

Doğrusal çizgilerden oluşan raster görüntülerin vektörizasyonu için yeni bir yöntem ve 3 boyutlu CBS’de kullanılması

İsmail Rakıp KARAŞ^{1*}, Bülent BAYRAM², Gül BATUK², Uğur ACAR², Melis UZAR²

¹ Gebze Yüksek Teknoloji Enstitüsü, Jeodezi ve Fotogrametri Mühendisliği Bölümü, Gebze, Kocaeli

² YTÜ, İnşaat Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümü, 34349, Yıldız, Beşiktaş, İstanbul

Özet

Bu çalışmada raster görüntülerin vektörizasyonuna yönelik olarak yeni bir model geliştirilmiştir. Geleneksel yöntemlerden farklı olarak geliştirilen model bir yanıyla, bir çizgi inceltme algoritmasıdır. Söz konusu vektörizasyon algoritması sayesinde, zincir kodlama ve vektör azaltma işlemlerine gerek kalmadan, sadece basit komşuluk analizi ile vektörel çizgileri elde etmek mümkün olmaktadır. Model özellikle doğrusal çizgilerin vektörizasyonunda başarılı sonuçlar vermektedir. Dolayısı ile sadece haritalar değil, büyük ölçüde doğrusal hatlardan oluşan ve kağıt ortamından sayısal hale getirilecek her türlü dökümanın dönüşümünde kullanılabilir. Nitekim çalışmada bu türden dökümanlar test edilmiş, farklı yazılımlarla yapılan vektörizasyonların sonuçlarıyla karşılaştırılmıştır. Bunların dışında model üzerinde çalışılarak, başka alanlarda da kullanılabileceği görülmüştür. Bunlardan biri Üç Boyutlu Coğrafi Bilgi Sistemlerinde ağ analizi uygulamalarıdır. MUSCLE Model kullanılarak karmaşık ve yüksek binalara ait mimari planlardan, geometrik ve topolojik bina modellerinin otomatik olarak elde edilebileceği, böylece 3 boyutlu ağ analizlerinin başarıyla gerçekleştirilebileceği gösterilmiştir. Bu bildiride modelin ayrıntılarından bahsedilecek ve söz konusu uygulamalar tanıtılacaktır.

Anahtar Kelimeler: Vektörizasyon, Raster vektör dönüşümü, Coğrafi Bilgi Sistemleri, 3 Boyutlu GIS, Çizgi inceltme.

A new method for vectorizing raster images with straight lines and its usage in 3D GIS

Abstract

This paper presents a new model, MUSCLE (Multidirectional Scanning for Line Extraction), for automatic vectorization of raster images with straight lines. The algorithm of the model implements the line thinning and the simple neighborhood methods to perform vectorization. In this model, various raster images can be vectorized such as township plans, maps, architectural drawings, and machine plans. The algorithm of the model was developed by implementing an appropriate computer programming and tested on some images. The results indicate that the model can successfully vectorize the raster data quickly and accurately. Also the model was adapted and applied for a 3D GIS application. It was indicated that it is possible to automatically obtain the network and building models of the complex building and perform 3D network analysis successfully using the MUSCLE models.

Keywords: Vectorization Raster to vector conversion, Line thinning, Geographic information systems, 3D GIS.

*Yazışmaların yapılacağı yazar: İsmail Rakıp KARAŞ. ragib@gyte.edu.tr; Tel: (262) 605 31 63.

