

**HAVA FOTOĞRAFLARINDAN ÇİZGİSEL DETAYLARIN YARI OTOMATİK  
OLARAK BELİRLENMESİ İÇİN BİR UYGULAMA**  
(AN APPLICATION FOR SEMI-AUTOMATIC EXTRACTION OF LINE FEATURES  
FROM AERIAL PHOTOGRAPHS)

**Oktay EKER**

Harita Genel Komutanlığı, Ankara.  
oktay.eker@hgk.mil.tr

## ÖZET

Bu çalışmada; dijital hava fotoğraflarından çizgisel ve alansal detayların sınırlarının ve merkez hatlarının yarı otomatik olarak belirlenmesini sağlayan bir yöntem ve bu yöntemin uygulamaya konmasına yönelik geliştirilen bir yazılım tanıtılmıştır. Geliştirilen yöntem, görüntü bölümlenme ve düzey kümesi algoritmalarının birlikte kullanılmasına dayanmaktadır. Yöntemin uygulanabilirliğinin araştırılması amacıyla 1:4000, 1:16000 ve 1:25000 ölçekli siyah beyaz hava fotoğrafları ile IKONOS uydu görüntüsü üzerinde yarı otomatik detay çizme işlemleri gerçekleştirilmiştir. Elde edilen sonuçlar, bu yöntemin harita üretiminde ve coğrafi bilgi sistemleri için veri toplamada başarılı bir şekilde kullanım alanları bulabileceğini göstermiştir.

**Anahtar Kelime:** Görüntü bölümlenme, düzey kümesi, yarı otomatik, dijital hava fotoğrafı.

## ABSTRACT

In this study, a semi-automatic extraction method for extracting the center lines and the boundaries of line and area features from digital aerial photographs and the developed software of this method is introduced. The developed method is based on the image segmentation by the level set algorithms. Some semi-automatic feature extraction applications are made by the help of the developed software using 1:4000, 1:16000 and 1:25000 scale black/white aerial photographs for determining the capabilities of this method. It can be said that this method can be used for producing the maps and for collecting vector data for geographic information systems, as the results of the applications.

**Key Words:** Image segmentation, level set, semi-automatic, digital aerial photograph

## 1. GİRİŞ

Harita üretimi ve coğrafi bilgi sistemlerinin (CBS) oluşturulması için gerekli olan vektör veriler, hava fotoğrafları üzerinden, çok uzun zamandır geleneksel yollarla ve operatörler tarafından elle kıymetlendirilmektedir. Bilgisayar teknolojisi ve dijital görüntü işleme alanlarındaki gelişmeler, günümüzde bu işlemlerin otomatikleşmesine olanak sağlamaktadır. Otomatikleşmenin hedefi hızı arttırmak ve değerlendirme masraflarını azaltmaktır. Bu kapsamda yapılan araştırma çalışmaları, öncelikle,

yüzey kaplaması, geometrik şekil, genişlik gibi karakteristik özelliklere sahip olan ve modellenen detayların (yollar, binalar vb.), dijital görüntülerden otomatik ve yarı otomatik olarak çizilmesi üzerine yoğunlaşmıştır.

Gerçekleştirilen çalışmalar, otomatik ve yarı otomatik yol çiziminde, kullanılan dijital görüntülerin çözünürlüğünün çok önemli bir rolü olduğunu göstermiştir. Ortaya konan yöntemler, kullanılan dijital görüntünün çözünürlüğüne göre; düşük, yüksek ve her ikisinin birlikte kullanıldığı çok çözünürlükte yol çizim yöntemleri olarak sınıflandırılabilir (Laptev, 1997). Bu yöntemler arasında özellikle yol izleme ve Aktif Kontur Modeller (snakes) tabanlı yöntemler öne çıkmaktadır (Kass, vd., 1987).

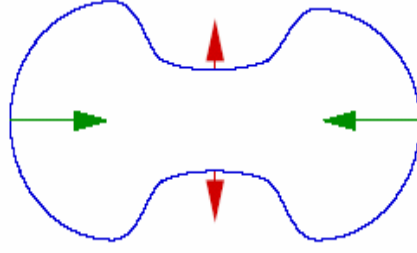
Otomatik ve yarı otomatik olarak detay belirleme ve görüntüler üzerinde sınıflandırma işlemlerinde kullanılan bir başka yöntem de görüntü bölümlenmesidir (image segmentation). Görüntü bölümlenme, tıbbi görüntüler üzerinde detay belirleme ve yorumlama konularında çok sık başvurulan bir yöntemdir (Adalsteinsson, vd., 1995). Son yıllarda, görüntü üzerindeki bölümlenme sonucunda elde edilen yüzeylerin görüntü üzerinde gelişmesini ve yayılmasını sağlamak için Düzey Kümesi ve Hızlı İlerleme (Level Set and Fast Marching) yöntemleri başarıyla kullanılmaktadır (Malladi, vd., 1994).

## 2. DÜZEY KÜMESİ ALGORİTMALARI

Dünya üzerindeki tüm nesnelere fiziksel sınırlara sahiptir. Bu sınırları; dinamik ve statik olmak üzere iki ayrı sınıfta değerlendirmek mümkündür. Dinamik sınırlara örnek olarak okyanustaki dalgaların kırılması verilebilir. Hava fotoğraflarındaki detayların sınırları ise statik sınırlara örnek olarak gösterilebilir. Sınırlarla gösterilen her bir nesne bir yüzeydir. Bu yüzeyler gelişen ya da büyüyen yüzeyler olarak kabul edilirse; bunların gelişme hızları ve yönlerinin çok iyi bilindiği durumlarda bile şekillerinin doğru olarak izlenmesi oldukça zordur. Birinci zor nokta; bir kar tanesindeki kadar karışık şekillerin keskin köşelerinin belirlenmesidir. İkinci zor nokta ise uzak kenarların birbirlerine dolanması ve karışması durumunda ortaya çıkan problemin çözümüdür. Bu probleme örnek olarak, bir orman yangınının sınırlarının, birbirinden ayrı alevlerin birlikte yanması ve kıvılcımların rüzgarla birlikte saçılmasıyla değişiklik göstermesi verilebilir. Üçüncü zor nokta ise, uygun bir şekilde temsil yolu bulunması durumunda bile üç ve daha büyük boyutlu dalgalı bir sınırdan söz etmenin zorluğudur (Sethian, 1997).

Düzey kümesi yöntemleri, keskin köşeli, topolojik değişimlere uğramış ve üç boyutlu gelişen yüzeylerin izlenmesine matematiksel ve programlanabilir araçlar sağlamaktadırlar.

Bir eğrinin karakteristik özelliği, herhangi bir noktadaki eğriliğidir. Eğrilik, eğrinin kıvrılma hızını gösterir.

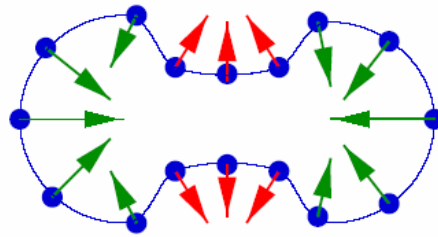


Şekil 1. Eğriliğe bağımlı olarak değişen hareketlerin yön ve büyüklükleri (Sethian, 1997).

Eğrinin her bir parçasının, eğriliğe orantılı bir hızda eğriye dik olarak hareket ettiği varsayalım. Eğrinin saat yönünde veya saat yönünün tersinde hareket etmesine bağlı olarak eğrilik pozitif ya da negatif değer alabilir. Bunun sonucunda eğrinin bazı bölümleri dışa hareket ederken diğer bölümleri de içeri doğru hareket eder. Şekil 1'deki gibi koyu renkli oklar eğriliğin negatif olduğu bölgeleri, açık renkli oklar ise eğriliğin pozitif olduğu bölgeleri göstermektedir. Oklar farklı uzunluktadırlar çünkü açık renkli oklardaki kuvvet büyüklüğü, koyu renklilerden daha büyüktür (Sethian, 1997).

