşümle tek değere indirgenmiş siyah - beyaz görüntü (run-length encoded binary image (RLE)) haline dönüştüren bir yöntem geliştirilmiştir. Burada tek değere indirgenmiş siyah - beyaz görüntü, görüntü boyunca tüm piksellerin 0 ve 1 sayısı ile ifade edildiği ve kapama ve silme operatörleri ile hem siyah hem beyaz içeren piksellerin yalnız beyaz veya siyaha dönüştürüldüğü görüntülerdir. Vektörel gösterimdeki hataları morfolojik operatörler yardımıyla gidermek ve alan - çizgi dönüşümünü sağlamak bu yöntemle mümkündür. Burada sözü geçen morfolojik operatör kavramı çok geniş bir içeriğe sahiptir. Bu yöntemde bu operatörlerden sadece genişletme (dilation) ve silme (erosion) operatörleri kullanılmıştır.

- **Genişleme (dilation) :** Bir bölgenin parçası olan bölge piksele, arka alan pikselin az miktarda girmesi durumunda, arka alan piksel bölge pikseli olarak etiketlenir. Böylece bölgelerde oluşabilecek küçük boşluklar ortadan kaldırılır.

- **Silme (erosion) :** Arka alanın bir parçası olan piksele, obje pikseli az oranda girerse, objenin o piksel içerisindeki kısmı silinir ve piksel arka alan piksel olarak etiketlenir.

Hesaplama maliyetini ve bellek gereksinimini en az tutmak amacıyla bu yöntemde morfolojik operatörler doğrudan görüntüye uygulanır. Kısacası vektörel harita önce raster formata dönüştürülür, daha sonra morfolojik operatörler uygulanır, bunun ardından inceltme işlemi ile iskeletleştirme yapılır ve tekrar vektör formata dönülür.

Bu yöntemin işleyişi Thomas (1998) tarafından 4 aşamada incelenmiştir.

1. RLE

2. Alanların bölümlenmesi

3. Alanların iskeletinin çıkarılması

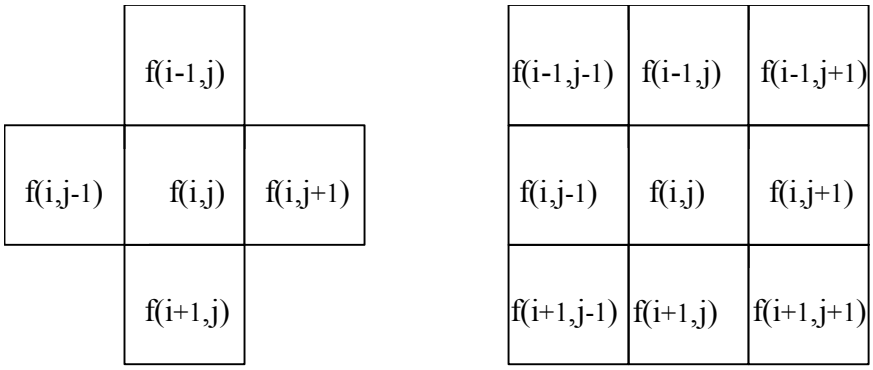
4. İskeletleşmiş elemanların vektörizasyonu ve topoloji elde edilmesi

Bu işleyiş içerisinde yukarıda bahsedilen hatalar ilk iki aşamada yok edilir.

4.4.1. RLE (Run Length Encoding) yöntemi

RLE, verilen çözüm seviyesinde vektör haritaları, haritayı kapsayan gridler şeklinde siyah - beyaz görüntüye dönüştürür ve haritanın geometrik elemanlarını kapsayan bu grid hücrelerle (pikseller) haritayı tanımlar (Thomas 1998).

Bu işlemde öncelikle görüntü taranır, arka alan piksellere 0 değeri, geri kalanlara 1 değeri verilir. Tarama bitince görüntü aldatması nedeniyle yanlış etiketlemenin önüne geçmek için tekrar tarama yapılır ve pikseller 4 veya 8 komşulu olarak etiketlenir **(**Şekil 4-9**)** (Sonka ve ark. 1999, s.233) .

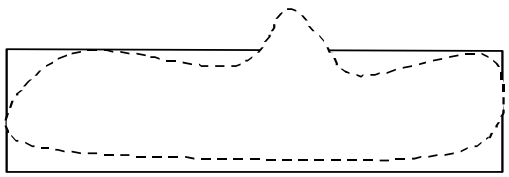


Şekil 4-9 : Piksellerin 4 veya 8 komşulu olarak etiketlenmesi

Basit bir RLE kodu aralığın başlangıç ve bitiş adresiyle tanımlanabilir. Yani objenin başlangıç ve bitiş pikseli objeyi tanımlar. Tam bir RLE uygulanırsa, boşluk boyutu piksel boyutunun yarısından daha küçük ise kesilme hatalarının bir çoğu düzeltilir. Ayrıca yığın yani üst üste gelmiş çizgiler tek değere indirgenmiş görüntüyü hiç etkilemez. Çünkü adresleri aynı olduğundan sadece biri değerlendirmeye alınır.

Bu işlem esnasında uygun raster çözünürlüğüne karar vermek de önemlidir. Bu çözünürlük mümkün olan en düşük çözünürlük olmalıdır. Çünkü çözünürlüğün artması bellek gereksinimini artırır ve bilgisayar hızını olumsuz etkiler. Yalnız, seçilen çözünürlükte kesilme hataları görülmemeli ve çizgiler birbirini tutacak şekilde topoloji korunmalıdır.

Çözünürlük düştükçe şekiller yumuşar. Bu yumuşama gereksiz iskelet parçalarının oluşması ihtimalini azaltır. Çünkü şeklin kenarındaki tek ekstra piksel, ekstra iskelet ayaklarına neden olur. Eğer şekil çok fazla girinti ve çıkıntı içeriyorsa, bu hatanın oluşmaması için şeklin önce bir dış tamponla (buffer) çevrelenip daha sonra iskeletleştirilmesi daha iyi sonuç verir **(**Şekil 4-10**)** (Sonka ve ark. 1999, s.258) .



Şekil 4-10: Bozuk şekillerin tamponla çevrilmesi

Bu işlem adımında morfolojik operatörlerin uygulanmasıyla kesilme hataları ortadan kalkar (genişletme olayı). Arta kalan hatalar yani üst üste binmeler ve tabaka yerinin yanlış seçilmesi hataları alanların bölümlenmesinden sonra ortadan kalkar.

4.4.2. Alanların bölümlenmesi

Kesilme hatalarından arındırılmış vektör haritaların tek değere indirgenmiş versiyonu, düzlemi birbirine bağlı bölgelere bölen kapalı çizgiler kümesi içerir. Bu bölgeleri birbirinden ayırmak amacıyla öncelikle görüntü tersine çevrilir. Böylece çizgiler arka alana alınır ve sonuçta oluşan birbirine bağlı bölgeler ön plana çıkarılır. Bu bölgeler etiketleme algoritması yardımıyla birbirinden ayrılır.

Tek değere indirgenmiş görüntülere uygulanan etiketleme algoritmasının ilk bölümü, birbirini takip eden çizgilerin oluşturduğu komşu pikselleri seçer, onlara aynı etiketi verir ve aynı etiketlilerin bir listesini çıkarır. Algoritmanın ikinci parçası ise aynı etikete sahip bütün pikselleri tek bir bütünün parçası olacak şekilde etiketler. Örneğin şehir haritalarında yolların çevrelediği ada bloklarının isimlendirilmesi gibi **(**Şekil 4-15**)**.

Bu etiketleme sayesinde üst üste binmeler ve tabaka yerinin yanlış seçilmesi gibi hatalar giderilir. Eğer halen blok içlerinde istenmeyen boşluklar varsa genişletme operatörleri kullanılarak blok bütünlükleri sağlanır. Bundan sonra görüntü, asıl amaç olan iskeletleştirmeye hazır hale gelmiş olur.

4.4.3. Alanların iskeletinin çıkarılması

Arzu edilen iskelet, 1 piksel kalınlığında ve ağ topolojisini koruyan bir yapıda olmalıdır. Yani iskelet, 1 piksel kalınlığı sağlarken iskelet üzerindeki noktaların bağlantısı kesilmemelidir. İskelet oluştuğunda, iki komşudan daha fazla pikselle, sadece farklı iskelet parçalarının buluştuğu bölgelerde karşılaşılır. Örneğin şehir haritalarında yolların kesiştiği kavşak bölgeleri gibi. İstenilen iskeleti elde etmek

için, bölgedeki dış piksellerin ardışık olarak atılmasından ibaret olan bir inceltme yöntemi uygulanır. Daha fazla atılacak piksel kalmayıncaya kadar işleme devam edilir.

Bu bölümde objelerin kenar piksellerini, tek bir iskelet kalıncaya kadar silen algoritmalar anlatılacaktır. Silinecek piksel P, bir siyah pikseldir ve P'nin silinmesi, P'yi içeren yerel komşu piksellerin konfigürasyonuna bağlıdır **(**Şekil 4-11**)**. Pikselleri incelemelerine göre inceltme algoritmaları Lam ve ark.(1992) tarafından sıralı ve paralel olarak sınıflandırılmıştır. Sıralı algoritmalarda pikseller, her iterasyonda değişmeyen sabit bir sıra ile incelenerek silinip silinemeyeceklerine karar verilir. n. iterasyonda P'nin silinebilmesi, belirli bir yere kadar gerçekleştirilmiş olan iterasyonların sonuçlarına bağlıdır. Örneğin (n-1). iterasyonun sonucu n. iterasyondaki pikseli etkiler. Diğer yönden paralel algoritmalarda, n. iterasyonda pikselin silinmesinde sadece (n-1). iterasyondan arta kalanlar etkilidir. Bu nedenle, her iterasyonda bütün pikseller paralel bir düzende bağımsız olarak incelenir.

Silinmek için incelenen P pikseli bir siyah pikseldir ve etrafında 3*3 lük bir pencere şeklinde etiketlenmiş pikseller bulunur **(**Şekil 4-11**)**. x_1, x_2, ….., x_8 P'nin 8 komşusudur ve ortak olarak N(P) (neighbours of P) ile gösterilir. x_1, x_3, x_5, x_7 P'nin 4 bitişik komşusudur.

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

Şekil 4-11: Piksel ve komşuları (N(P)) (Lam ve ark. 1992)

N(P) içindeki siyah piksellerin sayısı b(P) ile gösterilir. P'nin bağlantı sayısı, P'nin tümleyeni $\overline{P}$'nün bileşen sayısıdır ve eğer bu sayı 1 ise P'ye basit bağlantılı (simple), değilse çok bağlantılı adı verilir.

8 veya 4 bağlantılı bir piksel eğer silinmesi durumunda P pikselinin 8 veya 4 bağlantısını değiştirmiyorsa silinebilir. Piksellerin silinmesi için kenar piksel olması gerekir. Kenar pikseller genellikle en az biri beyaz 4 komşu ile tanımlıdır. Yalnız eğer b(P)=1 ise pikseli, kenar piksel olarak yorumlayıp silmeden önce, pikselin bir son piksel olup olmadığı da incelenmelidir. Çünkü topolojinin korunması amacıyla son pikseller silinmemelidir. Bunun belirlenmesi de son piksel şartlarına bağlıdır. Bu şartlar değişik yazarlar tarafından farklı olarak belirlenmiştir. Ama genel olarak P pikseli, N(P)'nin bir kenarında iki veya üç ardışık siyah piksel varsa korunur ve daha sonra tekrar değerlendirilerek silinip silinemeyeceğine karar verilir.

İnceltme algoritmalarında önemli bir tanım da "geçiş sayısı"dır. Geçiş sayısının iki tanımı vardır. Rutovitz, N(P) içindeki beyaz pikselden siyah piksele geçis sayısını dikkate almıştır. Bu geçiş sayısı aşağıdaki gibi tanımlanabilir.

$$X_R(P) = \sum_{i=1}^{i=8} \left| x_{i+1} - x_i \right| \quad \text{..... (4.4.1)}$$

$X_R(P) = 2$ ise, P pikseli silinebilir.

Hilditch ise geçiş sayısını, N(P)'deki pikseller bir düzen içerisinde yer değiştirirken beyaz pikselden siyah piksele bir geçişin tekrar sayısı olarak tanımlamıştır.

$$X_H(P) = \sum_{i=1}^{i=4} b_i \quad b_i = \begin{cases} 1 & eger\ x_{2i-1} = 0\ ve\ (x_{2i} = 1\ veya\ x_{2i+1} = 1) \\ 0 & \text{değilse} \end{cases} \qquad (4.4.2)$$

Eğer $X_H(P) = 1$ ise, P pikseli silinebilir. İki sistemde de pikselin 8 komşusu da siyah ise geçiş sayısı sıfırdır ve piksel ayrık pikseldir.

$X_H(P)$ ile $X_R(P)$ arasındaki fark, $X_H(P) = 1$ şartı aynı zamanda P'nin bir kenar piksel olmasını zorunlu kılar. Yani silinmesi sorun çıkarmaz. Ancak $X_R(P) = 2$ durumunda P kenar piksel olmayabilir. Çünkü eğer P bir beyaz köşe komşuya sahipse, P bir kenar piksel olmasa da bu şart sağlanır **(**Şekil 4-12**)**. Bu durumda P'nin silinmesini engellemek amacıyla $b(P) \leq 6$ gibi bir ek şart gereklidir. Ancak bu durumda P'nin kesinlikle bir kenar piksel olduğu kanaati oluşabilir.

0	1	1
1	P	1
1	1	1

Şekil 4-12: Pikselin beyaz köşe komşuya sahip olma durumu

4.4.3.1.Paralel İnceltme Algoritması

Paralel inceltme algoritmasında, piksellerin silinip silinemeyeceği sadece önceki iterasyonun sonucuna bağlıdır. Yani her piksel, paralel bir düzende bağımsız olarak değerlendirilir. Her iterasyon sonucunda korunan pikseller bir sonraki iterasyon için güncellenir ve ikinci iterasyon güncellenmiş durumdaki N(P)' ye uygulanır.

Paralel algoritmada P pikselinin silinebilmesi için Lam ve ark.(1992) tarafından aşağıdaki şartların yerine gelmesi gerektiği belirtilmiştir.

- R1: $b(P) \geq 2$
- R2: $X_R(P) = 2$
- R3 : $X_1X_3X_5 = 0$
- R4 : $X_7X_1X_3 = 0$

Bu algoritmaya, P'nin bir kenar piksel (en az biri beyaz 4 komşusu olan piksel) olduğunu garanti etmek için $b(P) \leq 6$ şartı da eklenmelidir **(**Şekil 4-13**)**.

1	1	0
1	P	0
1	1	0

1	0	0
1	P	1
1	1	1

Şekil 4-13: Paralel inceltme algoritmasında silinme şartlarını sağlayan iki piksel örneği

4.4.3.2.Sıralı İnceltme Algoritması

Bu algoritmada öncelikle sınır izleme algoritmaları yardımıyla tüm sınırlar izlenir. Sınırlar boyunca silinebilecek pikseller belirlenir ve işaretlenir. Tek iterasyon sonucunda işaretli tüm pikseller silinir.