Giriş

Coğrafi Bilgi Sistemi kurulumunda en külfetli ve pahalı aşama veri üretimidir. Bazı araştırmalara göre, bir CBS için veri toplamının toplam maliyete oranı % 60-80 düzeyindedir. Bu yüzden bir CBS'nin ihtiyacı olan veriyi mevcut verilerden yararlanarak elde etmek sıkça başvurulan bir yöntemdir. Bunun da fizible olabilmesi için hızlı ve doğru bir şekilde elde edilmesi gerekir. Mevcut grafik verilerden yararlanmada kullanılan en yaygın yöntemlerden biri ise çeşitli altlıklar üzerine çizili olan haritaların sayısallaştırılarak raster hale getirilmesi ve ardından vektörizasyonudur. Değişik mühendislik uygulamaları kapsamında kullanılan kağıt üzerindeki her türlü çizimin CAD sistemlerine aktarılmasında da aynı yöntem kullanılır. Grafik bilgi sistemlerinde çizgi en temel veri elemanlarından birisidir. Dolayısı ile raster bir görüntüdeki çizgilerin doğru bir şekilde elde edilmesi çok önemlidir (Zhong 2002). Vektörizasyon işleminin başarısı, raster görüntünün kalitesi ile doğru orantılıdır. Görüntüdeki detaylar ne kadar belirgin ise, hatasız bir vektör veri elde etmek o ölçüde mümkün olacaktır (Dori ve Wenyin 1999). Vektörizasyon işleminin ardından görüntüdeki her bir çizginin tek tek seçilmesi, işlenmesi manipüle edilmesi mümkün hale gelir (Nieuwenhuizen et al. 1994). Raster görüntülerdeki çizgilerin elde edilmesine yönelik olarak bugüne kadar bir çok yöntem ve algoritma geliştirilmiştir (Miao vd. 2002, Lagunovsky ve Ablameyko 1999, Madhvanath vd. 1999, Hori ve Tanigawa 1993). Bu yöntemleri 6 sınıfa ayırmak mümkündür: (1) Hough Dönüşümü tabanlı (2) inceltme tabanlı, (3) kontur tabanlı, (4) run-graph tabanlı, (5) mesh pattern tabanlı, ve (6) seyrek piksel tabanlı (Wenyin ve Dori 1999). Tarama, filtreleme ve eşikleme işlemlerinin ardından (HT yöntemi dışında) bu yöntemlerin hepsinde şu üç aşama da mevcuttur: Çizgi İnceltme, çizgi izleme ve zincir kodlama ve vektör azaltma. Çizgi inceltme işlemiyle raster görüntüdeki çizgilerden her biri bir piksel kalınlığına indirilerek iskeleti ortaya çıkarılır. Ardından çizgi izleme işlemi gerçekleştirilerek, her bir çizginin piksel piksel ilerleme yönleri belirlenerek zincir kod üretilir. Son aşamada ise zincir koduna bağlı olarak, raster görüntüye çakışık fakat en az sayıda vektör doğru parçasından oluşacak şekilde vektör doğrular elde edilir (Jennings 1993). Bu çalışmada raster görüntülerin vektörizasyonuna yönelik olarak yeni bir model geliştirilmiştir. Görüntünün değişik yönlerde taranması (analiz edilmesi) yoluyla çizgilerin elde edilmesi temeline dayanan ve MUSCLE (Multidirectional Scanning for Line Extraction) Model adı verilen bu yöntem ile, geleneksel yöntemlerden farklı olarak geliştirilen vektörizasyon algoritması sayesinde, zincir kodlama ve vektör azaltma işlemlerine gerek kalmadan çizgileri elde etmek mümkün olmaktadır. Söz konusu modele ait algoritma bir yanıyla, bir çizgi inceltme algoritmasıdır. Model özellikle doğrusal çizgilerin vektörizasyonunda çok etkilidir. Bu açıdan büyük ölçüde doğrusal çizgilerden oluşan kat planlarından, 3 boyutlu (bina içi) yol ağlarının otomatik olarak elde edilmesi amacıyla modellenen faydalanılmış ve başarılı sonuçlar elde edilmiştir. Bu bildiride söz konusu model ve 3 boyutlu yol ağlarının otomatik olarak elde edilmesi uygulaması ile ilgili detaylar anlatılacaktır.

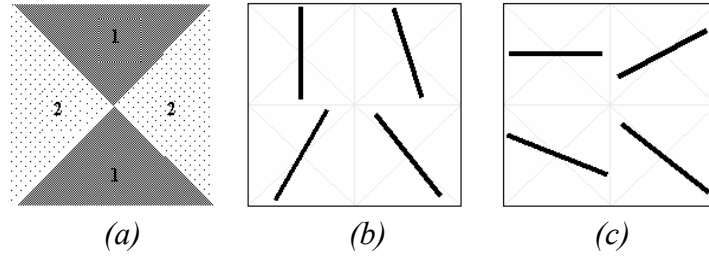
Materyal ve Yöntem

Eşikleme İşlemi

MUSCLE modelin ilk aşamasında, önce raster görüntü eşikleme (threshold) işlemine tabi tutulmakta, belirlenen gray level değerinden daha açık olan pikseller beyaz, daha koyu olanlar siyah olacak şekilde ikili (binary) veri setine dönüştürülmektedir.