Bir yüzey boyunca yürürken geçilen noktaların  $x$  ve  $y$  koordinatlarının,  $xy$  düzleminde bu noktaları çizen bir başka kişi tarafından kaydedildiği varsayalım. Yürüme hızına "parametrizasyon", diğer kişi tarafından noktaların birleştirilmesiyle çizilen eğriye de "görüntü" adı verilir. Hızlı veya yavaş olarak yürünse de hep aynı sınır çizilecektir. Bu yaklaşımın avantajı, koordinat sistemine olan bağımlılığın ortadan kaldırılmasıdır.

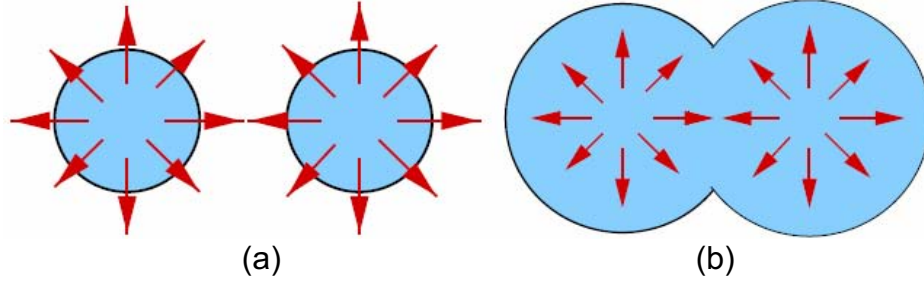
Bir yüzeyin, parametrize edilmiş sunumu bir sayısal algoritmanın omurgasını oluşturacak şekilde kullanılmak istendiğinde, eğrinin etrafında düzenli aralıklarla işaretler yerleştirilmeli ve bu işaretler bir iple birbirlerine bağlanmalıdır. İşaretler ve bağlı oldukları ip bir sınır oluşturur (Şekil 2) (Sethian, 1997).



Şekil 2. Birbirine bağlı işaretlerin oluşturduğu eğrinin sınırı (Sethian, 1997).

Okların uzunlukları ve yönleri, buldukları noktalardaki eğriliklerle belirlenir. Burada şöyle bir genişleme ya da ilerleme stratejisi uygulanabilir. Okların yönlerine bakarak işaretler ilerletilip, yeni oklar hesaplanabilir ve işaretler yeniden ilerletilebilir. Böylece ne kadar çok işaret kullanılırsa, o kadar doğru bir cevap alınacağı beklentisi içine girilebilir. Ancak bu yaklaşım, bazıları kaçınılmaz bazıları da düzeltilebilir bir takım hatalar içermektedir. Şekil 2'ye dikkatlice bakıldığında düzeltilebilecek hatalardan birini görmek mümkündür. İşaretler birbirlerinin üzerine geçmeye çalışacaklar ve geçeceklerdir. Bu durumda işaretleri birbirine bağlayan ipi düzenli bir

şekilde bir arada tutmak zorlaşacaktır. İp dolanacak ve işaretler birbirlerine karışacaklardır. Bir çözüm olarak yüzeyin ilerlemesini periyodik olarak durdurmak ve eğri boyunca yeniden yürüyerek, yeni eşit aralıklı işaretler yerleştirmek verilebilir.

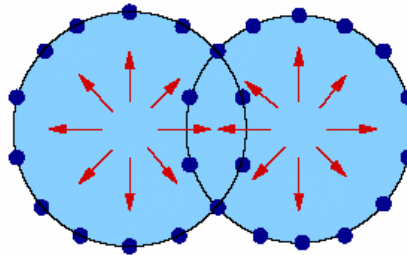


Şekil 3. (a) Başlangıç alevleri, (b) Yayılan alevlerin sonraki zamandaki durumu (Sethian, 1997).

Yayılan sınırın topolojisi değiştirilmeye çalışıldığında ise çok daha ciddi bir problem ortaya çıkacaktır. Her ikisinin de sabit hızda dışarıya doğru yandığı ve genişlediği varsayılan iki ayrı dairesel alev ele alındığında yayılan yüzeyin şekli kolayca öngörülebilir (Şekil 3) (Sethian, 1997).

İki ayrı alev birlikte büyüdüğünden, gelişen yüzeyler birleşerek tek bir yayılan yüzey oluşturmaktadırlar. Bununla birlikte; parametrizasyona dayanan bir sayısal algoritma burada büyük bir hataya neden olacaktır. İlerleyen alevin gerçek kenarı izlenmek isteniyorsa iki alevin kesişim bölgesindeki iki çift işaret bir şekilde ortadan kaldırılmalıdır (Şekil 4) (Sethian, 1997).

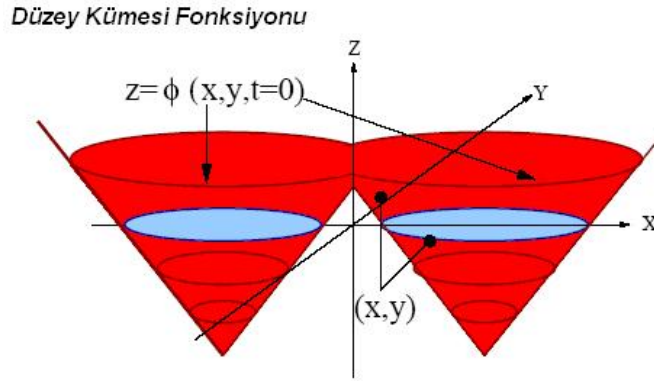
Düzye kümesi yaklaşımı bu probleme farklı bir çözüm getirmektedir. Arayüzeyin kendisini izlemek yerine, orijinal arayüzey ele alınmakta ve probleme ilave bir boyut daha eklenmektedir. Arayüzeyin içinde bulunduğu xy düzlemine ek olarak yüksekliğin ölçüldüğü z boyutunun da dahil edildiği yeni bir koordinat sistemi oluşturulmaktadır. Problem iki boyutlu bir problemden üç boyutlu bir probleme dönüştürülmüş olmaktadır.



Şekil 4. Yayılan yüzey üzerindeki işaretler (Sethian, 1997).

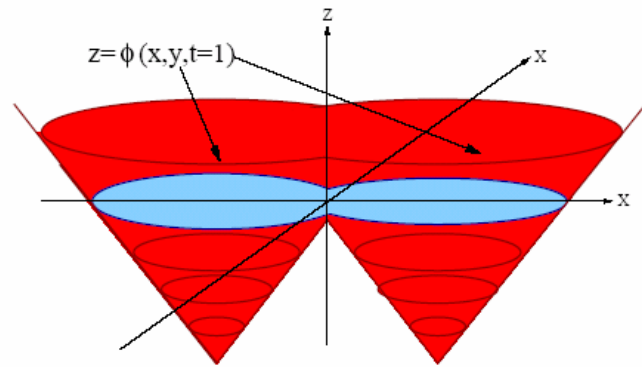
$z = \phi(x, y, t=0)$  olan bir fonksiyon ele alınsın. Bu fonksiyonda  $(x, y)$  koordinatlarıyla temsil edilen bir nokta, girdi verisi olarak alınmakta ve bu noktaya  $z$  yüksekliği atanmaktadır.  $Z$  yüksekliği,  $t=0$  anındaki arayüzey ile  $t$  anındaki arayüzey arasındaki uzunluğu ifade etmektedir. Başka bir deyişle,  $t$  anındaki arayüzeyin  $t=0$  anındaki arayüzeyden olan yüksekliği  $z$  değeri ile temsil edilmektedir. Böylece  $xy$  düzlemini arayüzeyde mutlak olarak kesen ve koyu renk ile renklendirilmiş bir yüzey

oluşturulmaktadır (Şekil 5) (Sethian, 1997). Koyu renkli yüzey, düzlem içerisindeki herhangi bir noktayı girdi olarak kabul ettiğinden ve çıktı olarak bir yükseklik değeri verdiği için, düzey kümesi fonksiyonu olarak isimlendirilmektedir. Açık renkli arayüzey, yüksekliği sıfır olan tüm noktaların birleşiminden oluştuğundan, sıfır düzey kümesi olarak adlandırılmaktadır (Şekil 5) (Sethian, 1997).



