

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

İsmail Rakıp KARAŞ<sup>1\*</sup>, Bülent BAYRAM<sup>2</sup>, Fatmagül BATUK<sup>2</sup>, Uğur ACAR<sup>2</sup>, Melis UZAR<sup>2</sup>

<sup>1</sup> Gebze Yüksek Teknoloji Enstitüsü, Jeodezi ve Fotogrametri Mühendisliği Bölümü, 41400, Gebze, Kocaeli

<sup>2</sup> YTÜ, İnşaat Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümü, 34349, Yıldız, Beşiktaş, İstanbul

## Özet

Bu çalışmada raster görüntülerin vektörizasyonuna yönelik olarak yeni bir model geliştirilmiştir. MUSCLE (Multidirectional Scanning for Line Extraction) Model adı verilen ve klasik yöntemlerden farklı olarak geliştirilen model, görüntünün değişik yönlerde taranması (analiz edilmesi) yoluyla çizgilerin elde edilmesi temeline dayanmaktadır. Söz konusu vektörizasyon algoritması sayesinde, zincir kodlama ve vektör azaltma işlemlerine gerek kalmadan, sadece basit komşuluk analizi ile çizgileri vektör formatında elde etmek mümkün olmaktadır. Algoritma bir yanıyla, bir çizgi inceltme algoritmasıdır. Model özellikle doğrusal çizgilerde 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, model ile farklı yazılımların sonuçları karşılaştırılmıştır. Bunların dışında model üzerinde çalışılarak, başka alanlarda da kullanılabilirliği görülmüştür. Bunlardan biri üç boyutlu coğrafi bilgi sistemlerinde ağ analizi uygulamalarıdır. MUSCLE Model kullanılarak binalara ait mimari planlardan, geometrik ve topolojik bina modellerinin otomatik olarak elde edilebileceği, böylece üç boyutlu ağ analizlerinin başarıyla gerçekleştirilebileceği gösterilmiştir. Bu bildiride modelin ayrıntılarından bahsedilecek ve sözkonusu uygulamalar tanıtılacaktır.

**Anahtar Kelimeler:** Vektörizasyon, Raster-vektör dönüşümü, Çizgi inceltme, Üç boyutlu coğrafi bilgi sistemleri.

## 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 buildings and perform 3D network analysis successfully using the MUSCLE models.

**Keywords:** Vectorization, Raster to vector conversion, Line thinning, 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 toplamanın toplam maliyete oranı % 60-80 düzeyindedir. Bu yüzden bir CBS'nin ihtiyacı olan veriyi mevcut verilerden yararlanarak elde etmek sıkça başvurulmuş bir yöntemdir. 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 ve işlenmesi mümkün hale gelir (Nieuwenhuizen vd. 1994).

Raster görüntülerdeki çizgilerin elde edilmesine yönelik olarak bugüne kadar bir çok yöntem ve algoritma geliştirilmiştir (Miao vd. 2002, 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, (6) seyrek piksel tabanlı (Wenyin ve Dori 1999). Birincisi hariç bu yöntemlerin hepsinde de çizgi inceltme, çizgi izleme-zincir kodlama ve vektör azaltma aşamaları mevcuttur.

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 görüntüdeki çizgilere çakışık fakat en az sayıda 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, klasik yöntemlerden farklı olarak geliştirilen vektörizasyon algoritması sayesinde, çizgi izleme-zincir kodlama ve vektör azaltma işlemlerine gerek kalmadan, sadece basit komşuluk analizi ile çizgileri elde etmek mümkün olmaktadır. Söz konusu modele ait algoritma bir yanı sıra, bir çizgi inceltme algoritmasıdır. Model özellikle doğrusal çizgilerin vektörizasyonunda oldukça etkilidir. Bu açıdan büyük ölçüde doğrusal çizgilerden oluşan kat planlarından, üç boyutlu (bina içi) yol ağlarının otomatik olarak elde edilmesi amacıyla modelden faydalanılmış ve başarılı sonuçlar elde edilmiştir. Bu bildiride söz konusu model ve üç boyutlu uygulama ile ilgili detaylar sunulmuştur.