Yatay ve Düşey Tarama

Görüntü yatay olarak taranarak düşeye yakın doğrular, düşey olarak taranarak ise yataya yakın doğrular elde edilmektedir. Bu noktada yataya yakın ve düşeye yakın doğrularla neyi kastettiğimizi açıklamaya çalışalım:



Şekil 1. (b) Düşeye yakın doğru örnekleri (c) Yataya yakın doğru örnekleri

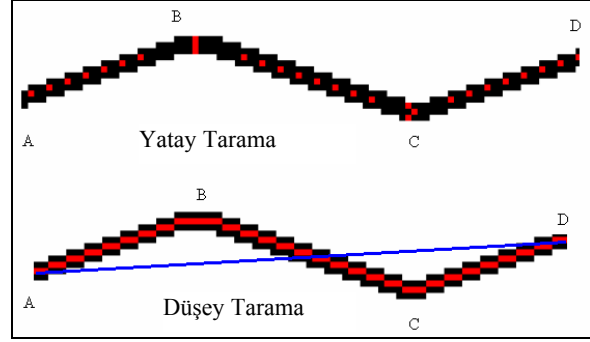
Eğimi -1 ile 1 arasında ($-1 < m < 1$) olan doğrular yataya yakın, -1'den küçük yada 1'den büyük ($m < -1$ ve $m > 1$) olan doğrular ise düşeye yakın doğrular olarak tanımlanmıştır. Dolayısıyla, Şekil 1a için, O merkezini kesmek üzere, 1. bölgeden geçen doğrular düşeye yakın, 2. bölgeden geçen doğrular yataya yakın doğrular olarak tanımlanmıştır. Şekil 1b'de düşeye yakın, 1c'de ise yataya yakın doğru örnekleri çizilmiştir. İlk aşamada, raster veri seti yatay olarak taranarak (satırlar boyunca pikseller üzerinde gidilerek) raster görüntüdeki çizgi kalınlıkları tespit edilmekte ve çizgi kalınlığının orta noktasına düşen pikseller belirlenmektedir. Bu işlem esnasında görüntü üzerinde soldan sağa gidilerek her bir pikselin renk değerine bakılmaktadır. Siyah bir pikselle karşılaşıldığında bunun kaçınıcı piksel olduğu belirlenmektedir. Sonra ilerlemeye devam edilmekte ve sonraki ilk beyaz piksele rastlandığında yine kaçınıcı piksel olduğu belirlenmektedir. Buna göre ilk siyah piksel m ve ilk beyaz piksel n sırasında olmak üzere; $m + \text{Tamdeğer}((n - m) / 2)$ formülü ile çizginin orta noktasındaki pikselin, yeri belirlenmiş olmaktadır. Satır üzerinde devam edilerek diğer çizgiler için de aynı işlemler yapıldıktan ve tüm satırlarda aynı işlemler gerçekleştirildikten sonra, kırmızı piksellerin dağılımı, düşeye yakın doğrular için Şekil 2a, yataya yakın doğrular için ise Şekil 2b'deki gibi olmaktadır. Şekillerde de görüldüğü üzere, kırmızı noktalar, düşeye yakın doğrular için süreklilik gösterirken, yataya yakın doğrular için kesik kesik bir durum arz etmektedir. Bir sonraki aşamada raster veriseti bu kez düşey olarak taranmakta ve aynı işlemler, sütunlar boyunca pikseller üzerinde gidilerek, tekrar gerçekleştirilmektedir. Bu durumda tam tersi bir durum ortaya çıkmakta, Şekil 2c ve 2d'de görüldüğü üzere, kırmızı noktalar, yataya yakın doğrular için süreklilik gösterirken, düşeye yakın doğrular içinse kesintili olarak ortaya çıkmaktadır. Hem yatay hem de düşey taramanın ardından, rasterda sadece çizgi orta noktalarını oluşturan kırmızı pikseller bırakılmakta ve ardından bu pikseller üzerinde komşuluk analizi yapılmaktadır. Bu yöntem ile kırmızı piksele bitişik diğer kırmızı piksel aranmakta ve doğrunun bir ucundan diğer ucuna kadar bu işlem devam etmektedir. Komşu piksel bulunamadığı an doğrunun bitiş noktasına gelindiği anlaşılmaktadır. Bu şekilde her bir kırmızı hattın başlangıç ve bitiş noktaları tespit edilmektedir. Böylece kesintisiz olarak birleşen kırmızı hatlar vektör yapıya dönüşmüş olmakta, yatay taramada sadece düşeye yakın, düşey taramada ise sadece yataya yakın doğrular kesintisiz olarak elde edilerek, bu iki grup bir araya getirildiğinde bütüne ulaşılmaktadır. Şekil 4a'deki örnekte, yatay tarama ile belirlenen kırmızı noktalardan süreklilik arz edenlerinin analizi ile düşeye yakın doğruların vektör olarak nasıl elde edildiği görülmektedir. Şekil 4b ise düşey tarama ile yataya yakın doğruların nasıl elde edildiğini göstermektedir. Bu iki aşamadan sonra elde edilen vektörlerin birleştirilmesi ile ise Şekil 4c'de görülen bütüne ait vektör yapıya ulaşılmaktadır.