Bu yöntemde pikselin silinebilmesi için Lam ve ark.'na (1992) göre öncelikle şu özelliklere sahip olması gerekir:

- P bir siyah piksel olmalı
- P bir son piksel veya ayrık piksel olmamalı ($b(P) \geq 2$)
- P bir kenar piksel olmalı

Bu şartları sağlayabilmek için algoritma şu şekilde düşünülebilir **(**Şekil 4-14**)**.

- P'nin en azından bir komşusu işaretlenmemiş olmalıdır
- İterasyonun başlangıcında $X_H(P) = 1$ olmalıdır
- Eğer X_3 işaretliyse, X_3'ün sıfır olması $X_H(P)$'yi etkilememeli (bkz.4.4.2)
- Eğer X_5 işaretliyse, X_5'in sıfır olması $X_H(P)$'yi etkilememeli

0	1	1
0	P	0
0	0	0

Şekil 4-14: Sıralı inceltme algoritmasında silinme şartlarını sağlayan piksel örneği

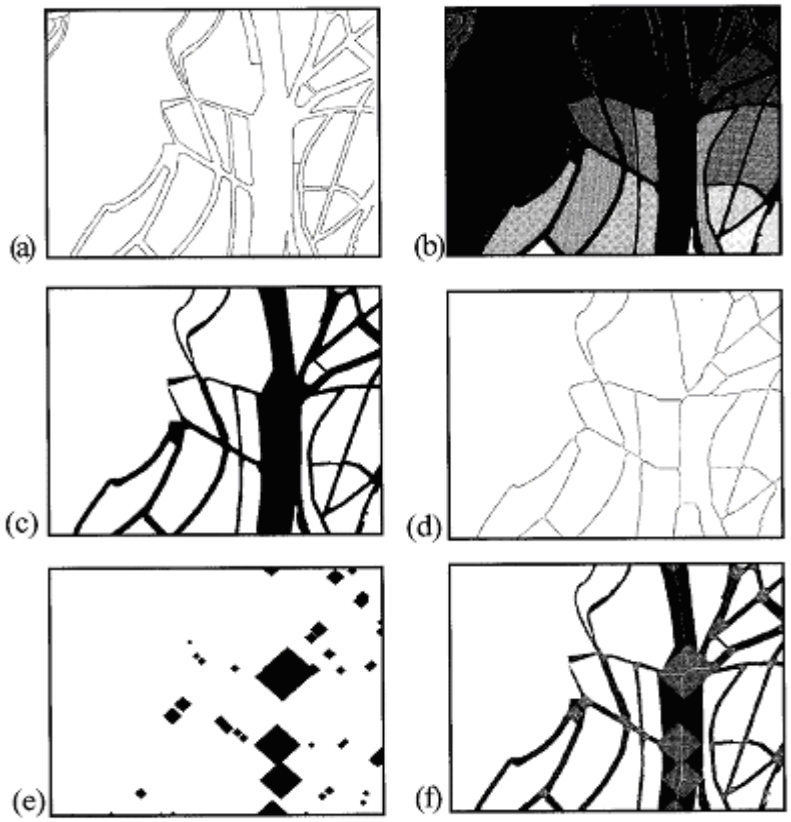
4.4.4. İskeletleşmiş elemanların vektörizasyonu ve topoloji elde edilmesi

Elde edilen iskeletin bir piksel kalınlığında olması nedeniyle her piksel, kesişim bölgelerine (yol haritalarında kavşak gibi) gelmediği sürece bir veya iki komşuya sahiptir. Bundan dolayı kesişim bölgelerinde her piksel, komşu sayılarına göre etiketlenerek, kesişim bölgeleri elde edilir. Sonuçta iskelet parçaları ve kesişim bölgeleri birbirinden ayrılmış olur **(**Şekil 4-15**)**.

Bu aşamadan sonra her iskelet parçasına bağımsız olarak muamele etmek mümkündür, kesişim bölgeleri çıkarılır ve tekrar etiketlenir. Tek değere indirgenmiş format kullanıldığından, her iskelet parçası, iyi bir şekilde düzenlenerek doğrudan

doğruya çoklu doğru haline dönüşebilen aralıklar listesi şeklinde düşünülebilir. Bir hat boyunca bu aralıklardan yalnızca iki tanesi "uç aralık"tır. Bu uç aralıkların son noktalarından bir tanesi diğer aralıklarla birleşmez. Hat boyunca bunun haricindeki tüm aralık son noktaları kesişim bölgelerinde birbiriyle birleştirilir.

Elde edilen kesişim bölgeleri bağımsız olarak, o kesişim bölgesindeki tüm iskelet parçalarını kapsayacak şekilde genişletilir. Böylece genişletilmiş kesişim bölgeleri elde edilir. Genişletilmiş kesişim bölgelerinin ağırlık merkezleri iskelet parçalarının kesişim noktasını oluşturur. Eğer iki kesişim bölgesine bağlı olan iskeletin genişliği uzunluğundan fazlaysa, genişletmeden sonra iki kesişim bölgesi örtüşmüş demektir ve tek kesişim bölgesi gibi kabul edilir **(**Şekil 4-15**)**. Bütün bunlardan sonra arzu edilen iskelet ve topoloji elde edilmiş olur.



Şekil 4-15: İnceltme yönteminin aşamaları a) Orijinal görüntü b)Bölgelerin etiketlenmesi (her farklı bölge ayrı gri tonda c) Görüntünün geri plana çekilmesi (bölgelerin arka alana alınması) d) İskeletin elde edilmesi e) Kesişim bölgelerinin genişletilmesi f) Son iskelet ağı (Thomas 1998)

4.5.Basit İskeletleştirme Ve Çatı Yöntemi

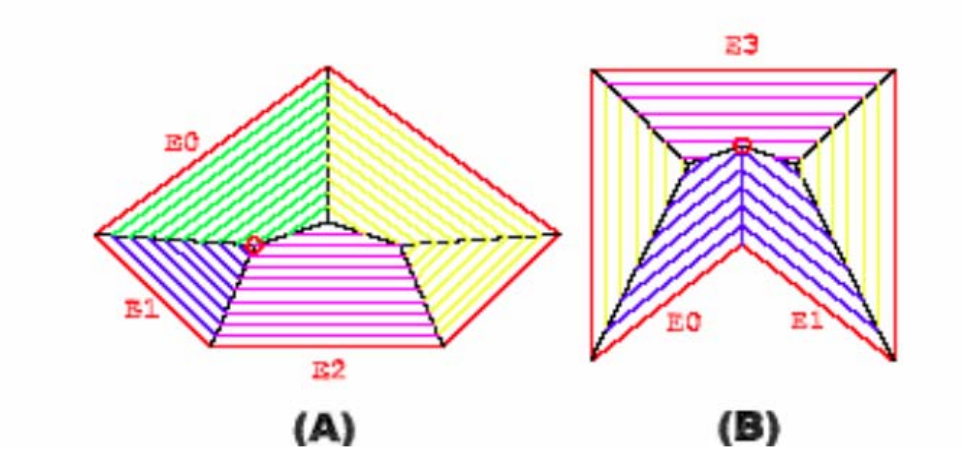
4.5.1. Basit iskeletleştirme yöntemi

Basit iskelet (straight skeleton) poligon kenarlarının açıortaylarının bir parçasıdır. P poligonu içerisinde benzersiz parçalardan oluşur ve P poligonunun her kenarına karşılık bir parça içerir (Aichholzer ve ark. 1995). Poligon kenarları poligon içerisine doğru sabit bir hızla hareket eder ve kenar uzunlukları değişir. Poligon köşeleri de poligonun topolojisi değişim aşamasına gelinceye kadar açıortay doğrultusunda hareket eder.

Basit iskeletleştirme P poligonunun içinde kalan tüm noktaları kullanan bilinen orta eksen uygulamalarından farklıdır. Genellikle orta eksen uygulamaları Voronoi diyagramı temelli iken, basit iskelet uzaklık fonksiyonu kullanılarak tanımlanmaz. Poligon topolojisindeki değişimler basit iskeleti tanımlar. Eğer poligon konveks ise orta eksen uygulamalarında elde edilen eksen ile basit iskelet yapısı örtüşür. Aksi halde basit iskelet, sert çıkıntılı bölgelerde (reflex vertices), orta eksen uygulamalarında karşılaşılan parabolik eğrileri içermez. Çünkü basit iskelet yapısı doğru parçalarından oluşur ve kesinlikle eğri içermez.

Basit iskeletin, P poligonunun sınırlarının poligon içerisine doğru paralel ve her kenarda aynı hızda olacak şekilde büzülmesiyle oluştuğunu söyleyebiliriz. Bu olay sınırların topolojisi değişmediği sürece devam eder. Sınırlardaki topoloji değişimleri basit iskeleti tanımlar. İki tür topoloji değişimi vardır.

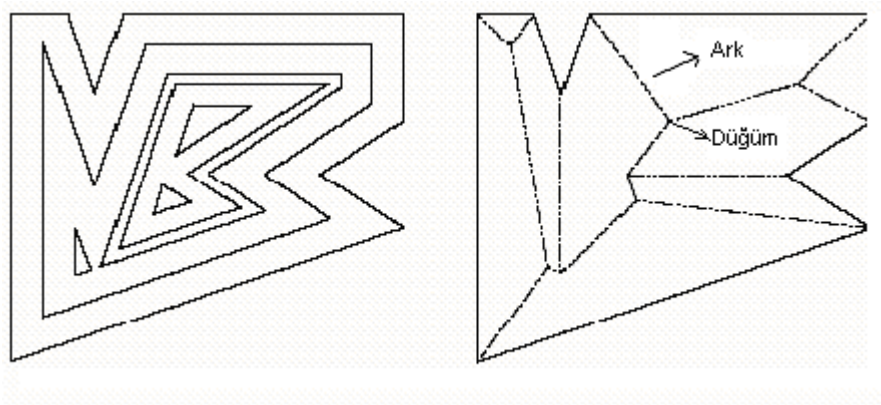
1. Kenar olayı (edge event): Bir kenar ortadan kaybolur ve bu kenarın komşuları bitişik hale gelir.
2. Ayrılma olayı (split event): Bir sert çıkıntılı köşe karşısındaki kenara doğru uzanır ve onu ikiye böler, bu olay aynı zamanda tüm poligonun bölünmesine neden olur. Ayrılan kenarla sert çıkıntılı köşenin iki kenarı arasında yeni bir komşuluk oluşur (Felkel ve Obdrzalek 1998) (Şekil 4-16).



Şekil 4-16 : Kenar olayı (A) ve Ayrılma olayı (B) (Cacciola 2000)

Poligonlar, bu iki olayı da içerecek şekilde sıfır alana sahip oluncaya kadar büzülürler.

Basit iskeleti S(P) ile gösterirsek, S(P), P poligonu içerisinde tanımlı benzersiz bir yapıdır, yani bir poligon içerisinde yalnız bir S(P) tanımlanabilir. P poligonunun her bir kenarı e, "e yüzeyi" denilen bir alanı tarar. Açıortay parçalarına "ark (arc)" ve arkların P'nin köşesi olmayan son noktalarına "düğüm (node)" denir **(**Şekil 4-17**)**.



Şekil 4-17 : Poligonal büzülme ve basit iskelet yapısı (Aichholzer ve ark. 1995)

4.5.1.1.Konveks poligonların iskeletleştirme hesabı

Algoritmada temel veri olarak poligonun aktif köşelerinin dairesel sıralanmış listesi kullanılır. Bu listeye SLAV adı verilir. Poligonun konveks olması durumunda her poligon için yalnız bir tane SLAV tanımlıdır. Konveks olmayan poligonlarda sert çıkıntılı köşeler poligonu böleceğinden dolayı (ayrılma olayı) birden fazla SLAV tanımlanır. SLAV'da bulunan bütün köşeler dairesel sıradaki her iki komşu poligon köşesi ile de ilişkilidir (Felkel ve Obdrzalek 1998).

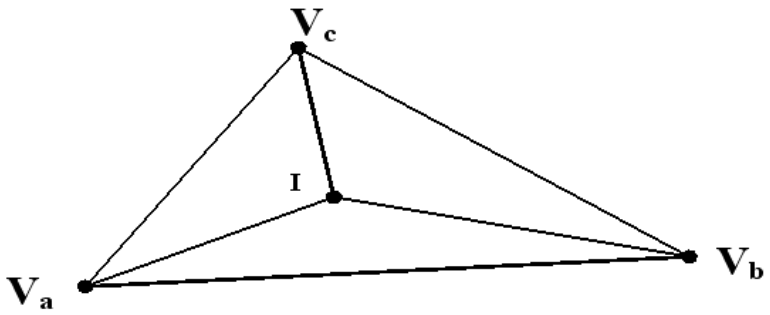
Poligon köşe ve kenarlarının saat ibresinin tersi yönünde sıralandığı düşünülürse algoritma Felkel ve Obdrzalek (1998) tarafından şu şekilde planlanmıştır.

❖ Verilerin algoritmaya hazır hale getirilmesi

- SLAV içindeki V_1 , V_2 , …..V_n köşeleri dairesel sırada komşusu olan her iki köşe noktasıyla ilişkilendirilir.
- SLAV'daki her V_i köşesi için ilgili iki kenara ($e_{i-1} = V_{i-1}\ V_i$ ve $e_i = V_i\ V_{i+1}$) işaret (pointer) konulur ve köşedeki açıortay ışını b_i hesaplanır.
- Her V_i köşesi için b_i'nin komşu açıortaylar ile en yakın kesişmeleri hesaplanır. Eğer kesişme varsa, kesişmeler e_i kenarına uzaklıklarına göre öncelik dizesinde depolanır. Her kesişim noktası I_i için aynı zamanda I_i kesişim noktasını oluşturan iki açıortayın iki orijini V_a ve V_b de depolanır. Bunlar algoritmanın diğer aşamasındaki açıortay hesaplamaları esnasında uygun kenarların tanımlanması için kullanılacaktır **(**Şekil 4-19**)**.

❖ Kesişim noktalarından oluşan öncelik dizesi boş olmamak üzere;

- Öncelik dizesinin başından I kesişim noktası alınır.
- I tarafından tanımlanan V_a ve V_b köşeleri işlem sırasında işaretlenmişse algoritma devam eder. Yani kesişim noktasını oluşturan ayaklar daha önce başka bir kesişim oluşturmuşsa algoritma devam eder, aksi takdirde V_a ve V_b köşeleri arasındaki e kenarı yok edilir, kesişim noktası bu kenarı temsil eder (kenar olayı) **(**Şekil 4-19 ⟋⟍ ile işaretli kenarlar**)**.
- Eğer V_a'dan önceki noktanın öncesi V_b'ye eşitse üçgen yuva oluşuyor demektir bu durumda V_aI, V_bI, V_cI iskelet arkları oluşturulur (Şekil 4-18). Daha sonra algoritma devam eder.