Şekil 5. Düzey kümesi yüzeyi her  $(x, y)$  noktasından açık renkli arayüzeye olan uzaklığı çıktı olarak vermektedir (Sethian, 1997).

Arayüzeyin gelişimini zamanında yakalayabilmek için  $\phi(x, y, t)$  yüzeyinin yüksekliğinin nasıl değiştirileceği belirlenmelidir. Düzey kümesi fonksiyonunun genişlemesine, yükselmesine, alçalmasına izin vermek ve bütün işlemleri yerine getirmesini sağlamak amaç olarak belirlenmektedir. Arayüzeyin herhangi bir zamandaki konumunu bulmak için yüzey, sıfır yükseklikte kesilmektedir. Başka bir deyişle, sıfır konturu çizilmektedir. İlk bakışta, hareket eden eğrinin problemini, hareket eden yüzeyinkiyi değiştirmek akıllıca görünmeyebilir. Daha çok boyut daha çok iş denebilir. Fakat burada getirilen ek boyut o kadar yararlıdır ki çarpışan ve birbirinden uzaklaşan işaretler yerine, şimdi her  $(x, y)$  noktasında durup, düzey kümesi fonksiyonun o noktadaki yüksekliği ayarlanabilecektir. Böylece, topolojik hatalar ortadan kaldırılabilir. Yayıldıklarında birleşerek tek bir alev meydana getiren iki ayrı alevin belirli bir zamandaki sıfır düzey kümeleri, iki yerine sadece bir eğri oluşturacaktır (Şekil 6) (Sethian, 1997).



Şekil 6. Koyu renkli düzey kümesi fonksiyonu hareket ettirilerek yeni açık renkli arayüzey oluşturulur (Sethian, 1997).

İşaretleme yöntemleri, basketboldaki adam adama savunmaya, düzey kümesi yöntemleri ise alan savunmasına benzetilebilir. Düzey kümesi yaklaşımının üç veya daha büyük boyutlu problemlerin çözümünde kullanılmasının hiçbir farkı yoktur. Daha büyük boyutlu problemlerde görüntülemenin zorlaşması dışında stratejide hiçbir değişiklik yoktur. Dikkat edilmesi gereken tek konu; hangi boyutta çalışılırsa çalışılsın düzey kümesi fonksiyonu oluşturulurken, ilave olarak bir boyut eklenmesi gerekliliğidir.

### 3. GELİŞTİRİLEN YÖNTEM

Tasarlanan yarı otomatik veri toplama yöntemi, düzey kümesi ile görüntü bölümlenme algoritmalarına dayandırılmıştır. Burada, çözülmesi gereken üç problemle karşılaşmıştır. Birincisi; algoritmanın nasıl başlatılacağıdır. Bu problem, operatör tarafından çizilmek istenen detayın üzerinde herhangi bir noktanın (pikselin) işaretlenmesiyle çözülmüştür. Böylece düzey kümesi algoritması operatörün seçmiş olduğu noktadan itibaren çalışmaya başlayacaktır.

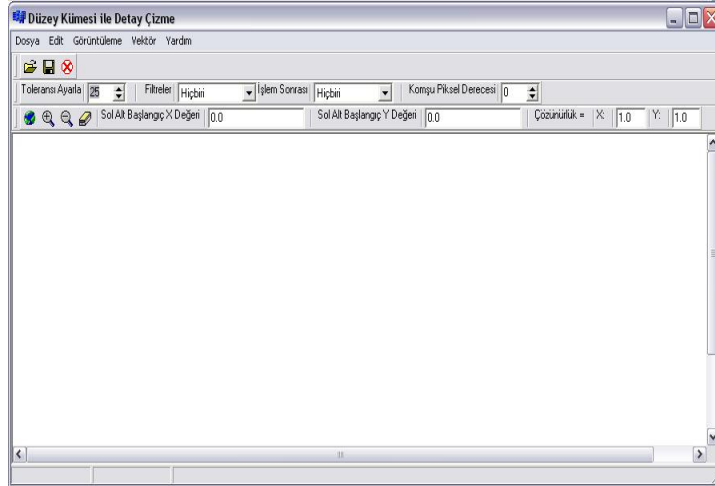
İkinci problem, işaretlenen noktadan itibaren detayın çizilmesi işlemine devam edilip edilmeyeceğine, hangi kritere göre karar verileceğidir. Bu problem ise raster görüntülerin meydana geldiği piksellerden her bir pikselin sahip olduğu renk değerleri kullanılarak çözülmüştür. İşaretlenen noktanın sahip olduğu renk değeri veya belirlenen komşuluk derecesinde, komşu piksellerin renk değerleri ile ortalamasının alınmasıyla bulunan renk değeri, komşu piksellerin renk değerleri ile karşılaştırılmış ve renk farkı belirlenen bir tolerans değeri içinde kalmışsa algoritma devam ettirilmiş, değilse durdurulmuştur. Başka bir deyişle renk farklılığına bağlı bir bölümlenme gerçekleştirilmiştir.

Çizilmesi istenen detay belirlendikten ve işaretleme işleri tamamlandıktan sonra bu detayın bir vektör veri olarak elde edilmesi gerekmektedir ki; bu sayede harita üretiminde veya CBS'nde kullanılabilir. Bu da üçüncü problemi oluşturmaktadır. Bu problemin çözümü için de raster veriden vektör veriye dönüşüm işlemi gerçekleştirilmiştir. Yukarıda bahsedilen üç problemi çözen yarı otomatik veri toplama yöntemi, iş akışı 5 ana adımdan oluşan bir işlemler bütünü şeklinde tasarlanmıştır.

Bu işlem adımları:

- Operatör tarafından değerlendirilmek istenen detayın çizilmesi için başlangıç noktasının veya pikselinin seçimi,
- Seçilen görüntü pikselinin, komşu piksellerle olan renk farkından faydalanılarak görüntü bölümlenmesinin gerçekleştirilmesi,
- Düzey kümesi yöntemi kullanılarak bölümlenmenin yayılması ve yığın yapısı (heap sort) şeklinde depolanması,
- Bu yapıdaki piksellerden 1 bitlik (1 renkli) raster maske görüntüsünün elde edilmesi,
- Elde edilen raster maskeden vektöre dönüştürme işleminin koordinatlı olarak gerçekleştirilmesini müteakiben, bilinen bir formatta, detaylara ait vektör verilerin elde edilmesidir.

İlk 4 işlem adımı Borland C++ programlama dili kullanılarak ve sistem kütüphaneleri dışında hiçbir kütüphane dosyası kullanılmadan gerçekleştirilmiştir (Şekil 7).



Şekil 7. Ana arayüz.

5'nci adım ise internet ortamında bulunan, bir raster görüntüden vektöre dönüşüm açık kodunun, Visual C++ kullanılarak yeniden düzenlenip, kırıklık toleransı ve köşe koordinatları giriş seçeneklerinin ilave edilerek işlevsellik kazandırılmasıyla gerçekleştirilmiştir (Şekil 8).