Yatay ve düşey tarama ile doğru olarak elde edilemeyen çizgilerin belirlenmesi

Tümü yataya yakın, yada tümü düşeye yakın doğru parçalarının birbiri ardına eklendiği durumlarda yukarıda anlatılan yöntem başarısız olmakta, bu türden doğrular düzgün olarak vektörize edilememektedir.

	Düşeye Yakın Doğrular	Yataya Yakın Doğrular
Yatay Tarama	(a)	(b)
Düşey Tarama	(d)	(c)

Şekil 2. Yatay ve düşey tarama ile kırmızı piksellerin belirlenmesi



Şekil 3. Hatalı vektörizasyonun oluşması

Örneğin Şekil 3'e göre raster verideki AB, BC ve CD doğrularının her üçü de yataya yakın doğru sınıfına girdiğinden yatay tarama yapıldığında, herhangi bir vektör veri elde edilememekte, düşey tarama yapıldığında ise, A'dan D'ye süreklilik arzettiği için kırmızı pikseller üzerinde komşuluk analizi yapıldığında doğrunun ilk noktası A, son noktası D gibi algılanmakta, B ve C köşe noktaları atlanmakta, dolayısı ile hatalı bir vektör doğru ortaya çıkmaktadır. Bu gibi durumlarda çapraz tarama uygulanarak çözüme gidilmektedir. Yatay ve düşey tarama ile vektörize edilmiş, bir raster görüntüdeki bu türden hatalı doğruların varlığı, kırmızı piksellerin, elde edilen vektörlerin doğru denklemlerini sağlayıp sağlamadığına bakılarak belirlenmektedir.

Buna göre örneğin, elde edilen vektör $A(x_a, y_a)$ ve $B(x_b, y_b)$ noktalarının birleştirdiği bir AB doğrusu ise, $(Y - y_a) / (y_a - y_b) = (X - x_a) / (x_a - x_b)$ olduğundan, vektöre ait $Y = mX + n$ formatındaki doğru denklemi;

$$Y = \underbrace{((y_a - y_b) / (x_a - x_b))}_m X + \underbrace{((y_b x_a - x_b y_a) / (x_a - x_b))}_n$$

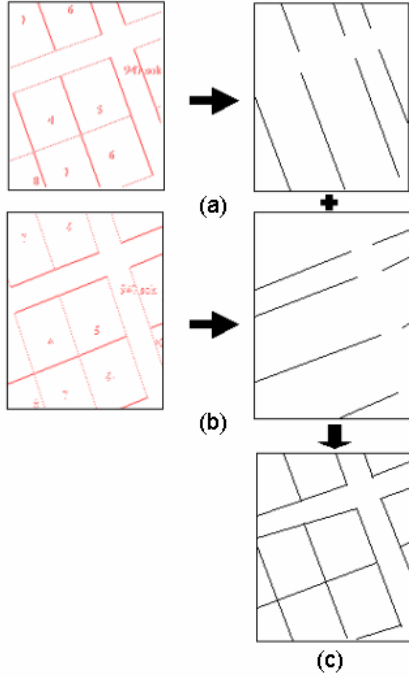
şeklinde dir. Vektör doğrunun bu şekilde hesaplanan denkleminde, o vektörün elde edildiği kırmızı piksellerin koordinatları yerine koyulduğunda, tüm pikseller eşitliği belirli bir limit dahilinde sağlar ise, elde edilen vektörün doğruluğuna, aksi takdirde hatalı olarak elde edildiğine karar verilmektedir. Bu işlemin sonucunda hata sınırı içinde kalan çizgiler tamamen beyaza dönüştürülerek görüntüden çıkarılmakta, raster veri setinde sadece kırmızı piksellerin oluşturduğu hatalı doğrular bırakılmaktadır. Ardından bu yeni raster görüntü çapraz tarama işlemine tabi tutulmaktadır.