Şekil 4-18 : Üçgen yuva oluşma durumu

- Yuva durumu söz konusu değilse basit iskeletin iki arkı V_aI ve V_bI oluşturulur.
- Aktif düğüm veya köşelerin listesi değiştirilir. (Kenar olayı nedeniyle eski düğüm noktası köşe haline gelebilir.)

➢ I'nın işaret ettiği V_a, V_b köşe veya düğümleri işlem esnasında işaretlenir **(**Şekil 4-19 × ile işaretli noktalar **)**.

➢ I'nın koordinatlarında yeni düğüm V noktası işaretlenir **(**Şekil 4-19 ▪ ile işaretli noktalar**)**.

➢ Bu yeni düğüm V, SLAV'a eklenir. Böylece V, V_a'nın öncesi ve V_b'nin sonrası ile bağlantılı olur ve V_a ve V_b noktaları yok olur.

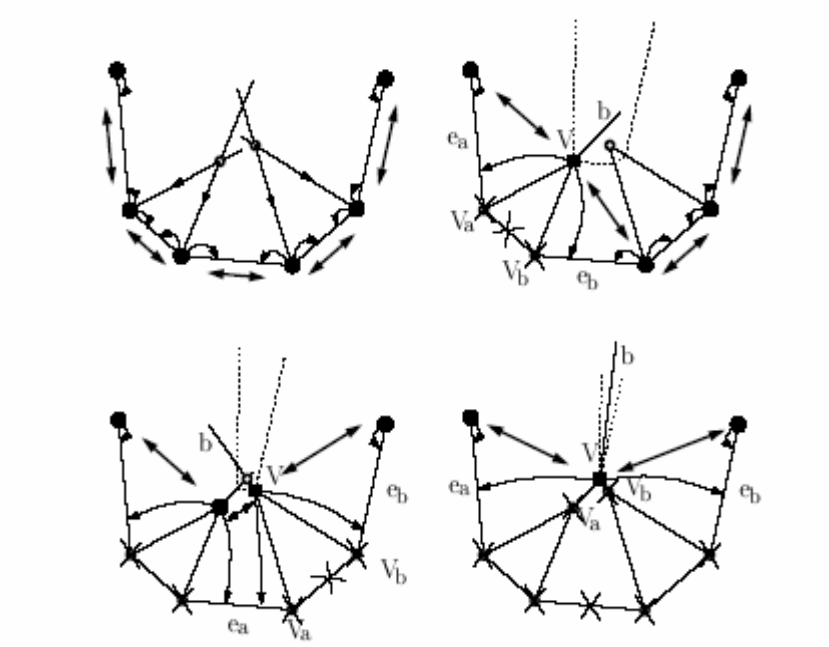
➢ Yeni düğüm V ile uygun kenarlar e_a, e_b arasında ilişki kurulur.

- Yeni düğüm V için aşağıdaki hesaplamalar yapılır;

➢ e_a, e_b doğru parçaları arasındaki yeni açıortay b hesaplanır.

➢ SLAV'daki komşu köşelerden başlayan açıortaylarla bu açıortayın kesişimleri algoritmanın başındaki gibi araştırılır.

➢ En yakın kesişim öncelik dizesine eklenir. Algoritma bu şekilde devam eder.



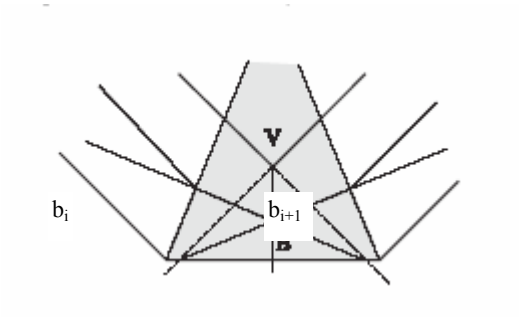
Şekil 4-19 : Konveks poligonlar için algoritmanın akış aşamaları (Felkel ve Obdrzalek 1998)

4.5.1.2.Konveks olmayan poligonların iskeletleştirme hesabı

Konveks olmayan poligonlarda da konveks poligonlarda olduğu gibi yeni kesişim noktası hesaplandıktan sonra öncelik dizesine eklenir. Fakat konveks olmayan poligonlarda ayrılma olayı da görülebileceğinden olayın türü de kaydedilmelidir. Konveks olmayan poligonlarda iskeletleştirme hesabında önemli olan ayrılma olayının oluştuğu B noktasının yerinin belirlenmesidir **(**Şekil 4-20**)**. Yani büzülme esnasında sert çıkıntılı köşenin karşı kenarı kestiği noktanın bulunmasıdır.

Bunun için algoritma içerisinde öncelikle poligonun orijinal tüm kenarları incelenerek olası karşı doğru durumuna düşecekler belirlenmelidir. Maalesef hem herhangi bir kenarın karşı doğru olup olmadığını, hem de karşı doğruysa sert çıkıntılı köşe ile kesişimi olan B noktasının yerini birlikte belirlemek mümkün değildir. Bu yüzden öncelikle poligonun orijinal e_i kenarında bir ayrılma olup olmayacağını incelemek, daha sonra eğer ayrılma varsa aday ayrılma noktası B_i' nin gerçek B olup olmadığını incelemek gerekir. Eğer ayrılma söz konusu ise B_i noktasının, test edilen e_i kenarı ve e_i kenarının iki köşesinden çıkan b_i ve b_{i+1} açıortayları arasında kalan alanda yer alıp almadığı test edilmelidir **(**Şekil 4-20**)**. Eğer bu sağlanıyorsa B_i noktası

için ikinci şart , B_i, V sert çıkıntılı köşedeki açıortay ve V köşesini oluşturan kenarlardan birinin e_i kenarı ile oluşturduğu açının açıortayının kesişiminde olmalıdır **(**Şekil 4-20**)**. Daha sonra bu şartları sağlayan B_i noktaları arasından V'ye en yakını B noktası olarak belirlenir.



Şekil 4-20 : Ayrılma noktası B'nin bulunması (Felkel ve Obdrzalek 1998)

Bu B noktasının belirlenmesiyle poligon da ikiye bölünmüş olur. Poligonun ikiye bölünmesi ise SLAV'ı da ikiye bölmeyi gerektirir. Bunun için SLAV içerisine her biri bir parça için koordinatı B ile aynı iki tane X düğüm noktası eklenir. Her iki nokta da dairesel düzen içerisinde bir önceki ve bir sonraki noktalarla ilişkilendirilir. Algoritma olarak bundan sonraki tüm aşamalar konveks poligonlardakiyle aynıdır.

4.5.2. Çatı yöntemi

Basit iskeletleştirme yönteminin 3 boyutlu versiyonu çatı modelidir. P poligonunu zemin duvarı gibi düşünürsek, bu yöntemi duvarın üzerine sabit eğimli bir çatı oluşturmak gibi hayal edebiliriz. Çatının sırtı ve olukları iskeleti oluşturur.

Çatı yönteminde iskelet P poligonunun köşelerini yaprak gibi düşünürsek bir ağaca benzer ve n tane bağlantılı yüzeye, n-2 tane düğüme ve 2n-3 tane arka sahiptir (Aichholzer ve ark. 1995). Her ark, P'nin iki kenarı e ve (e') nün açıortaylarının bir parçasıdır. l(e) ve l(e') çizgileri iskelette bu kenarlara karşılık gelir. R^3'te her l(e) ışını bir h(e) yarı düzlemi tanımlar. Yine R^3'te açıortay çizgilerinden sadece bir tanesi h(e) $\cap$ h(e') kenarında kesişir ve hesaplamalarda sadece bu açıortay dikkate alınır.

Basit iskelet, arkları P'nin kenarlarındaki açıortaylarla tanımlanan geometrik grafik olarak görülebilir. Arkların bir tarafı bir poligon köşesiyle, diğer tarafı ise iskeletin düğümlerince sınırlanır. Her düğüm 3 açıortayın kesişimidir. Bu özelliklere sahip grafiğe P'nin açıortay grafiği denir **(**Şekil 4-21**)**(Aichholzer ve ark. 1995). Çatı bu grafik yardımıyla tanımlanır.

P poligonu π_0 yatay düzlemi içerisinde bulunur ve P'nin her kenarı 3. boyutta l(e) ile sınırlı olan ve π_0'a göre 45° lik sabit bir eğim açısına sahip olan π(e) yarı düzlemiyle ilişkilendirilirse P üzerinde istenilen çatı oluşturulmuş olur. Burada 45° lik açının özelliği su çizgileri yöntemiyle çatı yöntemi kıyaslanırsa ortaya çıkar. İkizkenar üçgen özelliği dikkate alınırsa, su çizgileri yönteminde kesişim noktasının, düzlemde poligon kenarından olan uzaklığı, çatı yönteminde üçüncü boyutta paralel düzlemle olan uzaklığına eşittir.

G, P'nin düzlem açıortay grafiği olmak üzere G'nin her düğümü u, u'nun tanımlandığı P'nin 3 kenarına teğet bir dairenin merkezidir. u düğüm noktası bu dairenin yarıçapına dik olacak şekilde yukarı kaldırılırsa 3. boyutta λ(u) noktası elde edilir. Yukarı kaldırılma miktarı 45° özelliğinden dolayı düğüm noktasının poligon kenarına olan uzaklığına eşittir. λ(u) çatının bir köşesidir. Eğer u'nun arkları (a,b), (b,c), (c,a) olarak isimlendirilirse, λ(u) da $\lambda(u) \in \pi(a) \cap \pi(b) \cap \pi(c)$ 'dir.

Eppstein ve Erickson (1999) çatı oluşturmada aşağıdaki gibi bir yol izlemişlerdir.

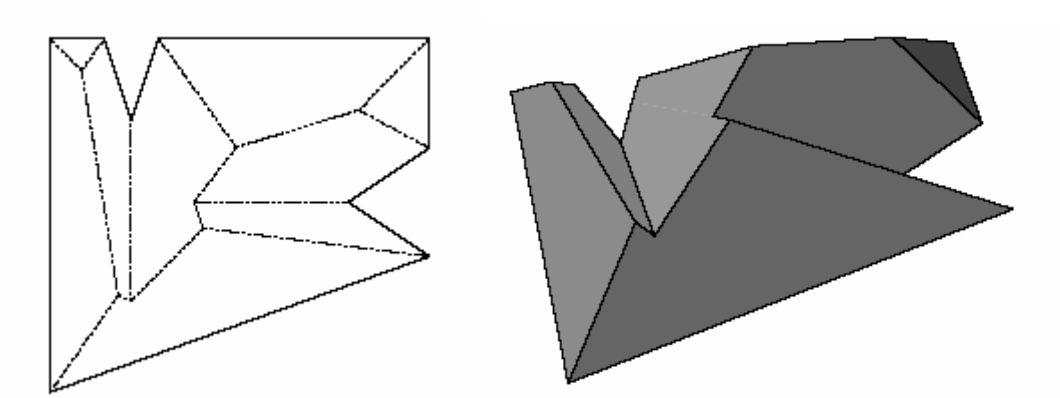
Her poligon kenarı R^3'te sınırsız üçgen tanımlar. Ancak bunlardan yalnız bir tanesi π_0 yatay düzlemiyle 45° lik bir açı yaparken, poligon kenarı olmayan diğer iki kenarının yönü poligon iç bölgesine doğrudur. Her köşe R^3'te bir ışın tanımlar. Eğer büzülme esnasında kenar olayı oluşmuşsa bu, üçgenin tepe noktasında gerçekleşmiştir. Eğer ayrılma olayı oluşmuşsa bu durumda e kenarından çıkan π(e) üçgeni bir ışın veya başka bir üçgenle kesişmiş demektir.

Bu algoritmada her tipteki ilk olayı oluşturan üçgen ve ışınlar bir veri yapısında saklanır. Her olayda veri yapısı güncellenir. Her kenar olayında kaybolan kenarca tanımlanan üçgenle ona komşu iki üçgen silinir ve yeni bitişik kenarlarca tanımlanan

iki üçgen eklenir. Her ayrılma olayında ise bir ışın ve ayrılan kenar ve sert çıkıntılı köşede birleşen iki kenarın oluşturduğu 3 üçgen silinir. Daha sonra ayrılma ile tanımlanan bitişik kenarlarda 2'şer çift olmak üzere 4 üçgen eklenir.

Olabilecek kenar olayları, üçgenlerin en yüksek köşelerinin z değerleri dikkate alınarak oluşturulan bir öncelik dizesiyle belirlenebilir. Doğaldır ki ilk kesişme en yüksek noktalarının z değeri en düşük olan üçgende oluşacaktır. Olay belirlendikten sonra ekleme ve silme işlemleri yapılır ve öncelik dizesi güncellenir.

Olabilecek ayrılma olaylarının belirlenmesi için ise ışın kümesi ve R^3'teki üçgenler kümesi arasındaki z değerleri açısından en alt seviyedeki kesişimi arayan bir veri yapısı kullanılır. Bu iki şekilde yapılabilir. 1) Üçgen kümesi saklanır ve herhangi bir üçgeni kesen ışın aranır 2) Işın kümesi saklanır ve en alt seviyedeki üçgen kesişimi sorgulanır.
Sonuçta oluşan çatı sırt ve oluklarının izdüşümleri iskeleti oluşturmaktadır.



Şekil 4-21 : Açıortay grafik ve çatı modeli (Aichholzer ve ark. 1995)

Çatı yöntemi basit iskeletleştirme yöntemine göre belli avantajlar sağlar. Daha açık bir ifadeyle bu yöntem basit iskeletleştirme yönteminin oldukça karışık olan yapısını biraz daha kolaylaştırmayı amaçlar. İskeleti oluşturan kenar ve ayrılma olaylarının bağlı olduğu öncelikli kesişimler, yalnızca üçgenlerin poligon kenarı olmayan köşelerinin z değerlerine bakılarak belirlenebilmekte ve öncelik dizesi oluşturmak oldukça kolay bir hal almaktadır. Basit iskeletleştirmede iskeleti oluşturan noktalar

belirlenirken, çatı yönteminde çatının sırtı ve olukları doğrudan iskeleti oluşturmaktadır, yani iskelet doğrudan oluşmaktadır.

Basit iskeletleştirme yöntemi genel orta eksen oluşturma yöntemleriyle kıyaslanırsa, basit iskelet sadece doğru parçalarından oluşmaktadır ve hiçbir eğri içermez. Her ikisi de poligonun şeklini yansıtırlar. Basit iskelet poligon kenarlarındaki şekil değişikliklerine daha duyarlıdır. Bir sert çıkıntılı köşe iskelet yapısını tamamen değiştirir. Eğer bu istenmiyorsa sert çıkıntılı köşeler yerel olarak silinmeli böylece köşelerdeki dış açıların en az 90° olması sağlanmalıdır. Basit iskeletin bir dezavantajı da Voronoi diyagramı temelli olmaması ve geometrik objelerin kesişimlerinin iskeleti belirlemesidir. Bu da poligon kenarındaki ufak değişikliklerin iskeleti çok fazla etkilemesine neden olmaktadır. Yine çok girintili çıkıntılı poligonlarda her kenar için üçgen tanımlamak oldukça büyük zaman ve bilgisayarda hafıza kaybına neden olmaktadır. Böyle poligonlarda önce yumuşatma algoritması daha sonra basit iskeletleştirme uygulanması tercih edilebilir.