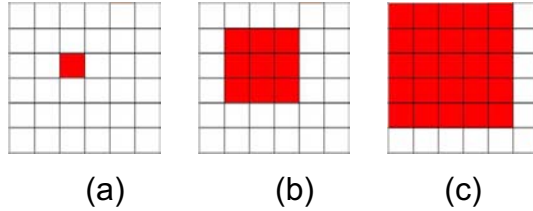


Şekil 8. İkinci arayüz (vektör işlemleri penceresi).

### a. Görüntü Bölümleme Algoritması

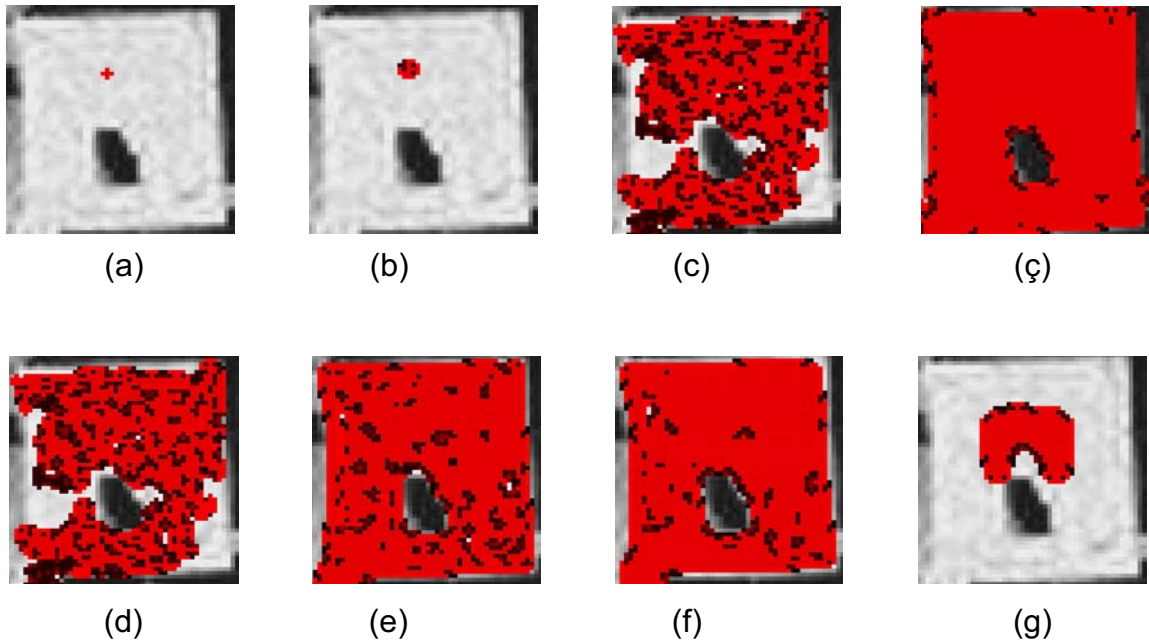
Görüntü bölümlenmesi piksellerdeki renk farkına dayandırılmış olup, bu basit algoritmaya operatör tarafından ayarlanabilir bir tolerans değeri eklenerek esneklik kazandırılmıştır. Böylece operatöre, kontrastlığın fazla olduğu yerlerdeki detaylarda toleransı yüksek tutarak sadece bir kez işaretlemek suretiyle büyük alanları seçebilme kabiliyeti kazandırılması öngörülmüştür. Bölümlenme algoritmasının eşik değeri olarak görüntünün genişliği ve boyunun çarpımının 2 katı kadar bir değer belirlenmiştir.

Bölümlleme, ilk olarak işaretlenen detay noktası temel alınarak başlatılır. Bu detay noktasına karşılık gelen pikselin ve/veya belirlenen komşuluk derecesine göre, komşu piksellerin renk değerleri ile ortalamasının alınması sonucunda elde edilen renk değeri, referans olarak alınır. Komşuluk derecesi 0 ise, o pikselin renk değeri, komşuluk derecesi 1 ise etrafındaki 8 piksel ile birlikte ortalamasının alınması sonucunda bulunan renk değeri, başlangıç değeri olarak alınır (Şekil 9).



Şekil 9. Komşuluk derecesi kavramı, (a) Komşuluk derecesi 0, (b) Komşuluk derecesi 1, (c) Komşuluk derecesi 2

Burada referans noktasının renk değerleri ile komşu piksellerin renk değerleri arasındaki fark hesaplanırken üç banttaki (kırmızı, mavi ve yeşil) renk farklılıkları ayrı ayrı hesaplanır ve her üçünün de tolerans değerinden küçük olup olmadığı araştırılır.



Şekil 10. Tolerans değeri ve komşu piksel derecesi değerlerinin detay belirleme işlemlerine etkileri.

Tolerans değeri ve komşu piksel derecesi değerlerinin detay çizim işlemine etkileri Şekil 10' da gösterilmiştir. İlk dört şekilde, komşuluk derecesi 0 olarak alınmış, başka bir deyişle, işaretlenen noktanın renk değerleri referans olarak alınmış olup, diğer piksellerle bir ortalama alma işlemi gerçekleştirilmemiş ve sadece tolerans değerleri sırasıyla 1, 5, 10 ve 25 olarak ayarlanmıştır (Şekil 10 (a), (b), (c), (ç)). İkinci dört



şekilde ise tolerans değeri sabit ve 10 olarak ayarlanmış olup, sadece komşuluk derecesi sırasıyla 1, 2 ve 16 olarak değiştirilmiştir (Şekil 10 (d), (e), (f), (g)). Burada, ortaya çıkan farklılıkların hatasız olarak tespit edilebilmesi amacıyla her şekilde aynı piksel başlangıç pikseli olarak seçilmiştir (Şekil 10 (a)). Bu araştırma göstermiştir ki; komşu piksel derecesinin artırılması her zaman iyi sonuçlar vermemektedir. Özellikle seçilen piksele yakın pikseller değerlendirilmeye katıldığında sonuçlar oldukça iyileşmekte fakat pikselden çok uzaklaşınca aynı sonuçları elde etmek için daha büyük tolerans değerine ihtiyaç duyulmaktadır.

## **b. Düzey Kümesi ile Yayılma ve Depolama**

Düzey kümesi algoritması ile yayılmanın ya da detay boyunca ilerlemenin gerçekleştirilebilmesi için gerekli olan başlangıç değerine, geliştirilen yöntemin yarı otomatik doğasıyla birlikte çözüm bulunmuştur. Yüzeyin yayılmasına, operatör tarafından işaretlenen pikselden itibaren başlanacak ve sıfır düzey kümesi fonksiyonu bu pikselin konumu ile tanımlanacaktır.

Düzey kümesi fonksiyonu için gerekli olan diğer bir bileşen ise yayılmanın kontrolünü sağlayan sınır değeridir. Sınır değeri için çözüm yukarıda anlatılan renk farkının hesaplanmasıyla gerçekleştirilmiştir.

Geriye, yayılmanın başlatıldıktan sonra, sınır değerini sağlayan komşu piksellerden hangisinden devam edeceği probleminin çözümü kalmıştır. Bu problemin çözümünde, "Hızlı İlerleme" (Fast-Marching) yönteminden faydalanılabilir. Fonksiyon, en küçük değere sahip pikselden itibaren yayılmaya devam edebilir (Sethian, 1998). Fakat en küçük değer neyi ifade edecektir?