Çapraz Tarama ile hatalı doğruların elde edilmesi

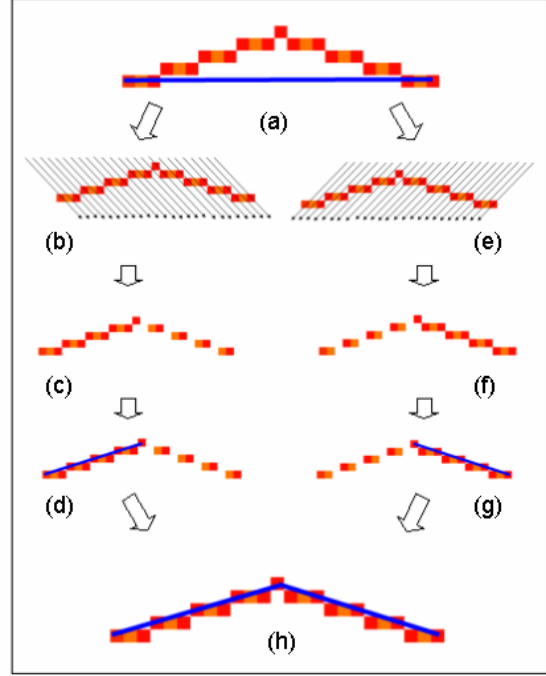
Hatalı doğrulara ait kırmızı piksellerin oluşturduğu raster görüntü 45 derecelik açılarla, önce soldan sağa, ikinci aşamada ise sağdan sola taranmaktadır. Çapraz taramadaki temel espri, tarama yönündeki ardışık piksellerden birinin iptal edilmesi esasına dayanmaktadır. Şekil 5b'daki örneği inceleyecek olursak, görüntü soldan sağa çapraz taranıp, tarama yönünde ardışık piksellerden biri iptal edildiğinde, sağ kanattaki doğru parçası kesikli bir hale dönüşmektedir (Şekil 5c). Ardından komşuluk analizi ile pikseller izlendiğinde sol kanattaki doğru parçasını elde etmek mümkün olmaktadır (Şekil 5d). Komşuluk analizi esnasında belirli bir uzunluktan küçük piksellerden oluşan doğruların gözardı edilmesi ile sağ kanattaki 2'şer pikselden oluşan doğrular dikkate

Doğrusal çizgilerden oluşan raster görüntülerin vektörizasyonu için yeni bir yöntem ve 3 boyutlu CBS'de kullanılması

alınmamaktadır. İkinci aşamada görüntü sağdan sola tarandığında ise aynı yöntemle bu kez sağ kanattaki doğru parçası elde edilmektedir (Şekil 5e, 5f, 5g). Her iki taramanın ardından elde edilen doğru parçaları bir araya getirilerek raster görüntü doğru bir şekilde vektörize edilmiştir (Şekil 5h).



Şekil 4. Yatay ve düşey tarama ile vektör verinin elde edilmesi



Şekil 5. Çapraz taramalarla hatalı doğruların elde edilmesi

Elde edilen vektör veri üzerinde Gerçekleştirilen Topolojik Düzeltmeler

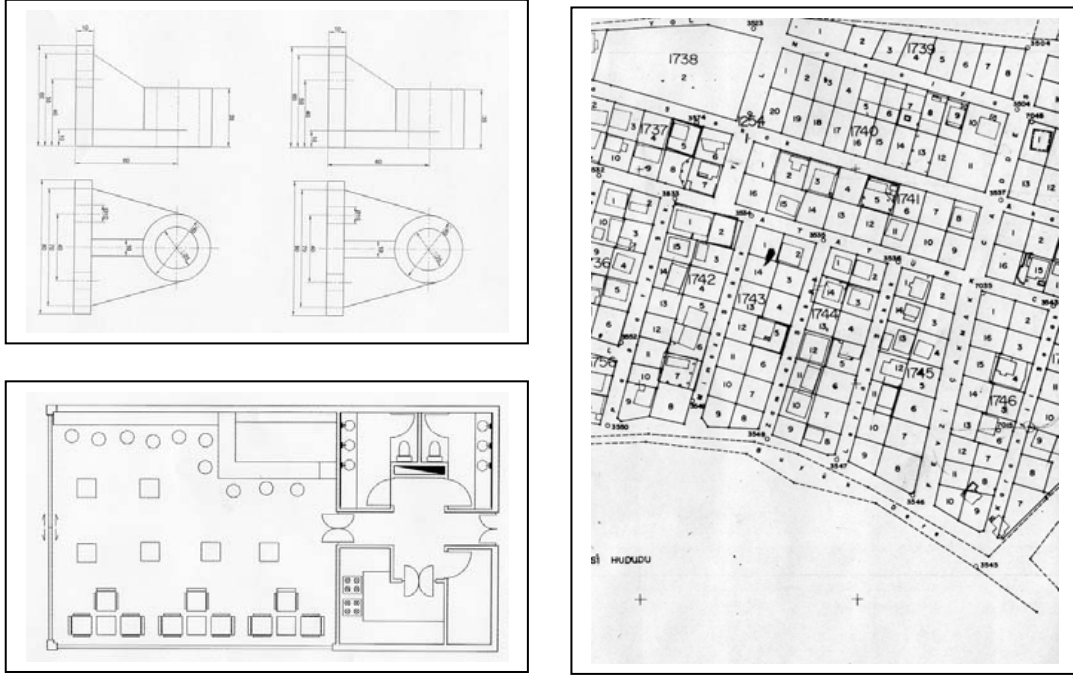
Yatay, düşey ve çapraz taramaların ardından elde edilen vektör veri üzerinde topolojik düzeltmeler gerçekleştirilmektedir. Gerçekleştirilen topolojik düzeltmeler ile doğru parçalarının uç noktalarının bir araya getirilmesi, sarkan ve erişmeyen doğru uçlarının düzeltilmesi, kesişim noktalarının tanımlanması sağlanmaktadır.