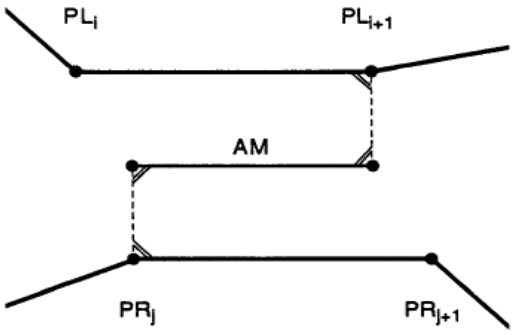
5. ÇİFT ÇİZGİ – TEK ÇİZGİ DÖNÜŞÜM YÖNTEMLERİ

5.1.Yapısallaştırılmış Vektör Formatında Eksen Belirleme

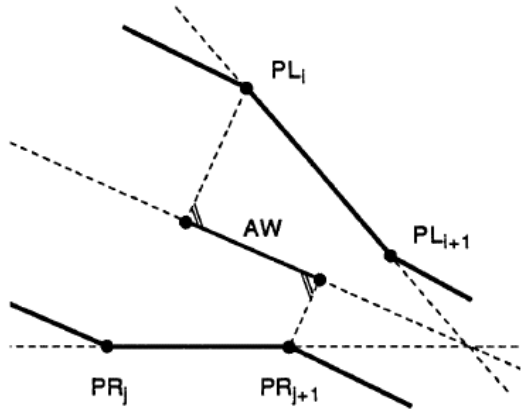
Bu yöntemde yol vb. objeler poligon olarak değil, iki çizgi arasındaki alan olarak görülürler. Objeyi oluşturan karşılıklı kenarlar ve karşılıklı kenarlar üzerinde karşılıklı noktalar belirlenirse eksen oluşturulabileceği düşüncesiyle Menke (1983) tarafından geliştirilmiştir. Bu yöntemin uygulanmasında sayısallaştırmanın kullanıcı tarafından aşağıdaki şartlara uyularak yapılması gerekmektedir (Bildirici 2000, s.38).

- Eşlenik sınır çizgileri aynı yönde sayısallaştırılmalıdır.
- Eşlenik sınır çizgileri veri tabanına sıra ile kaydedilmelidir.
- Her eşlenik sınır çizgi çifti, yalnızca bir tek alan objeye ait olmalıdır.
- İşlenecek verilerde eşlenik olmayan çizgiler olmamalıdır.

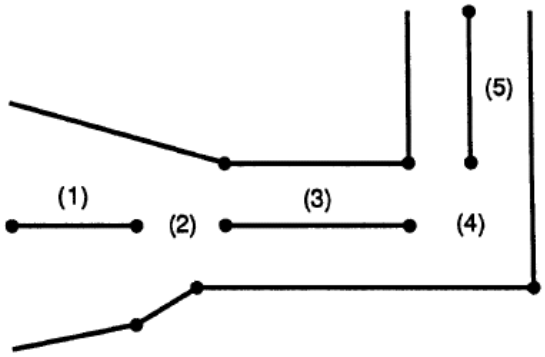
Vektör veriler üzerinde çalışan yöntem böylesi önkoşullar gerektirdiğinden "yapısallaştırılmış vektör" deyimi kullanılmıştır.
Menke noktasal eksen belirleme ve parça parça eksen belirleme olarak iki yaklaşımdan söz etmesine rağmen yalnızca ikinci yöntemi uygulamıştır. Noktasal yöntemde sınır çizgilerinin kırık noktaları analiz edilerek, parça parça eksen belirleme yönteminde ise sınır çizgilerinin karşılıklı parçaları analiz edilerek eksenler belirlenir. Bu başlık altında yalnızca ikinci yöntemden bahsedilecektir.



Şekil 5-1 : Eksenin orta paralel olma durumu (Bildirici 2000)



Şekil 5-2 : Eksenin açıortay olma durumu (Bildirici 2000)



Şekil 5-3 : Boşlukların ara parça elemanlarla (2,4) doldurulması (Bildirici 2000)

Parça parça eksen belirleme yönteminde sınır çizgilerinin karşılıklı parçaları analiz edilerek üç tür eksen parçası belirlenir. Parçalar birbirine paralel ise *orta paralel elemanı* üretilir **(**Şekil 5-1**)**. Parçalar paralel değilse *açıortay elemanı* üretilir **(**Şekil 5-2**)**. Açıortay ve orta paralel elemanları arasında kalan boşluklar *ara parça elemanları* ile doldurulur **(**Şekil 5-3**)**.

Uygulamada çok değişik kaynaklardan gelen veriler kullanıldığından ve bu verilerin yukarıda bahsedilen şartları sağlaması beklenemeyeceğinden, yöntem özellikle küçük alanlarda ve tüm sayısallaştırmayı özel olarak aynı kullanıcının yaptığı durumlarda uygulanabilmektedir.

6. UYGULAMA:ÜÇGENLEME YÖNTEMİYLE EKSEN BELİRLEME

Kartografik genelleştirmenin modellenmesi, kartografik objelerin çok çeşitli olması ve birbirlerinden çok farklı yapıya sahip olmaları nedeniyle oldukça karmaşık bir iştir. Kartografik objelerin modellenmesindeki zorluklar ve ihtiyaçların çeşitliliği sayısal ortamda her ihtiyaca cevap verebilen kartografik genelleştirme yazılımlarının üretilmesini bugüne kadar engellemiştir. Ayrıca kentsel planlamaları düzenli olan orta ve batı Avrupa ülkelerinin verileri temel alınarak geliştirilmiş olan genelleştirme yazılımlarının ülkemizdeki veri yapısı dikkate alındığında istenilen sonuçları vermediği görülmektedir.

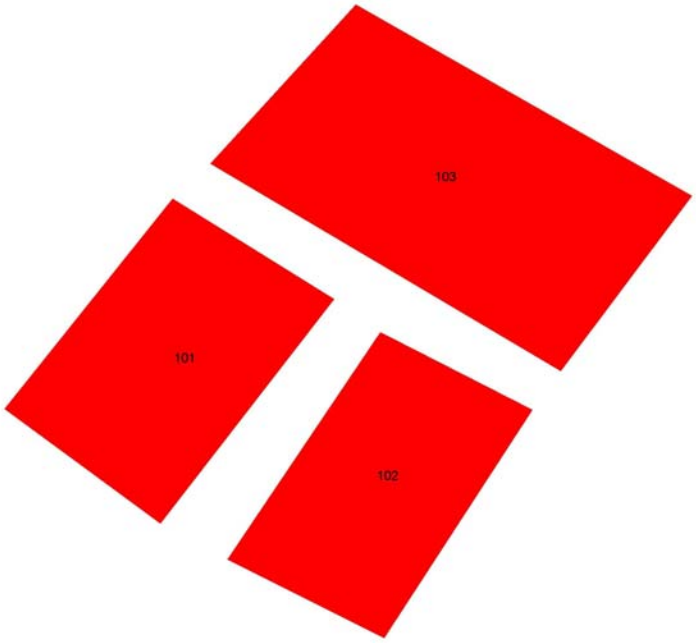
Bu uygulama kapsamında yıllarca ülkemizde ihmal edilen konulardan biri olan alan – çizgi geometrik dönüşümü üzerinde durulmuştur. Çünkü bu dönüşümle ilgili dünya çapında birçok yöntem geliştirilirken, ülkemizde ne yazık ki bu konuyla ilgili hemen hemen hiçbir çalışma yapılmamıştır. Oysa günümüzde herkesin ilgilenmeye çalıştığı CBS alanındaki yazılımların dahi birçoğunda ideal bir genelleştirme ve bunun alt kollarından biri olan alan – çizgi dönüşümü yapılamamaktadır. Bu da kullanıcıların bu ihtiyacı kendi ek yazılımlarıyla gidermelerini zorunlu kılmaktadır.

Bu uygulama kapsamında 4.1 bölümünde anlatılan üçgenleme yöntemini esas alan, ülkemizdeki veri yapısına uygun, ülkemizde yaygın kullanım alanına sahip bir CAD (Computer Aided Design) ve CBS yazılımı olan NETCAD üzerinde çalışan, özellikle yol eksenlerinin belirlenmesinde kullanılabilecek bir alan – çizgi dönüşüm scripti geliştirilmiştir. Uygulama bölümünde ilk olarak Türkiye'deki mevcut veri yapısından bahsedilecek, daha sonra kullanılan NETCAD yazılımı ve onun macro dili NCMACRO hakkında kısaca bilgiler verilecektir. Son bölümde ise geliştirilen yazılım ayrıntılı olarak sonuç ürünleriyle aşama aşama anlatılacaktır.

6.1. Türkiye'deki Mevcut Veri Yapısı

Avrupa ülkelerinde yol alanları kapalı alan olarak tanımlanmış ve yol eksenlerinin belirlenmesi için geliştirilen alan – çizgi dönüşüm yazılımları bu kapalı alanları

kullanacak şekilde tasarlanmıştır. Ülkemizde mevcut büyük ölçekli sayısal harita verilerinde özellikle yol alanları için Avrupa ülkelerinde olduğu gibi herhangi bir tanımlama yapılmamıştır. Fakat kapalı alan olarak tanımlanan adaların arka planları, yol alanları olarak düşünülebilir (Şekil 6-1). Ülkemizdeki bu durum geliştirilen alan – çizgi dönüşüm yazılımının temel düşüncesini etkilemiştir. Bu nedenle ada alanlarını ve ada köşelerini kullanarak yol alanlarının eksenleri belirlenmiştir.



Şekil 6-1 : Türkiye'de büyük ölçekli haritalarda veri yapısı

6.2. NETCAD ve NCMACRO

6.2.1. Netcad

Netcad geliştirilmeye 1987 yılında başlanmış, 19 yılı aşkın süredir Kamu ve Özel sektör ihtiyaçlarını karşılayan çeşitli mesleki çözümleri bünyesinde bulunduran bir yazılımlar bütünüdür. Uluslararası standartlarda tek Ulusal CAD ve CBS yazılımıdır. Netcad üzerinde 30'a yakın temel uygulama geliştirilmiştir. Halihazırda yaklaşık 4000 Kullanıcı ve 13000 lisans çalışmaktadır.

Netcad ortamında sunulan çözümler farklı araçlar kullanılarak genişletilebilir, programlanabilir. Script dilleri (VB ve Java) , C/C++/Delphi/VB/Power Builder gibi

COM/COM+ Mimarisini destekleyen diller bu amaçla kullanılabilir. Bu uygulama kapsamında Netcadin VB Script tabanlı Macro dili kullanılmıştır. Aşağıda Netcadin makro dili ve geliştirilen yazılımda kullanılan özellik ve fonksiyonları hakkında kısaca bilgi verilecektir.

6.2.2. Ncmacro

Netcad Makro Dili; VBScript ve JAVAScript ile Netcad'e ve Netcad objelerine erişim ve kullanma imkanı sağlayan yöntemler bütünüdür. Bu yöntemler ile Netcad üzerinde geliştirilecek bir takım özel uygulamalar ile yapılması zaman alan bir takım işlemler kısa sürede sonuçlandırılabilir, Netcad üzerine yeni modüller eklenebilir.

- **NCSObj:**

Kısaca tanımlamak gerekirse Netcad'in kendisidir. Dosya yükleme, ekleme, tabaka, hat tipi, obje oluşturma (nokta, doğru, çoklu doğru vb.) gibi tüm işlemlerin, ayarlamaların yapıldığı bölümdür. Bu işlemler makro içerisinde;

With Netcad
.
.
End With
bloğu içerisinde tanımlanır.

- **NetCadObj**

Genel Netcad objesidir. Bir obje **NetcadObj** tarafından üretilir, **NCSObj** tarafından şekillendirilir. Yani bir nokta objesi üretilecek ise, noktayı tanımlayan koordinat bilgisi, isim bilgisi, kod bilgisi, tabaka bilgisi **NCSObj** tarafından verilir, **NetCadObj** tarafından bir nokta objesi döndürülür.

- **NCBDialog**

Kullanıcıdan bilgiler almaya yarayan temel bir diyalog kutusudur. Verilen komutlarla diyalog kutusunun boyutunu, iç yerleşimini, etiketleri vb. her şeyi otomatik olarak ayarlamak ve kullanıcıya göstermek mümkündür.

- **NCWorldType**

Netcad Bölge Tipi. Netcad tarafından kullanılan bir bölge objesidir. Bölge sol alt köşe koordinatları ve sağ üst köşe koordinatları bilinen bir dikdörtgenden oluşur.

- **NCCoor**

Koordinat bilgisini tutan objedir.

➢ **NCCoor Komutları**

property x: Bir koordinat objesinin x alanının değerini öğrenmek ve değiştirmek için kullanılır.

property y: Bir koordinat objesinin y alanının değerini öğrenmek ve değiştirmek için kullanılır.

property z: Bir koordinat objesinin z alanının değerini öğrenmek ve değiştirmek için kullanılır.

property flag: Bir koordinat objesinin flag alanının değerini öğrenmek ve değiştirmek için kullanılır.

procedure Assignx(const c1:NCCoor)**:** Bir koordinat objesinin bilgilerini diğerine aktarır.

function GetCopy:NCCoor: Bir koordinat objesini kopyalar ve yeni bir değişkene aktarır.

function SamePoint(const c:NCCoor):Verilen c koordinatı ile kendisi aynımı diye bakar. Aynı ise true farklı ise false döner.

- **NCPline**

Çoklu doğru objelerinin nokta koordinatlarını (NCCoor objeleri) tutan bir yapıdır.

➢ **NcPline komutları**

Function AddCoor: Poligon objesinin noktalarını tutan NCPline objesine yeni nokta koordinatı ekler.

Procedure Clear: Poligon objesinin noktalarını tutan NCPline objesinin noktalarını siler.

Procedure CalcLimits: Obje limitini yeniden hesaplar.

Property name: NCPLine objesinin adını öğrenmek ve değiştirmek için kullanılır.

Property num: NCPLine objesinin nokta sayısını öğrenmek ve değiştirmek için kullanılır.

Property Limits: NCPLine objesinin limitini (NCPLine objesini içine alan en küçük bölge koordinatları) öğrenmek ve değiştirmek için kullanılır.

Property Cor: NCPLine objesinin belirtilen indeksteki nokta koordinat bilgisi erişmek kullanılır.

Property Area: NCPLine objesinin alan değerine ulaşmak için kullanılır. Çoklu doğru saat yönünde oluşturulmuş ise pozitif, ters saat yönünde oluşturulmuş ise negatif alan değeri döner.

Function InPoly: Verilen koordinat alanın içinde ise true, değil ise false döndürür.

Function Operation: Bu komut iki poligonun kesişim, çıkarma ve birleşimlerini bulur. Sonuçlar NCCollection cinsindendir. Çünkü işlem sonucunda birden çok çokludoğru çıkabilir.