Geliştirilen yöntemde, en küçük değer neyi temsil etmelidir sorusu için operatör tarafından işaretlenen ilk piksel sıfır düzey kümesi fonksiyonu olarak kabul edilmiş ve her komşu piksel (doğu, batı, kuzey, güney yönündeki ilk pikseller) yukarıda açıklanan renk farkı algoritmasıyla kontrol edilmiş ve şartı sağlayan pikseller için sıfır düzey kümesinden (ilk işaretlenen piksel) olan uzaklıklar hesaplanmıştır ve böylece en küçük uzaklığa sahip pikselden itibaren yayılmanın devam ettirilmesi sağlanmıştır.

Burada şu soru sorulabilir: Uzaklıkların eşit olması durumunda ilerlemeye hangi pikselden itibaren devam edilecektir? Bu problem, başlangıçtan sonraki ilk adımda ortaya çıkabilecek bir problemdir. Çünkü başlangıçta dört komşu piksel vardır ve şartı sağlayan pikseller aynı uzaklığa (sadece bir birim uzunluğa) sahiptirler. Böyle bir durumda, en küçük renk farkına sahip pikselden itibaren yayılma devam ettirilerek probleme bir çözüm getirilmiştir.

Yayılmanın tamamlanabilmesi için güncelleme işleminin yapılması gerekmektedir (Sethian, 1998). Güncelleme işleminde, uzaklığın hesaplanabilmesi için gerekli olan, kareselleştirme algoritmaları kullanılmıştır (Sethian, 1998).

İşaretlenen piksellerin depolanması ve erişimi için minimum yığın yapısı (heap sort) kullanılmıştır. Minimum yığın yapısında, en küçük uzaklığa sahip görüntü hücresi en tepede olacak şekilde bir yapılanma sağlanmıştır. Yığın yapısının

korunması için bazı fonksiyonlara ihtiyaç duyulmaktadır. Yığına görüntü hücresinin eklenmesi, görüntü hücresinin yığında sabitlenmesi, en küçük uzaklığa sahip görüntü hücresinin yığından çıkartılması ve yığın her yeni görüntü hücresi eklendiğinde veya çıkartıldığında yapısının yeniden güncellenmesi gerekmektedir.

Bu depolama yapısı sayesinde görüntü hücrelerine ulaşılması, yayılmanın test edilmesi ve hesaplanması ile görüntü hücrelerinin işaretlenmesi ve işaretlenen görüntü hücrelerinin depolanması gibi büyük hacimli işlemlerde hesaplama etkinliğinin artırılması amaçlanmıştır.

### **c. Rasterdan Vektöre Dönüştürme**

Seçilen detayın yarı otomatik olarak ekran üzerinde işaretlenmesinden sonra bu verilerin harita üretiminde, CBS için gerekli olan coğrafi veri tabanlarının oluşturulmasında kullanılabilmesi ve bu verilere anlam kazandırılabilmesi için vektör veri haline dönüştürülmesi gerekmektedir. Vektör veri yapısında coğrafi varlıklar nokta, çizgi veya alanlar olarak temsil edilirler. Bu modelde, bir coğrafi varlık nokta olarak temsil edilecekse sadece bir koordinat çifti, çizgi şeklinde temsil edilecekse koordinat çiftlerinden oluşan bir küme, alan şeklinde temsil edilecekse başlangıç ve son koordinatları aynı olan koordinat çiftleri kümesi kullanılır.

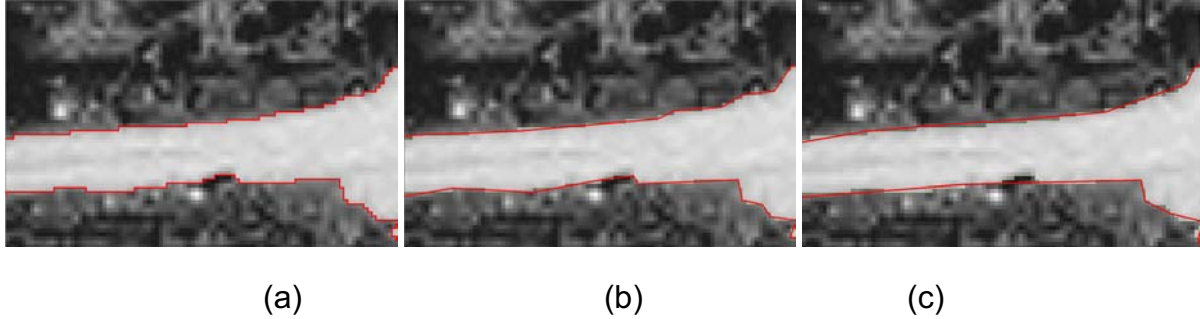
Rasterdan nokta detaya dönüşüm yapılırken, her bir piksele karşılık bir nokta detay oluşturulur. Çizgi detaya dönüşüm yapılırken, birbirine komşu sıralı pikselleri çevreleyen alanın merkezinden geçen çizgi elde edilir. Raster veri içerisinde aynı değere sahip birbirine komşu pikseller topluluğu bir alan oluştururlar. Bu alanın dış sınırı çevrilerek, raster veri alan tipindeki vektör detaya dönüştürülür (ESRI, 1997).

Rasterdan vektöre dönüşüm algoritmaları sadece iki tip veri (0 veya 1) içeren 1 bitlik görüntü dosyaları için geçerlidir. Bu yüzden geliştirilen yazılımda, görüntü hücrelerinin seçimi ve işaretlenmesi işlemleri tamamlandıktan sonra, işaretli pikseller dolu, diğerleri ise boş olacak şekilde bir görüntü dizisi oluşturulmaktadır. Bu görüntü dizisinin oluşturulması bir çeşit maskeleyme işlemidir. Elde edilen maske raster görüntü daha sonra 1 bitlik (tek renkli) BMP formatında kaydedilmektedir.

Elde edilen maske görüntü dosyasının, vektör veriye dönüştürülmesi amacıyla Davide Libenzi tarafından geliştirilen ve kendi internet sitesinde açık olarak bulunan Visual C++ kodu yeniden düzenlenmiş ve ilave imkanlar kazandırılmıştır. Ana programla bağlantısı kurularak, etkileşimli bir arayüzde raster veriden, detayların merkez hatlarının ve sınır hatlarının ayrı ayrı vektöre dönüştürülmesi sağlanmış, ana program üzerinde sol alt köşe koordinatları ile her iki boyutta (x,y) görüntü çözünürlüklerinin girilmesi suretiyle koordinatlı bir vektör veri elde edilmesi sağlanmıştır. Ayrıca etkileşimli arayüz üzerinde kırıklık toleransının girilmesiyle istenilen yumuşaklıkta ve kırıklıkta vektör veri elde etme olanağı sağlanmıştır. Kırıklık toleransı ile rasterdan vektöre dönüşüm sonucunda elde edilen vektörün kırıklığı ayarlanmaktadır.

Kırıklık toleransı 0 olduğunda hiç yumuşatma yapılmadan tüm pikseller hesaplamaya dahil edilir, kırıklık toleransı arttırıldıkça tüm pikseller yerine, artan

aralıklarla pikseller dikkate alınır ve elde edilen vektör daha düz bir hale gelir fakat kırıklık toleransının çok fazla artırılması durumunda geometrik doğruluğun bozulması söz konusudur (Şekil 11 (a), (b), (c)).