Elde edilen vektör yapının kayıt ortamında saklanması

Vektörizasyon sonucu elde edilen sonuç ürün, tüm CAD tabanlı yazılımlar tarafından açılabilen, DXF formatında saklanarak kullanıma hazır hale getirilmektedir. Tarama işlemleri ve topolojik düzeltmelerin, her aşamasında elde edilen her bir vektör yapı, ayrı birer katman olarak saklanmakta ve kullanıcı vektörizasyonun her aşamasını bu sayede inceleyebildiği gibi, ihtiyaçlarına göre bu katmanlardan yararlanabilmektedir.

Gerçekleştirilen testler ve sonuçları

Şekil 6'da görülen 3 görüntü üzerinde MUSCLE model, Wintopo ve Scan2Cad yazılımları ile test edilerek sonuçları karşılaştırılmıştır. Makine parçası, bir restorana ait kat planı ve imar paftasından oluşan A4 formatındaki söz konusu görüntüler 200 dpi da taranmış ve JPG formatında kayıt edilmiştir. Önce raster görüntüdeki, vektörizasyon işleminin ardından vektör verideki çizgiler sayılmıştır ve sonuçlar Tablo 1'de görülmektedir. Tablo 2'de ise, elde edilen sonuçlara bağlı olarak modelin diğer algoritmalarla karşılaştırması yapılmıştır.



Şekil 6. Test Verileri

Tablo 1. Test sonuçları

Görüntü No	Elde Edilen Çizgi Sayısı			Süre		
	1	2	3	1	2	3
Raster Görüntü	835 düz, 21 tireli	130 düz, 50 tireli, 44 noktalı	342 düz			
MUSCLE	2578 düz	602 düz, 72 tireli	351 düz	11 dk 20 sn	14 dk 48 sn	5 dk 22 sn
Wintopo	4160 düz, 20 tireli	653 düz, 20 tireli, 22 noktalı	3183 düz	3 sn	4 sn	2 sn
Scan2cad	1023 düz, 20 tireli	440 düz	320 düz	36 sn	53 sn	22 sn

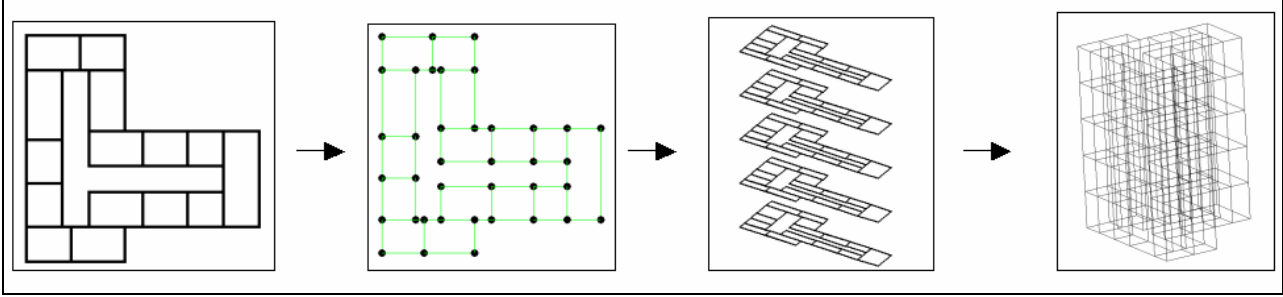
Tablo 2. MUSCLE Model ve diğer algoritmalar

	Avantajları	Dezavantajları
MUSCLE	<ul style="list-style-type: none"> Köşe noktalarında iyi birleştirme yapmaktadır Her bir parça çizgi için yine bir parça vektörleştirme yapmaktadır. Bu da düzelme de büyük zaman avantajı sağlamaktadır. Özellikle genel olarak düz çizgilerden oluşan görüntülerde çok iyi başarı sağlamıştır. 	<ul style="list-style-type: none"> Algoritma benzerlerine göre daha uzun zaman almaktadır. Algoritma her yönde yaptığı taramalarda bazen aynı çizgi için birden fazla çizgi kullanmaktadır bu da düzeltme işleminde zorluğa yol açmaktadır. Algoritma özellikle noktalı çizgilerde başarısız sonuçlar vermektedir. Ayrıca tireli çizgilerde de başarı oranı düşüktür.
Wintopo	<ul style="list-style-type: none"> En hızlı işlem yapan programdır Program dâhilinde hiçbir CAD programına ihtiyaç duymadan düzeltme yapabilmek mümkündür. Program eşikleme değeri istemeden kendi algoritmasıyla gerekli eşikleme değerini belirlemektedir. 	<ul style="list-style-type: none"> Çizgileri benzerlerine göre çok fazla parçaya bölmektedir. Bu da düzeltme işleminde büyük zaman kaybına yol açmaktadır. Özellikle birbirine yakın doğru parçalarında büyük hatalar yapmaktadır. Çapraz çizgi birleştirmesinde hatalar yapmaktadır.
Scan2cad	<ul style="list-style-type: none"> Program dâhilinde hiçbir CAD programına ihtiyaç duymadan düzeltme yapabilmek mümkündür. 	<ul style="list-style-type: none"> Eşikleme seçeneği olmadığı için görüntüler sadece binary yapılarak değerlendirmeye sokulabilmektedir. Birbiriyle çakışan doğrularla bölme yapmadığı için yoğun bir düzeltme işi çıkarmaktadır.