Function OnPoly: Bu komut verilen nokta, poligonun çizgileri üzerinde mi diye bakar. Poligonun kapalı mı / açık mı olduğunu kullanıcı verecektir.

Function CenterOfMass: Bu komut poligonun ağırlık merkezini bulur.

Function Perim: Bu komut poligonun çevre uzunluğunu bulur.

Procedure Traverse: Bu komut poligonun yönünü tersine çevirir .

Procedure SetDirection: Bu komut poligonun yönünü değiştirir.

Procedure Assignx: Bir objenin bilgilerini diğerine aktarır.

Function GetCopy: Verilen bir çoklu doğruyu kopyalar ve yeni bir değişkene set eder.

Function GetParalel: Pline objesinin istenen mesafede paralelini alır.

- **NCMath Objesi**

Çizgisel ve noktasal grafik nesnelerle ilgili matematiksel fonksiyonlarını gerçekleştiren objedir.

- **NCMath komutları**

Function Intersection: Verilen a1,a2 ile b1,b2 çizgilerinin kesişim noktasını bulur. Sonucu cx içine doldurur.

Function MidPointOf : Verilen c1,c2 doğrusunun orta noktasını bulur. Sonuç noktayı yaratıp verir.

Function Rotate_Around: Verilen c noktasını orijin noktası etrafında döndürür. Sonuç noktayı yaratıp verir.

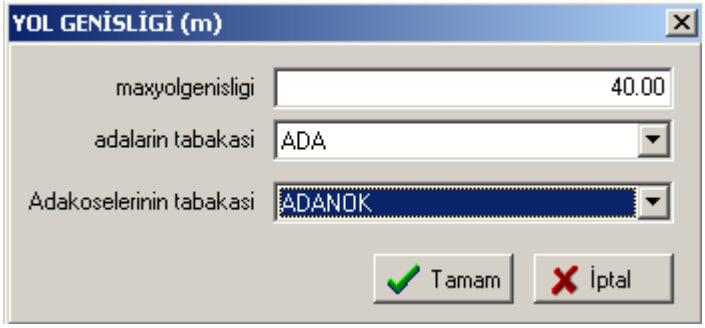
FunctionOnlineSeg: Verilen c noktası, c1,c2 doğrusu üzerinde mi diye araştırır. Çizginin EPSILON kadar tampon bölgesine bakılır. Bu değer kullanıcı tarafından girilmezse Netcad'de varsayılan olarak 0.5 dir.

Function Distance: Verilen c1,c2 noktaları arasındaki mesafeyi verir.

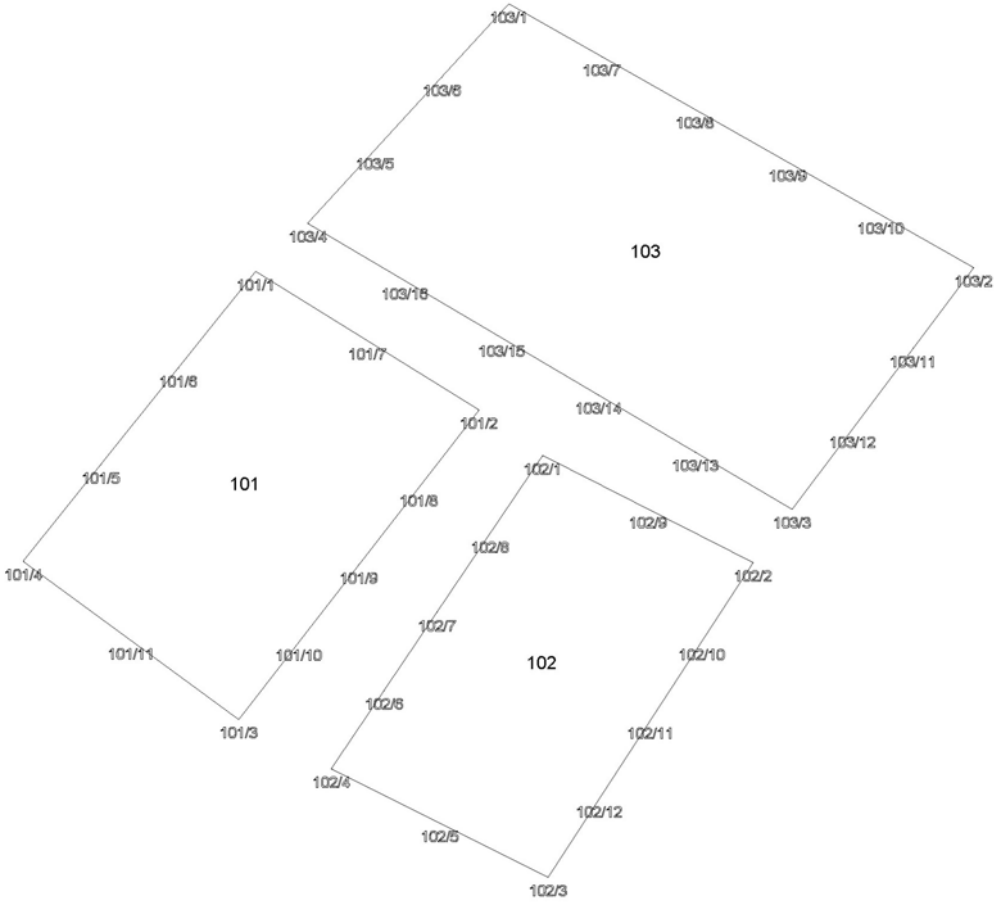
Function Side_Shoot: Verilen mesafe kadar c1,c2 doğrultusunda ilerler, yine verilen mesafe kadar sağa/sola açılarak yeni noktayı bulur.

6.3. Uygulama

Üçgenleme ve eksen oluşturma işlemlerine başlamadan önce ilk olarak adalar üzerinde nokta sıklaştırması yapılarak, arzu edilen Delaunay üçgenlerini oluşturmak için yeteri sıklıkta noktalar üretilmiştir. Bu amaçla kullanıcıdan maksimum yol genişliği ile birlikte ada tabakası ve adaların köşe nokta tabakası alınmıştır (Şekil 6-2). Burada kullanılan verinin belirli standartlara sahip olması gerekmektedir. Adalar aynı tabakada olmalı ve ada köşe noktaları farklı bir tabakada tutulmuş olmalıdır. Nokta sıklaştırması yapılırken kullanıcıdan alınan maksimum yol genişliği (d) temel kriter olarak kabul edilmiştir. Ada üzerindeki noktalar arasındaki uzaklık 1.5*d değerini geçmeyecek şekilde nokta sıklaştırması yapılmıştır (Şekil 6-3).



Şekil 6-2 : Nokta sıklaştırması için kullanıcıdan gerekli değerlerin alınması

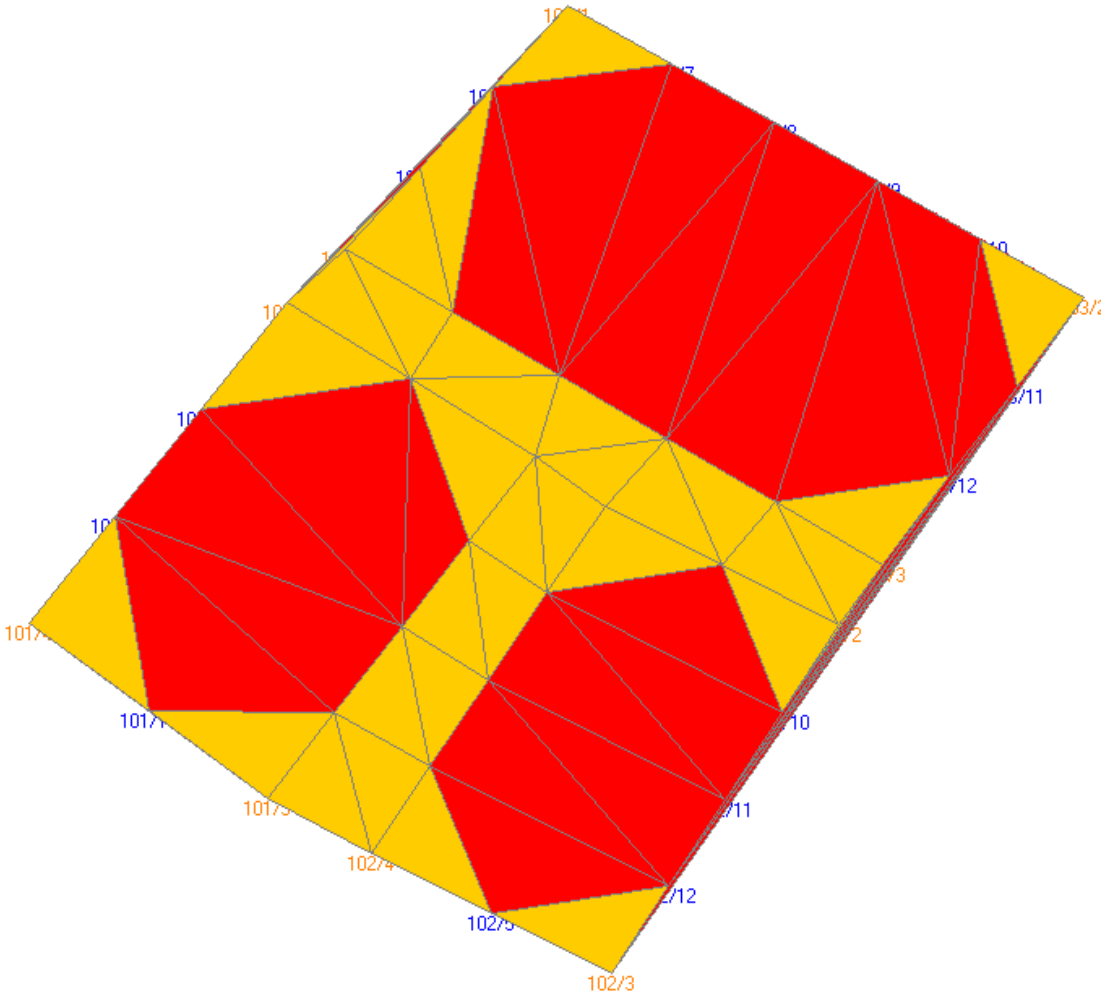


Şekil 6-3 : Adalar üzerinde nokta sıklaştırması yapılması

Yeni oluşturulan tüm noktalar geçici noktalar olduklarından farklı bir tabakaya alınmış ve eksen oluşumu tamamlandıktan sonra kolayca silinmeleri sağlanmıştır. Sıklaştırma noktalarını üzerinde bulundukları adayla ilişkilendirmek amacıyla tüm bu noktalara ada numaralarıyla bağlantılı numaralar verilmiştir (101/8, 102/6 vb.) (Şekil 6-3). Bu numaralandırmada 101, 102 ... değerleri ada numaralarını, 8,6... vb. değerler ise noktayı temsil etmektedir. (Ekteki CD'de noktaatan.nvb macrosu nokta sıklaştırmasını yapmaktadır.)

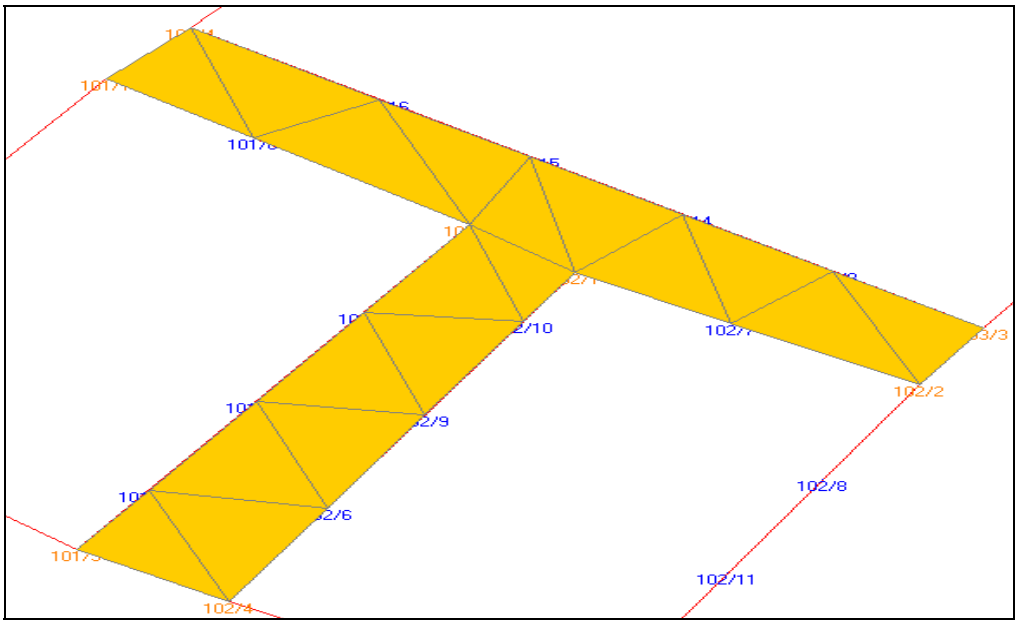
Nokta sıklaştırması tamamlandıktan sonra tüm noktaları içerecek şekilde Netcad yazılımı ile üçgenleme yapılmıştır. Burada üçgenleme işlemi de macro yardımıyla yapılmak istenmiş fakat üçgenleme işlemine ait objeler NCMACRO içerisinde şu ana kadar yer almadığından bu işlem manuel olarak yapılmıştır. Netcad yetkilileri ile yapılan görüşmeler sonucu üçgenlemenin kısa bir süre içerisinde NCMACRO içerisine katılacağı bilgisi alınmıştır. Bu sayede üçgenleme işleminin de macro yardımıyla yapılması planlanmaktadır. Tüm noktalarda üçgenleme yapıldığında

bozuk üçgenler ve normal üçgenlerden oluşan iki ayrı tabakada üçgenler oluşmuştur (Şekil 6-4).