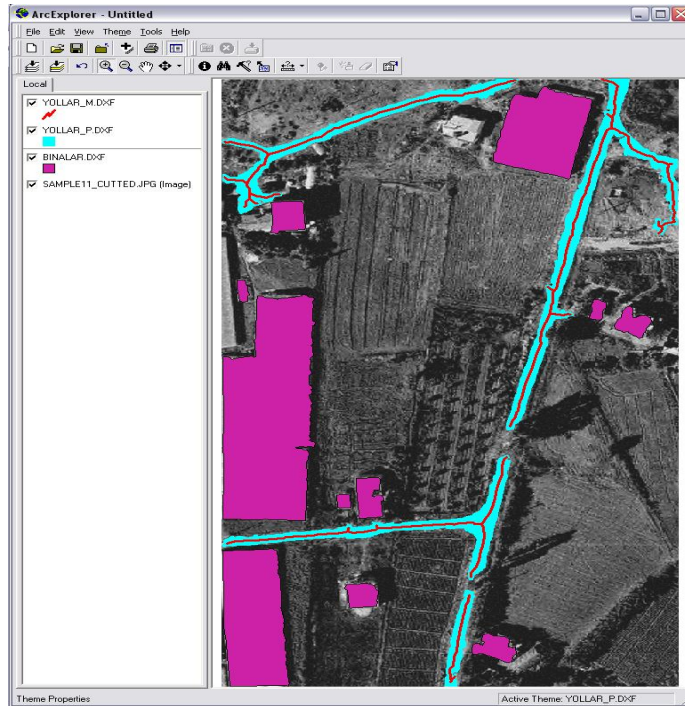
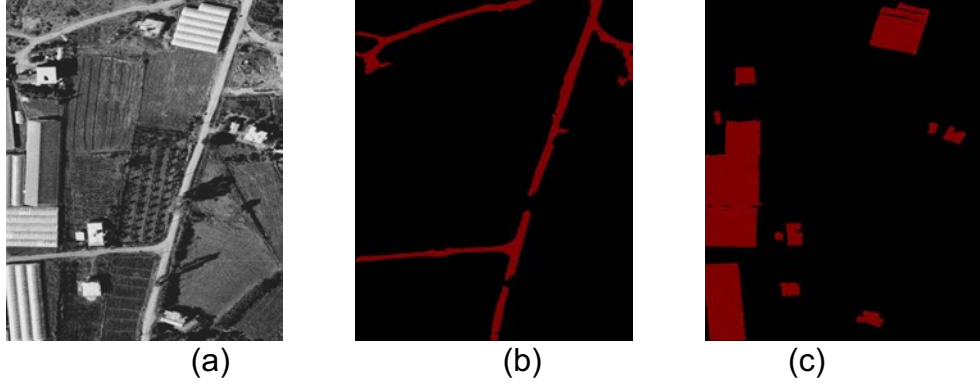
Şekil 11. Kırıklık toleransının vektör veriye dönüştürme işlemine etkisi, (a) Kırıklık toleransı=0, (b) Kırıklık toleransı=1, (c) Kırıklık toleransı=2.

#### 4. GERÇEKLEŞTİRİLEN UYGULAMALAR

Geliştirilen yazılımın test edilmesi amacıyla; 1:4000, 1:16000 ve 1:25000 olmak üzere 3 farklı ölçekte ve özelliklerde seçilmiş hava fotoğraflarıyla birlikte 1 adet IKONOS uydu görüntüsü üzerinde çizgisel detayların değerlendirilmesi çalışmaları gerçekleştirilmiştir. Test görüntülerinin seçiminde arazinin farklı dokularına sahip olmasına dikkat edilmiştir. Böylece yazılım ile farklı başarı ölçütlerinde elde edilen vektör verilerin hangi arazi dokusuna sahip fotoğraflarda ve hangi ölçeklerde harita üretiminde etkin bir şekilde kullanılabileceğinin tespit edilmesi amaçlanmıştır.

Bu amaç doğrultusunda yapılan çalışmalar ve elde edilen sonuçlar aşağıda ayrı ayrı maddeler olarak anlatılmıştır:

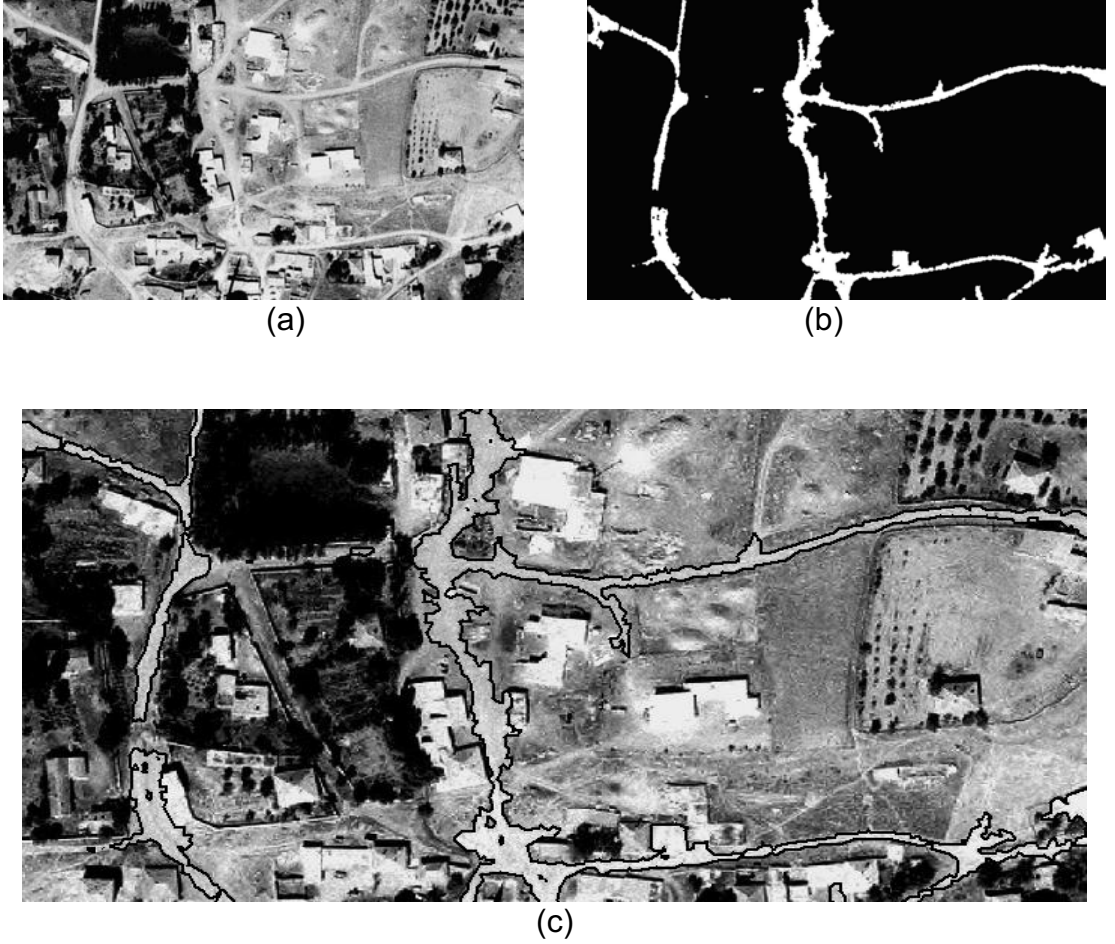
- 1:4000 ölçekli, 20 mikronda taranarak sayısallaştırılmış siyah/beyaz hava fotoğraflarının kullanıldığı birinci test çalışmasında, fotoğraf üzerinde çizgisel detay olarak yollar ve binalar seçilmiştir. Bu seçilen detaylara ait yapılan sayısallaştırmalar sonucunda maske raster görüntüler ve vektör veriler elde edilmiştir (Şekil 12). Burada yollar merkez hatlarından ve sınırlarından vektöre dönüştürülmüştür ama aslında merkez hatlarının vektöre dönüştürülmesi çok önemli değildir çünkü bu ölçekteki hava fotoğraflarının kullanıldığı büyük ölçekli fotogrametrik uygulamalarda yollar kenarlarından kıymetlendirilmektedirler. Elde edilen sonuçlar oldukça umut vericidir. Yolların ve binaların üzerine düşen ağaç vb. engeller dışında en küçük tali yollar bile kolaylıkla çıkarılabilmektedir. Detaylar üzerine gelen engellerin neden olduğu kesiklikler bu algoritmayla çözülememiştir ve bu eksiklerinin operatör tarafından elle düzeltilmesi gerekmektedir.