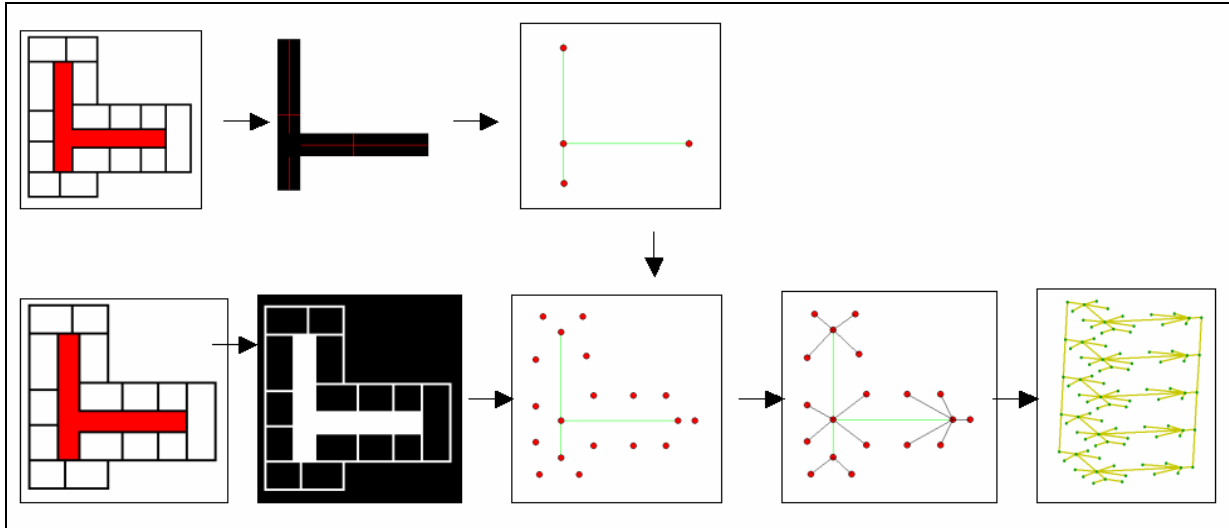
Doğrusal çizgilerden oluşan raster görüntülerin vektörizasyonu için yeni bir yöntem ve 3 boyutlu CBS'de kullanılması

MUSCLE model kullanılarak 3 boyutlu ağ analizleri için veri üretimi

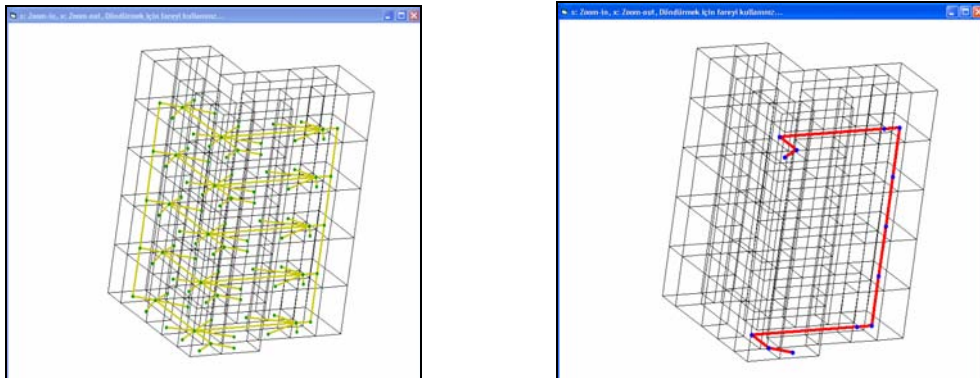
Yapılan testler sonucunda MUSCLE modelin özellikle doğrusal çizgilerin elde edilmesinde oldukça etkili olduğu görülmüştür. Bu yüzden Üç Boyutlu Coğrafi Bilgi Sistemlerinde ağ analizi uygulamaları için veri üretiminde kullanılabileceği düşüncesiyle model uyarlanmıştır. Büyük ölçüde doğrusal çizgilerden oluşan karmaşık ve yüksek binalara ait mimari planlardan, bina (Şekil 7) ve ağ modellerinin (Şekil 8) otomatik olarak elde edilebileceği, böylece 3 boyutlu ağ analizlerinin (Şekil 9) başarıyla gerçekleştirilebileceği gösterilmiştir.