Şekil 6-4 : Nokta sıklaştırması yapıldıktan sonra üçgenlerin oluşturulması

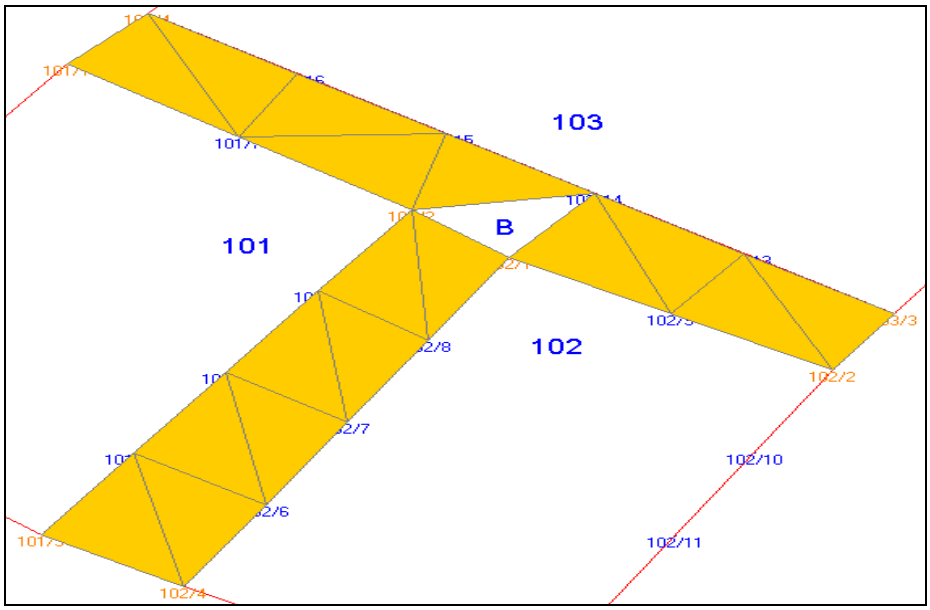
Üçgenleme yapıldıktan sonra eksenlerin oluşturulacağı yol alanları dışında kalan üçgenler silinmiştir. Bu aşamada farklı kavşak bölgelerinde yol alanları içerisinde de bozuk üçgenler oluşabileceği düşünülerek bozuk üçgen - normal üçgen ayrımı yapılmamıştır. Bu amaçla ada içerisine düşen ve alanının en büyük kenarına oranı çok küçük olan tüm üçgenler macro yardımıyla silinmiştir. Gerekli silme işlemleri yapıldıktan sonra hedeflenen, eksen oluşturmada kullanılabilecek, yol alanlarını kapsayan üçgenler elde edilmiştir (Şekil 6-5). Geri kalan bu üçgenler için bundan sonra "yol üçgenleri" ifadesi kullanılacaktır.



Şekil 6-5 : Eksen oluşturmada kullanılacak yol üçgenleri

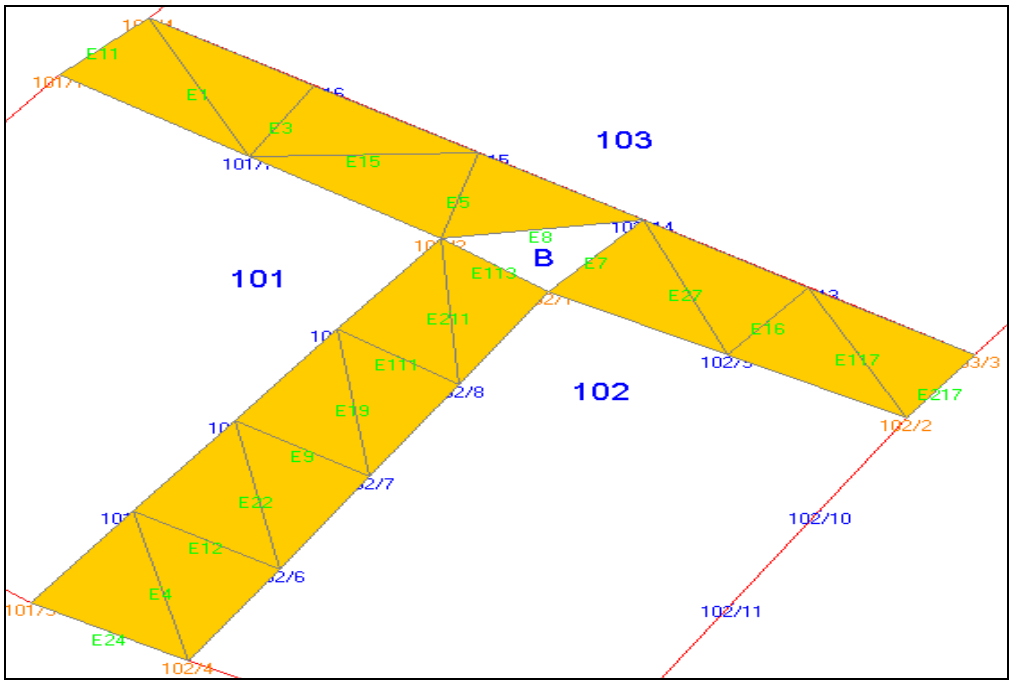
Yol üçgenlerinin oluşturulmasından sonra farklı eksenlerin birleştiği kavşak üçgenler, 3 noktalarının da farklı adaya ait olmalarından yararlanarak, eksenlerin oluşturulmasında dikkate alınmak üzere farklı bir tabakaya alınmıştır (Şekil 6-6, B üçgeni). Bu üçgen türü Şekil 6-6'daki gibi kavşak bölgelerinde bir tane olabileceği gibi, farklı kavşaklarda birden fazla da oluşabilir. Böyle durumlarda da tüm kavsak üçgenleri aynı tabakada tutulur.

Kavşak üçgenlerinin belirlenmesinden sonra eksen noktalarının belirlenmesi işlemine geçilmiştir. Bu amaçla bu tez kapsamında 4.1 başlığında anlatılan alan – çizgi dönüşüm yöntemlerinden üçgenleme yönteminde en iyi sonucu veren "orta nokta çizgi" yaklaşımından yararlanılmıştır. Hatırlanacağı üzere bu yaklaşımda, iki köşesi farklı iki ada üzerinde olan üçgen kenarlarının orta noktaları ekseni oluşturmaktaydı.



Şekil 6-6 : Kavşak bölgelerindeki üçgenlerin farklı tabakaya alınması

Bu özellikten yararlanılarak şartı sağlayan tüm üçgen kenarlarında kenar orta noktaları bulunarak eksen noktaları oluşturulmuştur. Burada faklı iki komşu üçgenin ortak kenarlarında birbirine çakışık iki eksen noktası oluşacağından bu fazla noktalar koordinatlarının aynı olmasından yararlanılarak silinmiştir. Tüm bu işlemlerden sonra amaca uygun eksen noktaları elde edilmiştir (Şekil 6-7 yeşil renkli noktalar).

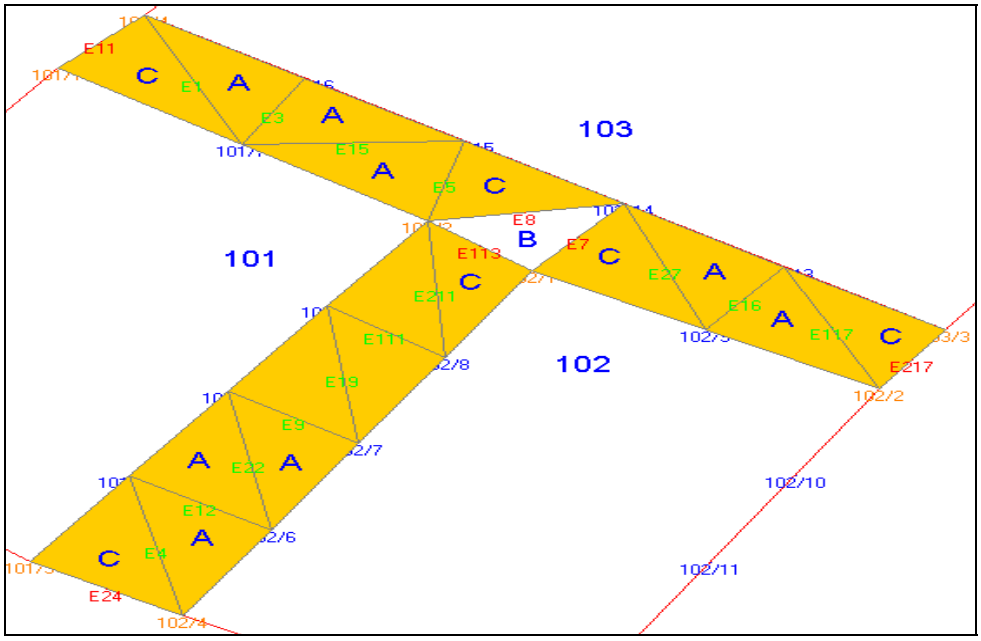


Şekil 6-7 : Eksen noktalarının oluşturulması

Eksen noktalarının belirlenmesinin ardından her bir yol ekseninin başlayacağı ve biteceği noktaların belirlenmesi gerekmektedir. Bu amaçla yol üçgenleri A, B, C tip üçgenler olmak üzere üçe ayrılmıştır (Şekil 6-8).

- A tipi üçgenler : İki tane komşu yol üçgenine sahip üçgenlerdir.
- B tipi üçgenler : Üç köşesi farklı adalarda olan kavşak üçgenlerdir.
- C tipi üçgenler : Yalnızca bir tane yol üçgenine komşu olan veya komşularından bir tanesi kavşak üçgen olan üçgenlerdir.

Eksen başlangıç ve bitiş noktaları, C tipi üçgen üzerinde olup A tipi üçgenler üzerinde olmayan noktalardır (Şekil 6-8 kırmızı renkli noktalar). Bu noktalar diğer eksen noktalarından ayrılmak amacıyla farklı bir tabakaya alınmıştır.

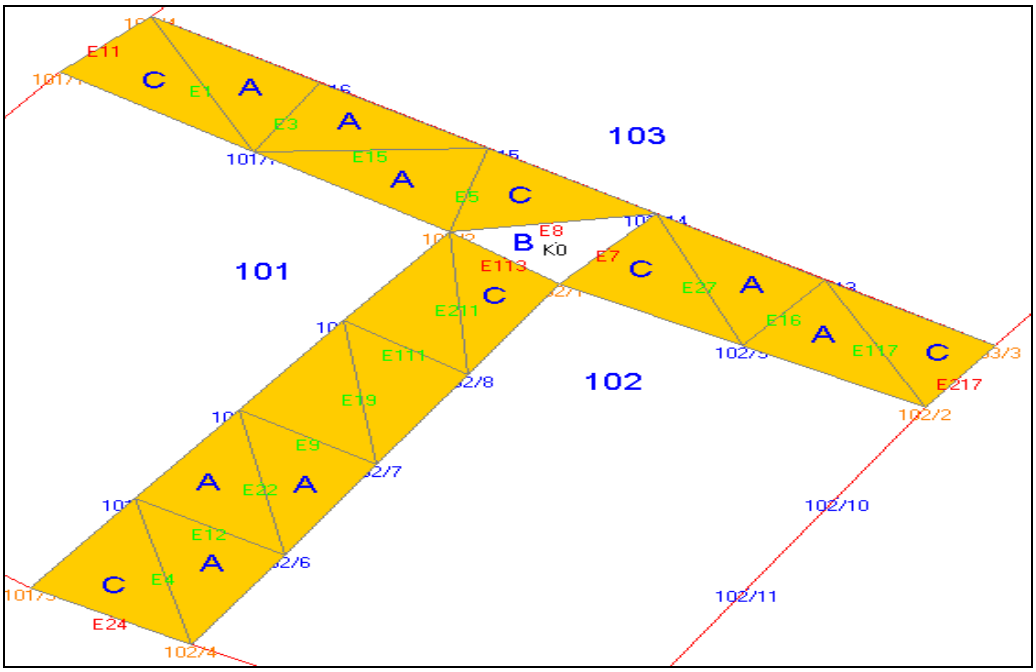


Şekil 6-8 : Eksen başlangıç ve bitiş noktalarının belirlenmesi

Eksen başlangıç ve bitiş noktalarının belirlenmesinden sonra kesişim bölgelerindeki (kavşaklar) eksen kesişim noktaları belirlenmiştir (Şekil 6-9). Bu amaçla eksen başlangıç veya bitiş noktalarının etrafında (2*d) yarıçaplı bir bölge oluşturularak bu bölge içerisinde bir başka eksen başlangıç veya bitiş noktasının olup olmadığı araştırılmıştır. Eğer bölge içerisinde eksen başlangıç veya bitiş noktası varsa, bu noktaların ortalama koordinatlarında, kavşak noktaları belirlenmiştir (Şekil 6-9, K0

noktası). Burada, kavşak üçgenin ağırlık merkezinin kavşak noktası olarak alınması, birden fazla kavşak üçgenin aynı bölgede olması durumunda iyi sonuç vermeyeceği gerekçesiyle tercih edilmemiştir.

Eksen başlangıç ve bitiş noktalarının ve gerekli bölgelerde kavşak noktalarının belirlenmesinin ardından eksenin oluşturulması aşamasına geçilmiştir. Eksen oluşturma algoritması aşağıda maddeler halinde verilecektir:



Şekil 6-9 : Kavşak noktalarının belirlenmesi

❖ Algoritma herhangi bir C tipi üçgen üzerindeki eksen başlangıç noktasına gider. Bu nokta etrafında (1.5*d) yarıçaplı bir bölge oluşturur. Bu bölge içine düşen kavşak noktası olup olmadığını araştırır. Kavşak noktası var ise eksen kavşak noktasından başlatılır ve eksenin ikinci noktası eksen başlangıç noktasıdır. Yok ise eksen bu başlangıç noktasından başlatılır.

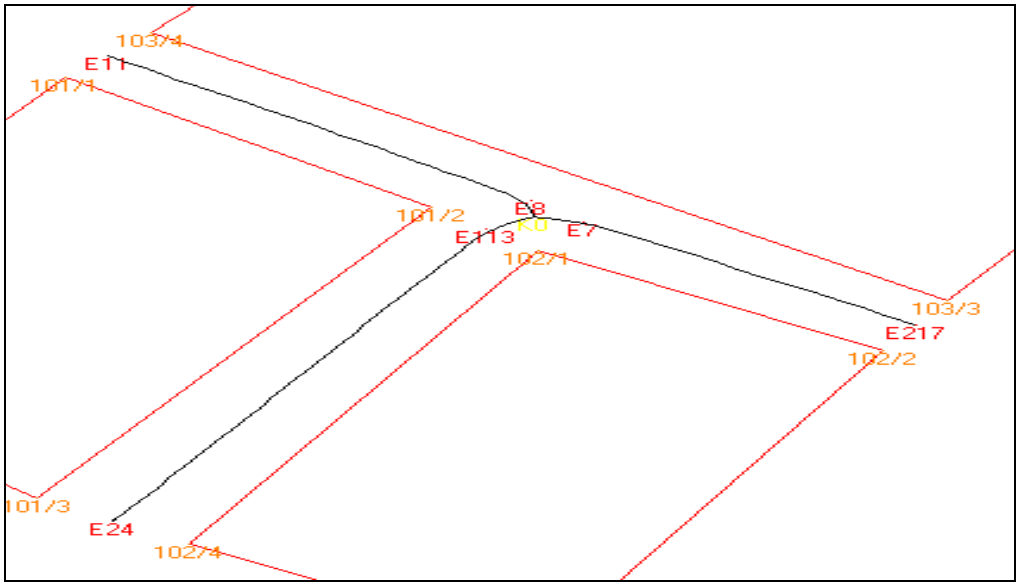
❖ Yine eksen başlangıç noktası etrafında d yarıçaplı bir bölge oluşturulur. Bu bölge içerisine düşen, eksen noktası tabakasında ve eksen başlangıç noktası ile aynı üçgen üzerinde olan nokta olup olmadığı araştırılır. Eksen, bulunan eksen noktasından

geçirilir ve kullanılan eksen başlangıç noktası, tekrar kullanılmaması için farklı bir tabakaya alınır.