Şekil 12. (a) Orijinal görüntü, (b) Yollara ait maske raster görüntü, (c) Binalara ait raster maske görüntü, (ç) Elde edilen vektör veriler.

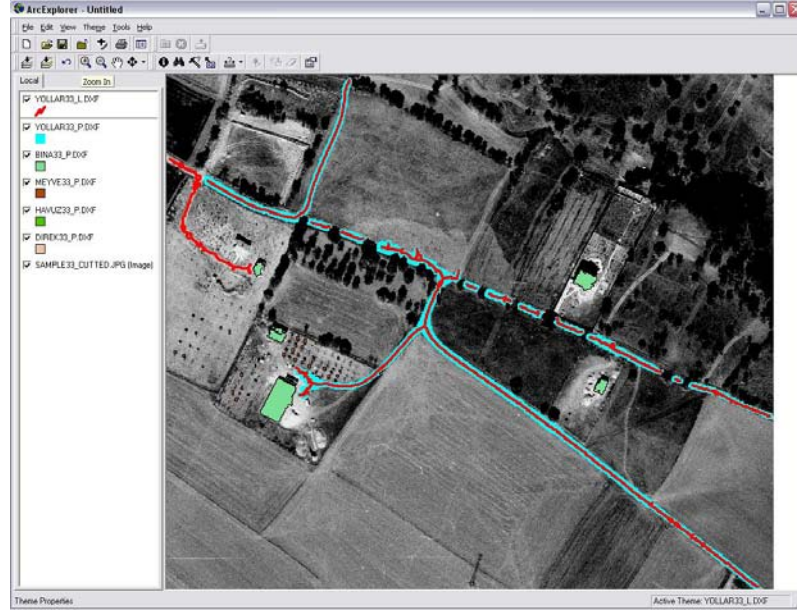
• 1:16000 ölçekli, 20 mikronda taranarak sayısallaştırılmış ve bir köy merkezini içeren siyah/beyaz hava fotoğrafının kullanıldığı ikinci test çalışmasında ise çizgisel detay olarak yollar ele alınmış, binalar dikkate alınmamıştır. Çünkü binalarda başarı derecesinin çok az olacağı baştan bellidir. Binaların çatıları çok belirgin değildir ve hava fotoğrafının kontrastlığı binaların kolayca çıkarılabilmesi için elverişli değildir. Köy merkezindeki yolların, yüzey kaplamalarının olmamasının çevrelerindeki detaylardan ayrılmalarını zorlaştırdığı görülmüştür. Burada tolerans değerinin yüksek tutulması durumunda algoritmanın yolların dışına doğru taşıdığı ve yol olmayan çevre detaylara ait piksellerinde işaretlendiği gözlenmiştir (Şekil 13). Bu durumdan kaçınmak için düşük tutulan tolerans değeri de yolların doğru olarak çıkarılması için oldukça fazla işaretleme yapılmasına ve algoritmanın küçük küçük adımlarla

ilerlemesine neden olmuştur. Bu istenmeyen bir sonuçtur çünkü yarı otomatik bir algoritmanın felsefesine ters düşmektedir. İşlem adımları hızlanmamış aksine çok yavaşlamıştır.



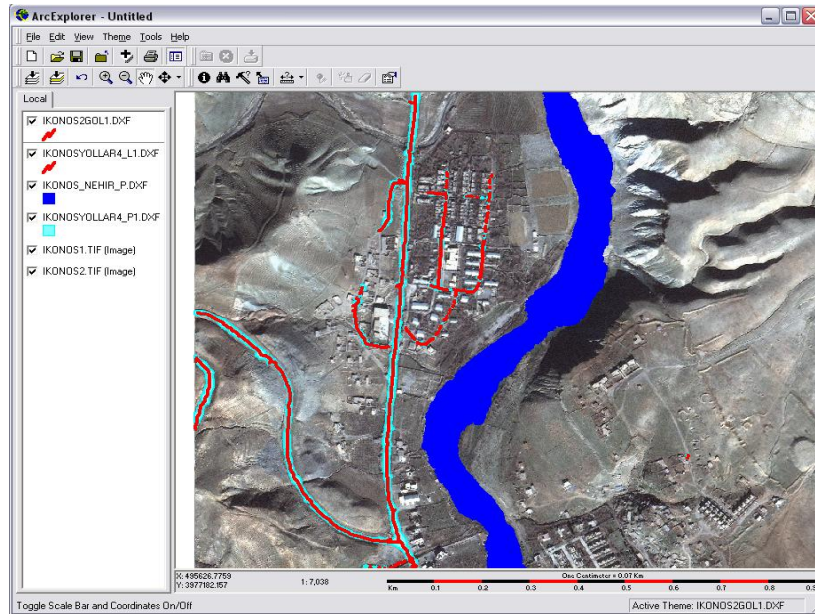
Şekil 13. (a) Orijinal görüntü, (b) Yollara ait maske raster görüntü, (c) Elde edilen vektör veriler.

- Üçüncü test çalışmasında; 1:25000 ölçekli, 20 mikronunda taranarak sayısallaştırılmış siyah/beyaz hava fotoğrafı kullanılmıştır. Bu test, bütün test verileri arasında en başarılı sonuçların alındığı test olarak ortaya çıkmıştır. Bunun nedenleri araştırıldığında en büyük etkenin görüntü kalitesinin ve kontrastlığının olduğu gözlenmiştir. Kullanılan ölçek küçüldüğünden yukarıdaki test çalışmalarından farklı olarak burada yolların sınırlarına ilave olarak merkez hatların sayısallaştırılmasına da önem verilmiştir. Bu şekilde elde edilen sonuçların çok tatminkar olduğu görülmüştür (Şekil 14).



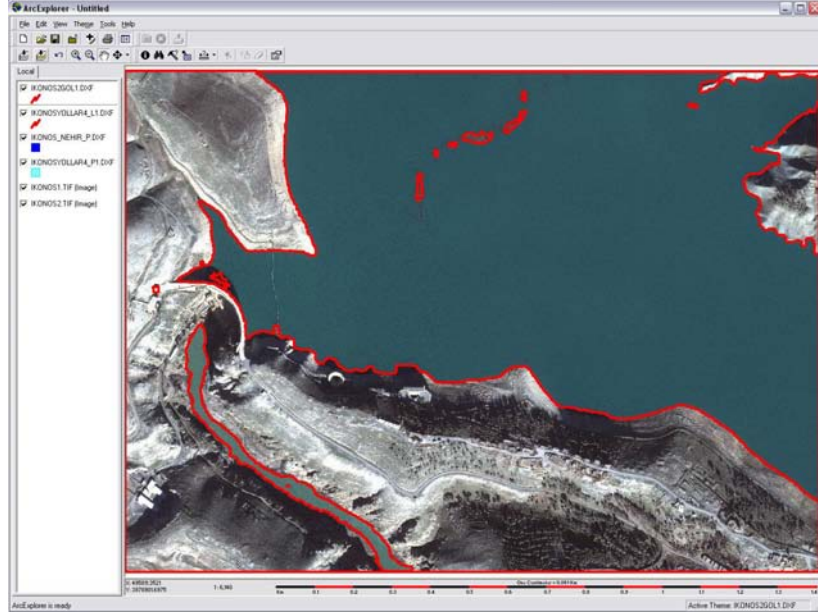
Şekil 14. 1:25000 ölçekli hava fotoğrafından elde edilen vektör veriler.