Şekil 7. MUSCLE model kullanılarak 3 boyutlu bina modelinin elde edilmesi



Şekil 8. MUSCLE model kullanılarak 3 boyutlu ağ modelinin elde edilmesi



Şekil 9. Elde edilen bina ve ağ modelleri üzerinde yapılan ağ analizleri (En kısa yol analizi vb.)

Sonuçlar

Bu çalışmada raster görüntülerin vektörizasyonuna yönelik olarak geliştirilen MUSCLE model ile doğrusal çizgilerden oluşan görüntülerin dönüşümünde başarılı sonuçlar elde edilebileceği ortaya konmuştur. Yöntem sadece haritaların (kadastral paftalar, imar planları vs.) değil, kağıt ortamından sayısal hale getirilecek (teknik resim, makina parçaları, mimari çizimler vs.) doğrusal hatlardan oluşan her türlü dökümanın dönüşümünde kullanılabilir.

Test sonuçlarında da görüldüğü üzere yöntem özellikle doğrusal çizgilerinin vektörizasyonunda diğer algoritmalara göre belirgin bir üstünlük göstermektedir. İşlem zamanı açısından bakıldığında diğer algoritmalara göre oldukça yavaş olsa da, bunun sebebi algoritma üzerinde henüz bu anlamda çalışılmamış olmasındandır. Şu ana kadar yapılan çalışmalarda hız değil, doğruluğu yüksek olan maksimum düzeyde çizginin elde edilmesi üzerinde yoğunlaşmıştır. Geliştirilen model, geleneksel yöntemlerden farklı olarak, zincir kodlama ve vektör azaltma işlemlerine gerek kalmadan, sadece basit komşuluk analizi ile vektör çizgileri elde etmeyi mümkün kılması bakımından özgün ve yeni bir yöntemdir. Devam eden çalışmalarla modelin geliştirilmesi, optimizasyon teknikleri kullanılarak performansının artırılması ve ilave modüllerle pratikte de kullanılabilmesi hedeflenmektedir.

Doğrusal çizgileri elde etmedeki başarısının bir sonucu olarak, kat planlarından 3 boyutlu bina ve topolojik ağ modellerinin otomatik olarak elde edilmesi, modelin önemli bir kullanım alanı olarak kendini göstermektedir. Gerçekleştirilen uygulamayla son yılların önemli araştırma alanlarından biri olan 3 boyutlu coğrafi bilgi sistemleri ve 3 boyutlu ağ analizi uygulamalarına da katkı yapıldığı düşünülmektedir. Modelin özellikle bu alanda kullanılması yönünde çalışmalara devam edilecektir.

Kaynaklar

- Dori, D., Wenyin, L. (1999). Automated CAD Conversion with the Machine Drawing Understanding System: Concepts, Algorithms, and Performance. IEEE Transactions On Systems, Man, And Cybernetics - Part A: Systems And Humans, Vol. 29, No.4.
- Hori, O., Tanigawa, S. (1993). Raster-to-vector Conversion by Line Fitting Based on Contours and Skeletons. Proc., Int'l Conf. Document Analysis and Recognition, Tsukuba (Japan), 353-358.
- Jennings, C. (1993). Computer Vision for Line Drawings. MSc Thesis, University of Calgary.
- Lagunovsky, D., Ablameyko, S. (1999). Straight-line-based primitive extraction in grey-scale object Recognition. Pattern Recognition Letters, 20, 1005-1014.
- Miao, L., Liu, X., Peng, Q., Bao, H. (2002). BRDC: binary representation of displacement code for line. Computers & Graphics, 26, 401-408.
- Nieuwenhuizen, P.R., Kiewiet, O., Bronsvort, W.F. (1994). An Integrated Line Tracking and Vectorization Algorithm, Eurographics'94, 3(3), 349-359
- Wenyin, L., Dori, D. (1999). From Raster to Vectors: Extracting Visual Information from Line Drawings. Pattern Analysis and Applications, 2(2), 10-21.
- Zhong, D. X. (2002). Extraction of embedded and/or line-touching character-like objects, Pattern Recognition, 35 2453 – 2466.