❖ Eksenin gelmiş olduğu eksen noktası etrafında, yine d yarıçaplı bir bölge oluşturulur. Bu bölge içerisine düşen ve eksen noktasıyla aynı üçgen üzerinde olan ikinci eksen noktası tespit edilir ve bu nokta eksene dahil edilir. Tekrar kullanılmaması için ilk eksen noktasının tabakası değiştirilir. Algoritma bu şekilde, eksen bitiş noktasına ulaşıncaya kadar devam eder.

❖ Eksen bitiş noktasına ulaşıldığında yine (1.5*d) yarıçaplı bir bölge oluşturulup, bu bölge içerisine düşen kavşak noktası olup olmadığı araştırılır. Var ise eksen bu kavşak noktasıyla bitirilir. Eğer yok ise eksenin son noktası eksen bitiş noktasıdır. Eksen bitiş noktasının tabakası tekrar kullanılmaması için değiştirilir. Böylece ilk eksen oluşturulmuş olur.

❖ İlk eksen oluşturulduktan sonra algoritma diğer bir eksen başlangıç noktasına gider ve aynı işlemler kullanılmayan eksen başlangıç ve bitiş noktası kalmayıncaya kadar devam eder. Bütün eksen başlangıç ve bitiş noktaları kullanıldığında arzu edilen eksenler üretilmiş olur. Daha sonra algoritma içerisinde kullanılan geçici sıklaştırma noktaları ve geçici üçgenler silinerek istenilen sonuç elde edilmiş olur (Şekil 6-10) (Ekteki CD ‘de eksenyapan.nvb dosyası bu yazılımı içermektedir).



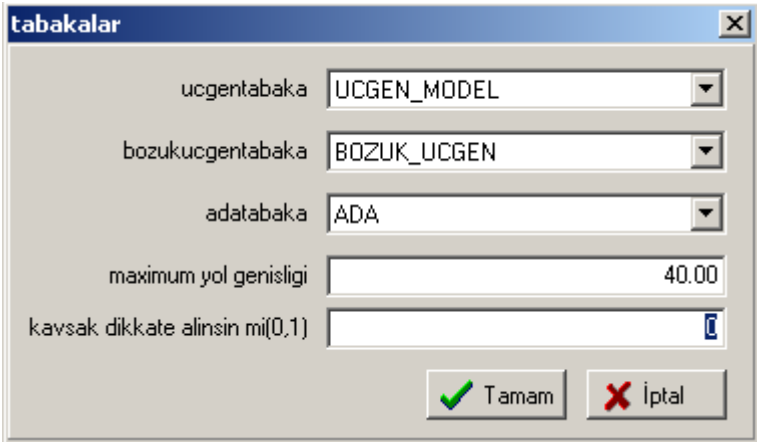
Şekil 6-10 : Eksenlerin oluşması

6.4. Sonuç

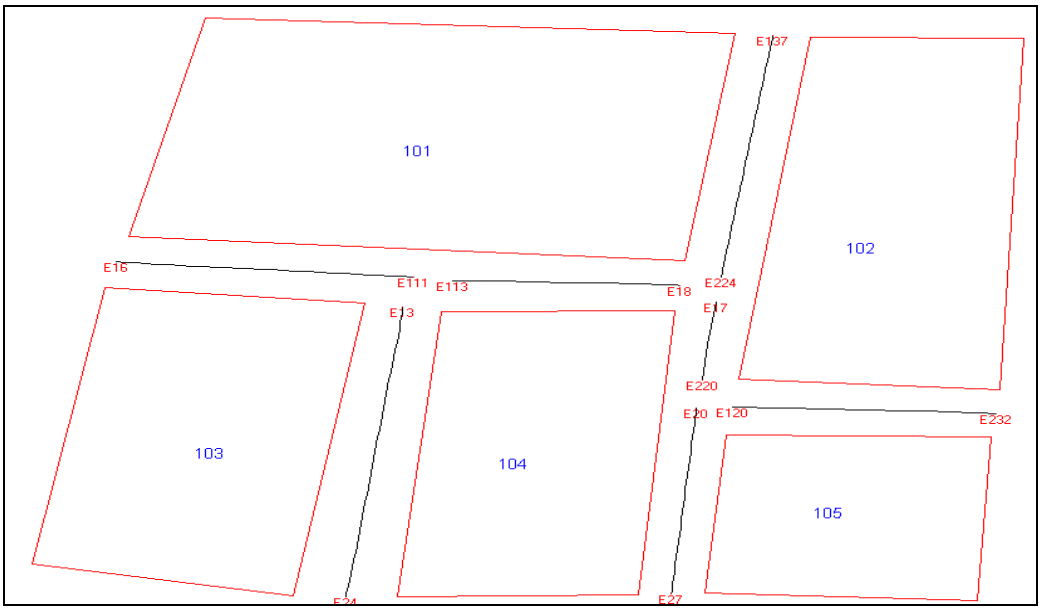
Çalışmanın uygulama kısmında ülkemizdeki veri yapısına uygun, üçgenleme yöntemini kullanarak alan – çizgi dönüşümü yapan bir yazılım yapılmaya çalışılmıştır. Elde edilen sonuçlar dikkate alındığında ise bu amaca büyük oranda ulaşıldığı sonucuna varılmaktadır.

Yapılan çalışma, alan – çizgi dönüşümünü otomatize etme amaçlı, ülkemizdeki ilk çalışma olma özelliğine sahiptir. Özellikle yol eksenlerinin belirlenmesinin CBS yazılımları için vazgeçilmez bir gereksinim olmasına rağmen, ülkemizdeki verilerde yollar için herhangi bir tanımlama yapılmamış olduğu düşünülürse, yapılan uygulamanın bu alandaki bir eksikliği gidermek için önemli bir adım olduğu düşünülebilir.

Ancak genelleştirmenin doğası gereği modellemesi çok zor bir konu olması, yapılan bu yazılımın da bazı özel durumlarda istenilen sonucu vermesini engellemiştir. Özellikle birbirine çok yakın kavşak bölgelerinde oluşan eksenler, bazen istenilen doğrulukta olmamaktadır. Bu gibi durumlarda yazılımın sıkıntı oluşturmaması için yazılım iki seçenekli olarak hazırlanmıştır (Şekil 6-11). Kullanıcı isterse kavşakların dikkate alınmamasını isteyerek (değeri 0 girerek) kavşaklar dikkate alınmadan eksenlerin oluşmasını sağlayabilir (Şekil 6-12).



Şekil 6-11 : Yazılım başlangıç ekranı



Şekil 6-12 : Kavşaklar dikkate alınmadan eksenlerin oluşturulması

Yazılımın kullanılabilmesi için kullanılan verilerin belli standartlara sahip olması gerekmektedir :

- Adalar kapalı alan olarak tanımlanmış olmalıdır.
- Tüm adalar aynı tabakada olmalıdır.
- Ada köşe noktaları farklı bir tabakada tanımlanmış olmalıdır.

Ne yazık ki ülkemizdeki sayısal veriler dikkate alındığında verilerin birçoğunun herhangi bir standarda sahip olmadığı görülmektedir. Tabaka yapılarının operatörlerin kendi isteklerince oluşturulması, bu verilerin CBS amaçlı kullanımlarını ve istenilen analizlerin yapılmasını imkansız hale getirmektedir. Yapılan uygulamanın en azından ilçe bazında denemesi yapılmak istenmiş, fakat yukarıda bahsedilen verilerdeki standartsızlık nedeniyle bu mümkün olmamıştır. Bu kapsamda 15 Temmuz 2005 tarihli "Büyük Ölçekli Harita ve Harita Bilgileri Üretim Yönetmeliği" getirdiği standartlar açısından bir fırsat olarak görülmektedir. Yazılımın farklı kavşak türlerindeki sonuçları EK A, EK B, EK C' de görülmektedir. Yazılımın sıkıntı yaşadığı durumlarla ilgili çalışmalar halen devam etmektedir.

7. SONUÇ VE ÖNERİLER

Son yıllarda bilgisayar teknolojisindeki gelişmeler birçok sektörde olduğu gibi kartografya alanında da önemli sonuçlar doğurmuştur. Bilgisayar teknolojisinin gelişmesi sayısal harita kavramını ortaya çıkarmış ve klasik haritaların yerini sayısal haritalar almıştır. Sayısal haritaların yaygınlaşması, önceleri hayal edilemeyen birçok şeyin olabileceğine inandırmış ve bunun sonucu coğrafi bilgi sistemi (CBS) ortaya çıkmıştır. Günümüzde gelinen bu noktada klasik haritaların kullanılmaz olmasının yanı sıra sadece geometrik objelerin bilgisayar ortamında gösterilmesi olarak tanımlayabileceğimiz sayısal haritalar da ihtiyaca cevap veremez hale gelmiştir. Artık üretilmesi gereken haritalar CBS analizlerine imkan verecek sayısal haritalardır.

Yukarıda anlatılan harita kavramındaki bu değişiklikler, yıllarca kartografların birikim ve becerileriyle tanımlanan kartografik genelleştirme kavramını da etkilemiştir. Belirli kuralları yoktur denilen genelleştirme, artık sayısal ortamda belirli standartları taşıyacak şekilde modellenmesi gereken bir kavram halini almıştır. Sayısal ortamda tanımlanması çok güç olan kartografik objelerin genelleştirilmesini modellemek de oldukça zordur. Tek bir modelle tüm kartografik objeleri ihtiyaca cevap verecek şekilde genelleştirmek ise halen ulaşılamamış bir hedeftir. Bu nedenle farklı tipteki kartografik objelerin genelleştirilmesi farklı metotlarla yapılmaktadır. Bu çalışma kapsamında incelenen alan – çizgi geometrik dönüşümü de bu ihtiyaçtan ortaya çıkmıştır. Yol, akarsu vb. kartografik objeler en çok 1:5000 ölçeğine kadar alan olarak gösterilebildiğinden, daha küçük ölçeklerde bu alan objeyi temsil eden çizgi obje olarak tanımlanır ve genelleştirme bu çizgi objeye uygulanır. Bu nedenle sayısal kartografik genelleştirme için de alan – çizgi geometrik dönüşümü vazgeçilmez bir gereksinimdir.

Bu çalışma kapsamında yıllarca ülkemizde ihmal edilen alan – çizgi geometrik dönüşümü üzerinde durulmuştur. Çünkü bu dönüşümle ilgili dünya çapında birçok yöntem geliştirilirken, ülkemizde ne yazık ki bu konuyla ilgili hemen hemen hiçbir çalışma yapılmamıştır. Ülkemizde çoğunlukla operatörlerin manuel olarak elle eksen

belirlemesi gibi bir yaklaşım uygulanmaktadır. Bu ise hem çok büyük zaman kaybına yol açmakta, hem de operatörlerin sübjektif kararları eksenlerin oluşmasında etkili olmaktadır.

Günümüzde herkesin ilgilenmeye çalıştığı CBS alanındaki yazılımların dahi birçoğunda ideal bir genelleştirme ve bunun alt kollarından biri olan alan – çizgi dönüşümü yapılamamaktadır. Bu da kullanıcıların bu ihtiyacı kendi ek yazılımlarıyla gidermelerini zorunlu kılmaktadır.

Alan – çizgi geometrik dönüşümü için değişik kartografik objeler için farklı ihtiyaçlara cevap verebilecek birçok yöntem geliştirilmiştir. Bu çalışma kapsamında bu yöntemlerden yaygın olarak kullanılan üçgenleme, su çizgileri, en büyük daire, inceltme, çatı ve çift çizgi tek çizgi dönüşüm yöntemleri incelenmiş, bu yöntemler birbirleriyle karşılaştırılmış ve üstün ve zayıf yönleri ortaya konmuştur. Yapılan çalışma sonucu bu yöntemlerle ilgili aşağıdaki sonuçlara ulaşılmıştır.

- Üçgenleme yöntemi, çok fazla girintili çıkıntılı olmayan yol vb. kartografik objelerin eksenlerinin belirlenmesinde kullanılabilir. Çok girintili çıkıntılı kartografik objelerde ideal Delaunay üçgenlemesi yapılamadığından iyi sonuç vermez.
- Su çizgileri yöntemi, girintisi çıkıntısı çok olan akarsu vb. kartografik objelerde iyi sonuç verir. Matematiksel olarak karmaşıktır ve üçgenleme yöntemine göre daha fazla bilgisayar belleği gerektirir.
- En büyük daire yöntemi, hem vektör hem raster formatta uygulanabilir. En büyük dairenin oluşturulmasındaki zorluklar ve CBS analizleri için olmazsa olmaz süreklilik şartını sağlamamasından dolayı çok fazla tercih edilmez.
- İnceltme yöntemi, raster formatta uygulanır. Vektör formatta karşılaşılan birçok sayısallaştırma hatasını yok etmesi nedeniyle tercih edilen bir yöntemdir. Fakat elde edilen eksen matematiksel olarak tam doğru bir eksen değildir ve büyük alanlarda uygulanması için çok güçlü bilgisayar donanımı ve belleği gerektirir. Elde edilen eksenin vektöre tekrar dönüştürülme zorunluluğu yöntemi pratik olmaktan çıkarmaktadır.

- Çatı yöntemi, vektör formatta uygulanmaktadır. Üçgenleme ve su çizgileri yönteminin birleşimi olarak düşünülebilir. Fakat farklı kartografik objeler için üretilen eksenlerin birbirine benzemesi yöntemin esnekliğini yok etmekte ayrıca girintili çıkıntılı objelerde oluşan eksen istenmeyen parçalar içermektedir. Bu nedenlerle bu yöntem çok eleştirilmektedir.
- Çift çizgi tek çizgi dönüşüm yöntemleri ise ancak veriler çok özel şartlarda sayısallaştırılırsa uygulanabileceğinden, farklı kullanıcılardan gelen verilerin de bu standartları sağlaması mümkün olmadığından çok uygulanabilir olarak görülmemektedir.

Bu çalışmanın uygulama kısmında ise Türkiye şartlarında alan – çizgi dönüşümünde en uygulanabilir olarak görülen üçgenleme yöntemi temelli bir script yazılmıştır. Çalışma sonucunda istenilen sonuçlar büyük oranda elde edilmiştir. Yazılımın eksiklikleri ise giderilmeye devam edilecektir. Ülkemizde de benzer yazılımların artması ve tüm sayısal verilerin arzu edilen CBS analizlerine imkan verecek yapıda olması için kullanılan verilerin belli bir standarda sahip olması gerekmektedir. Bu kapsamda tüm sayısal harita üreten kişi ve kurumların 15 Temmuz 2005 tarihli "Büyük Ölçekli Harita ve Harita Bilgileri Üretim Yönetmeliği" esaslarına göre sayısal harita üretmeleri vazgeçilmez bir gereksinim olarak değerlendirilmektedir. Yapılan yazılım yakın gelecekte daha kapsamlı verilerde test edilecektir.