- Bu araştırma çalışmasının kapsamına girmemekle beraber uydu görüntülerini kullanarak da test çalışmaları gerçekleştirilmiştir. İlk olarak; 1 m. mekansal ayırma gücüne sahip renkli (pan-sharpened) IKONOS uydu görüntüsü kullanılmış olup, söz konusu uydu görüntüsünün farklı iki bölgesinde detay belirleme ve çizim işlemleri gerçekleştirilmiştir. Kullanılan uydu görüntüsü orto rektifiye edilmiş bir görüntü olduğundan elde edilen vektör veriler koordinatlı olarak kaydedilebilmiştir. Birinci bölgede yollar, binalar ve sulı dere ile ilgilenilmiş ve çok iyi sonuçlar elde edilmiştir (Şekil 15).



Şekil 15. IKONOS görüntüsünden elde edilen koordinatlı vektör veriler (yollar, binalar ve dere).

İkinci bölgede ise bir göl ve ona bağlı bir sulu dere çıkarılmış ve vektöre dönüştürülmüştür. Bu bölgedeki çalışmada önemli olan nokta, göl ve sulu dere gibi büyük alan detayların sınırlarının, tolerans değeri yüksek tutularak sadece birkaç kez işaretlenmesi suretiyle çok kısa bir sürede sayısallaştırılmasının gerçekleştirilmesidir (Şekil 16).



Şekil 16. IKONOS görüntüsünden elde edilen koordinatlı vektör veriler (Kırmızıyla gösterilen göl ve dere sınırları).

## 5. SONUÇ

Bu çalışmanın amaç ve hedeflerine paralel olarak, renk farkları ile düzey kümesi yöntemlerini baz alarak geliştirilen yarı otomatik düzey kümesiyle görüntü bölümlenmesi ve buna bağlı olarak çalışan rasterdan vektöre dönüştürme yazılımlarının uygulanabilirliğinin, işlevselliğinin ve etkinliğinin test edilmesi amacıyla farklı ölçek ve türlerdeki hava fotoğrafları ile uydu görüntüleri üzerinde yarı otomatik vektör toplama işlemleri gerçekleştirilmiştir.

Bu yöntemin fotogrametrik harita üretiminde ve CBS için fotogrametrik veri sağlanmasında yeni bir yöntem olarak kullanılabilmesi düşünülmektedir.

a. Yapılan test çalışmaları sonucunda bu yöntemle veri toplama işlemlerinde diğer yöntemlere göre bazı avantajlar elde edilebileceği görülmüştür:

(1) Bu yöntem, özellikle göller, sulu dereler ve binalar gibi homojen yapıdaki detayların sınırlarına ait vektör verilerin toplanmasında çok başarılı ve etkili bir şekilde kullanılabilir.

(2) İstenildiği takdirde, tolerans değerinin ayarlanması suretiyle, söz konusu detaylar üzerinde gözle ayırt edilemeyen sınıflandırmalar ve bölümlenmeler gerçekleştirilebilecektir.

(3) Kaliteli yolların sınırları ve/veya merkez hatları (kullanılan fotoğrafın ölçeğine bağlı olarak) etkili ve hızlı bir şekilde çıkartılabilecektir.

(4) Kırıklık toleransı değerleri ayarlanarak istenilen kırıklıkta vektör veriler elde edilebilecektir.

(5) Rasterdan vektöre dönüşümde hem sınırların hemde merkez hatların kullanılabilmesi etkinliğe çok katkı sağlayacaktır.

(6) Bu avantajlarla birlikte programın hiçbir kütüphane veya paket program dosyası kullanmaması ve başka hiçbir yazılıma bağlı olmaması, yazılımın pratikliğini ve kullanılabilirliğini arttırdığı düşünülmektedir.

b. Test çalışmaları sırasında yazılımın olumsuz olarak etkilendiği ve eksik kaldığı bazı faktörler de tespit edilmiş olup, bunlarda aşağıda sunulmuştur:

(1) Özellikle büyük ölçekli görüntülerde detaylar üzerinde bulunan engeller (ağaç, araba, gölge gibi) detay çıkarma işlemini olumsuz olarak etkilemektedir. Bu etki ölçek küçüldükçe azalmaktadır.

(2) Tolerans değerleri uygun olarak ayarlanmadığı takdirde yanlış detay çıkarımları gerçekleşebilmektedir. Tolerans yüksek tutulduğunda ilgilenilen detaylar atlanabilmekte, küçük tutulduğunda ise çok fazla data vererek küçük küçük adımlarla ilerleme sağlandığından yazılımın etkinliği azalmaktadır.

(3) Çok büyük boyutlu görüntü dosyaları kullanıldığında, tasarlanan algoritmada piksellerin radyometrik değerleri bilgisayar hafızasına kayıt edildiğinden, çok fazla hafızaya ihtiyaç duyulması ve bunun sonucu olarak bazı donanımsal hatalarla karşılaşılabilme olasılığı söz konusudur.

(4) Görüntülerin kalitesi, kontrastlığı ve gürültü oranları algoritmanın başarısını önemli ölçüde etkilemektedir.

(5) Çizgisel detayların yüzey ve kaplama özellikleri ile kontrastlıkları da algoritmanın başarısını etkilemektedir.

Söz konusu problemlerin giderilmesi ve yazılımın daha etkin bir hale getirilebilmesi için, görüntülerdeki kontrastlığın artırılması, gürültü oranlarının azaltılması için görüntü işleme algoritmalarından faydalanılarak anisotropik difüzyon gibi filtreler ile kenar zenginleştirme algoritmalarının uygulanması, engellerden kaynaklanan boşlukların doldurulması için farklı enterpolasyon yöntemlerinin kullanılması, bölümlenme işlemlerinde renk farkı yerine daha gelişmiş algoritmaların (aktif kontur modeller (snakes) vb.) kullanılması ve büyük boyutlu görüntülerin bilgisayar



ortamlarında daha kolay ele alınabilmesi için piramit seviyelerinin kullanılması çözüme yönelik yaklaşımlar olarak değerlendirilmektedir.

## KAYNAKLAR

- Adalsteinsson, D., Sethian, J.A.**, 1995. Fast Level Set Method for Propagating Interfaces, *Journal of Comp.Phys.*, **118**, 269-277.
- ESRI**, 1997. *ARC/INFO User's Guide Cell-Based Modelling With GRID*, Redlands, USA.
- Kass, M., Witkin, A.**, 1987. Snakes: Active Contour Models, *International Journal of Computer Vision*, 1, 4, 321-331.
- Laptev, I.**, 1997. Road Extraction Based on Line Extraction and Snakes, *Master Thesis*, Kungliga Tekniska Högskolan (The Royal Institute of Technology), Stockholm, Sweden.
- Malladi, R., Sethian, J.A. ve Vemuri, B.C.**, 1994. Evolutionary Fronts for Topology-Independent Shape Modeling and Recovery, *Proceedings of Third European Conference on Computer Vision, Stockholm, Sweden, Lecture Notes in Computer Science*, **800**, 3-13.
- Sethian, J.A.**, 1998. Fast Marching Methods and Level Set Methods for Propagating Interfaces, *von Karman Institute Lecture Series*, Computational Fluid Mechanics, Belgium.
- Sethian, J.A.**, 1997. Level Set Methods: An Act of Violence. *American Scientist*, 85, 3, 12-35.