KAYNAKLAR

Aichholzer, O., Aurenhammer, F., Alberts, D., Gartner, B., 1995. A Novel Type of Skeleton for Polygons. Journal of Universal Computer Science,v.1,n.12,s.752-761.

Autodesk, 1999. AutoCad Map2000 User's Guide, Autodesk Inc.

Bildirici, İ.Ö., 2000. 1:1000-1:25000 Ölçek Aralığında Bina ve Yol Objelerinin Sayısal Ortamda Kartografik Genelleştirmesi. Doktora Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

Bildirici, İ.Ö., 2003. Coğrafi Bilgi Sistemlerinde Ağ Topolojisi ve Genelleştirme. 9. Harita Kurultayı, HKMO, Ankara, 31 Mart-4 Nisan, s.337-346.

Bildirici, İ.Ö., Uçar, D., 2003. Sayısal Kartografyada Genelleştirme Yaklaşımları. Yıldız Teknik Üniversitesi Dergisi 2003-2, İstanbul.

Cacciola, F., 2000. A CGAL implementation of the Straight Skeleton of a Simple 2D Polygon with Holes. Yayınlanmamış.

Christensen, A.H., 1996. Street Centerlines by A Fully Automated Medial – Axis Transformations. In Proceedings of GIS / LIS'96. Denver, Kasım 19-21, s.107-116.

Eppstein, D., Erickson, J., 1999. Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions. Discrete & Computational Geometry, n.22,s.569-592, Springer – Verlag New York Inc.

Felkel, P., Obdrzalek, S., 1998. Straight Skeleton Implementation. Proceedings of Spring Conference on Computer Graphics, Budmerice, Slovakia, s.210-218.

Lam, L., Lee, S.W., Suen, C.Y., Thinning Methodologies – A Comprehensive Survey. IEEE Transactions On Pattern Analysis And Machine Intelligence, v.14,n.9,s.869-885.

McAllister, M.ve Snoeyink, J., 2000. Medial Axis Generalization of River Networks. Cartography and Geographic Information Science, v.27,n.2,s.129-138.

Menke, K., 1983. Zur rechnergestützten Generalisierung der Verkehrswege-und Gewassernetzes, insbesondere für den Maβstab 1:25000. Doktora Tezi, WissArbUH Nr.119, Hannover.

Schürer, D., 2002. Ableitung von digitalen Landschaftsmodellen mit geringerem Strukturierungs-grad durch Modellgeneralisierung. Doktora Tezi, Schriftreihe des Institutes für Kartographie und Topographie der Rheinischen Friedrich-Wilhelms-Universität, Bonn.

Sonka, M., Hlavac, V., Boyle, R., 1999. Image Processing, Analysis, and Machine Vision. International Thomson Publishing Inc., London.

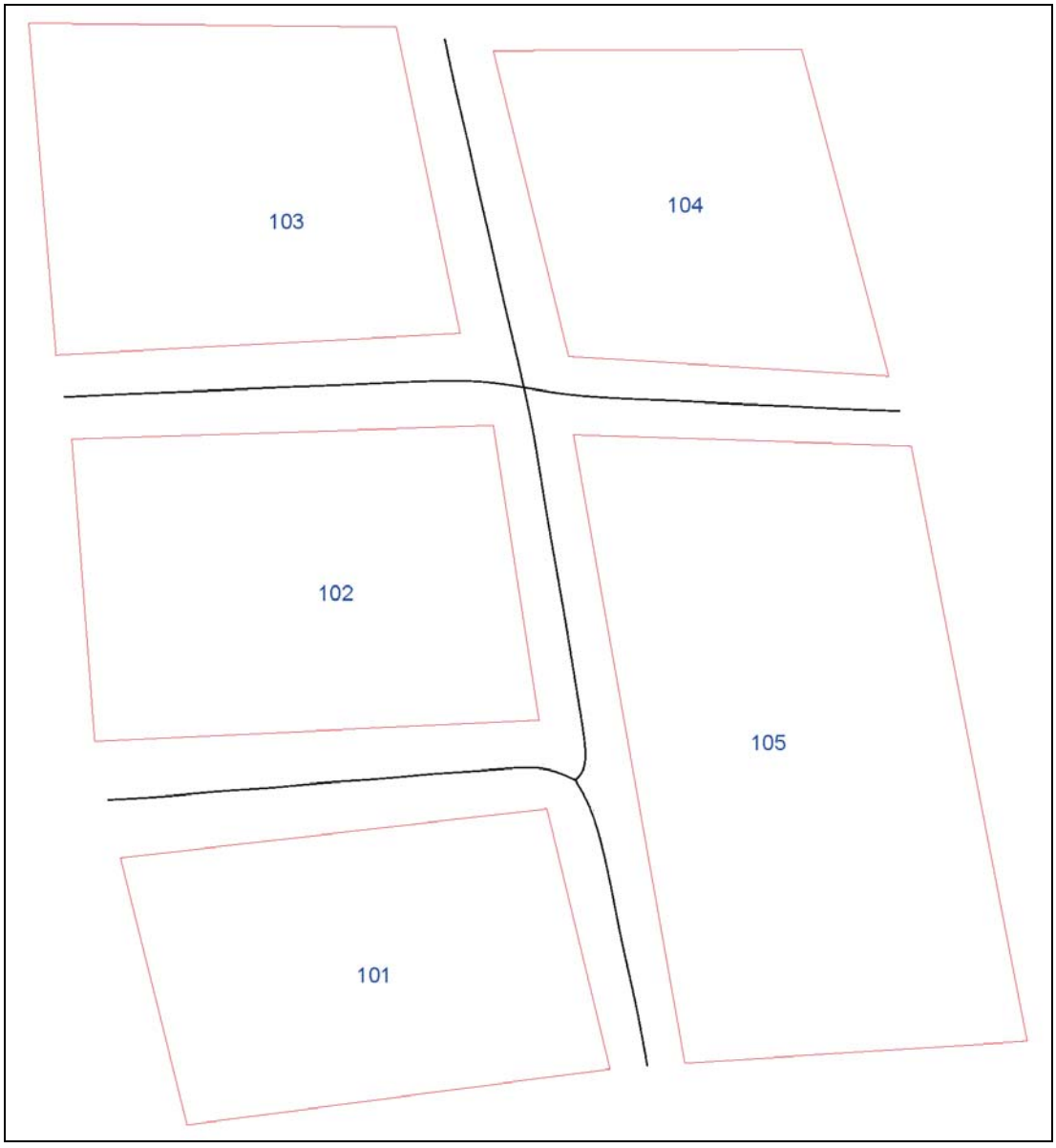
Thomas, F., 1998. Generating Street Center-Lines from Inaccurate Vector City Maps. Cartography and Geographic Information Systems, v.25,n.4,s.221-230.

Uçar, D., Bildirici, İ.Ö., Uluğtekin, N., 2003. Coğrafi Bilgi Sistemlerinde Model Genelleştirmesi Kavramı ve Geometri ile İlişkisi. Coğrafi Bilgi Sistemleri ve Jeodezik Ağlar Çalıştayı, SÜ, Konya, Eylül 24-26, s.94-103.

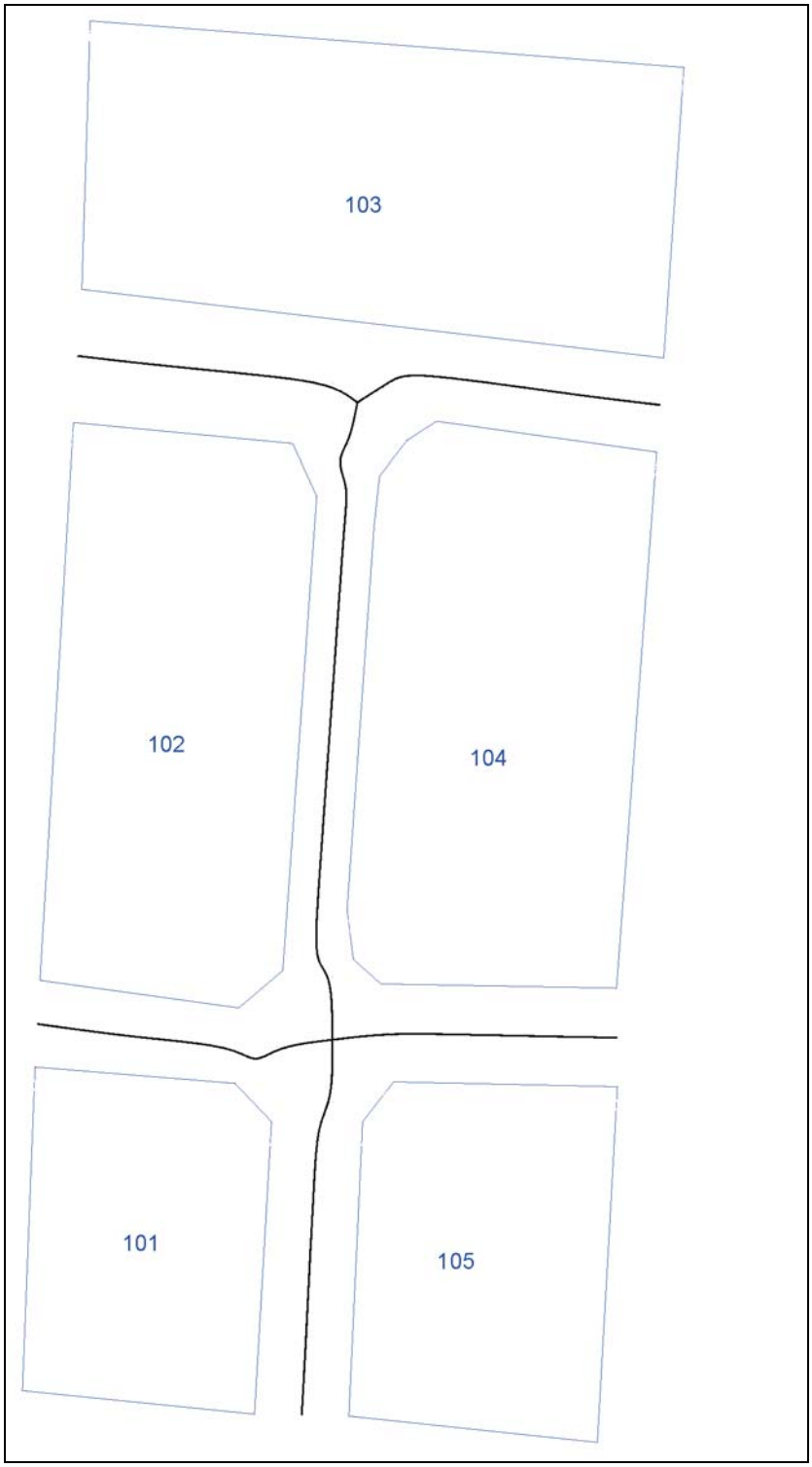
Yanalak, M., 1997. Sayısal Arazi Modellerinden Hacim Hesaplarında En Uygun Enterpolasyon Yönteminin Araştırılması.Doktora Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

Yomralıoğlu, T., 2000. Coğrafi Bilgi Sistemleri Temel Kavramlar ve Uygulamalar. Seçil ofset, İstanbul.

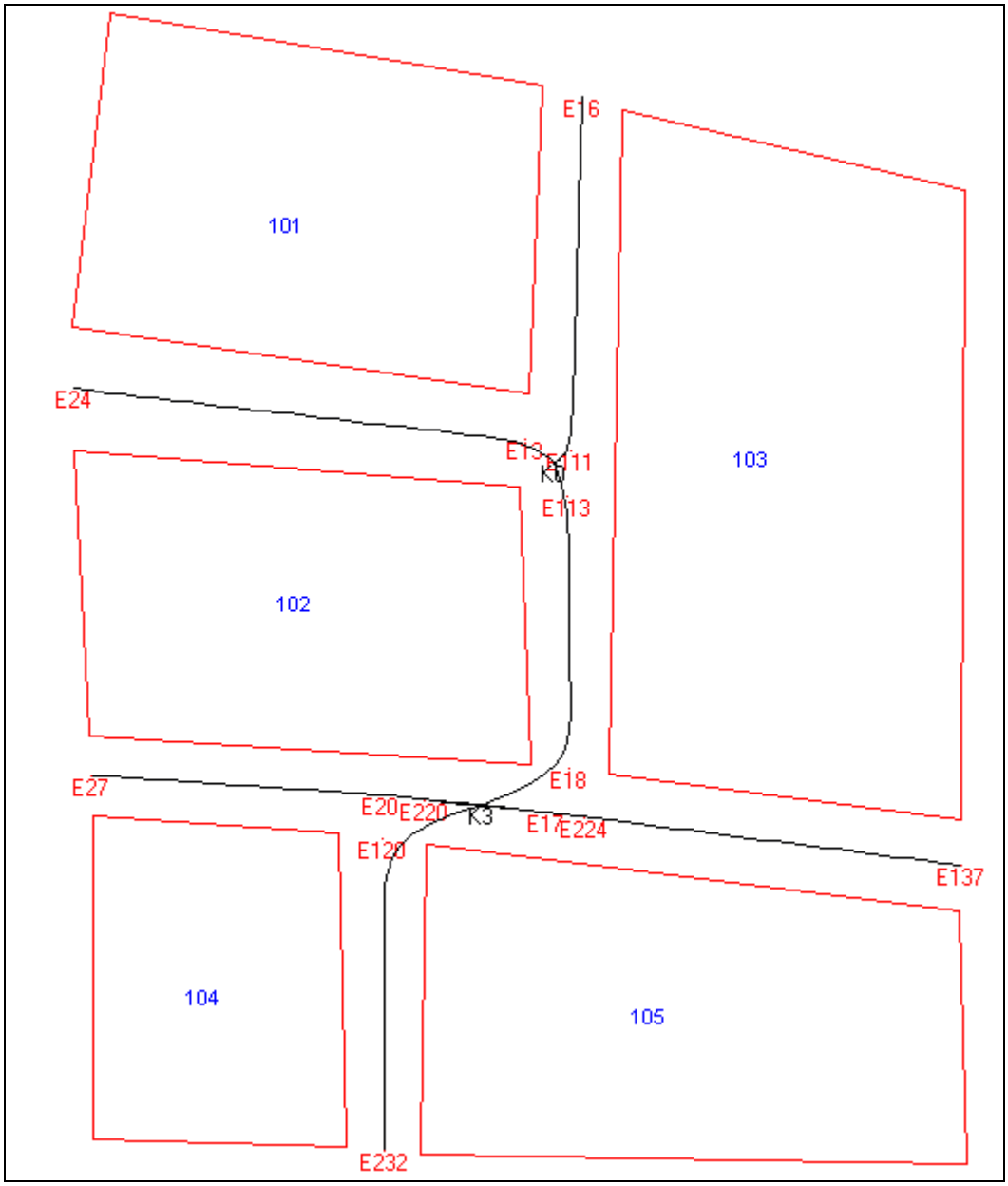
EK A:



EK B:



EK C:


101
E16
E24
E13
K1
E11
E113
103
102
E27
E18
E20
E220
K3
E17
E224
E120
E137
104
105
E232