## **Materyal ve Yöntem**

### **Eşikleme İşlemi**

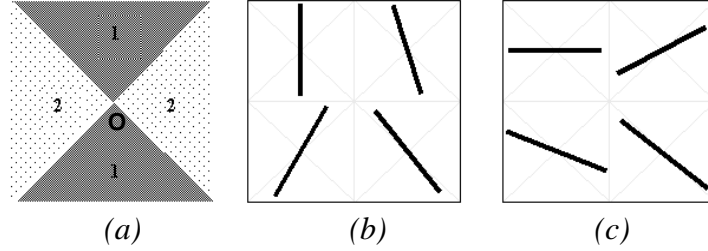
MUSCLE modelin ilk aşamasında, önce raster görüntü eşikleme (threshold) işlemine tabi tutulmakta, belirlenen gri değerden daha açık olan pikseller beyaz, daha koyu olanlar siyah olacak şekilde ikili (binary) veri setine dönüştürülmektedir.

### **Yatay ve Düşey Tarama**

Görüntü yatay olarak taranarak düşeye yakın doğrular, düşey olarak taranarak ise yataya yakın doğrular elde edilmektedir. Eğimi  $-1$  ile  $1$  arasında ( $-1 < m < 1$ ) olan doğrular yataya yakın,  $-1$ 'den

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

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, birinci bölgeden geçen doğrular düşeye yakın, ikinci bölgeden geçen doğrular yataya yakın doğrular olarak tanımlanmıştır. Şekil 1b'de düşeye yakın, 1c'de ise yataya yakın doğru örnekleri çizilmiştir.

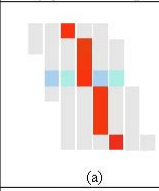
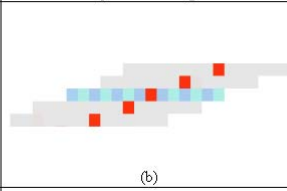
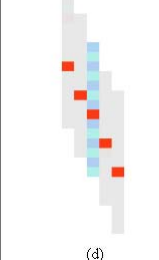
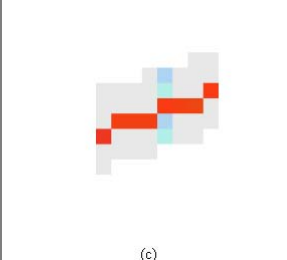


Şekil 1. (a) Bölgeler (b) Düşeye yakın doğru örnekleri (c) Yataya yakın doğru örnekleri

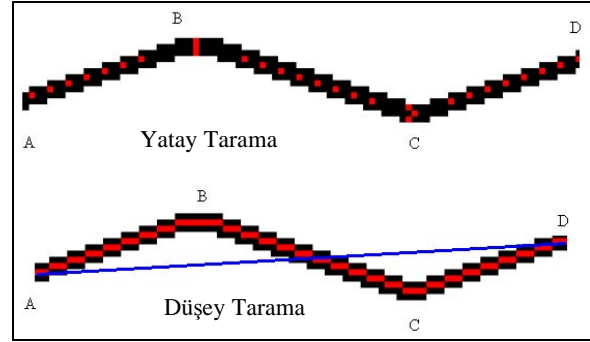
İlk aşamada, raster veri seti yatay olarak taranarak (satırlar boyunca pikseller üzerinde gidilerek) raster görüntüdeki çizgi kalınlıkları tespit edilmekte ve çizgi kalınlığının orta noktasına düşen pikseller belirlenmektedir. Bu işlem sırasında görüntü üzerinde soldan sağa gidilerek her bir pikselin 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; " $m + \text{tam değer}((n - m) / 2)$ " formülü ile çizginin orta noktasındaki pikselin yeri belirlenmiş olmaktadır ve bu piksel kırmızıya dönüştürülmektedir. 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 pikseller, düşeye yakın doğrular için süreklilik gösterirken, yataya yakın doğrular için kesintili bir durum arz etmektedir. Bir sonraki aşamada raster veri seti 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 pikseller, yataya yakın doğrular için süreklilik gösterirken, düşeye yakın doğrular içinse kesintili olarak ortaya çıkmaktadır. Hem yatay hem de düşey taramanın ardından, raster görüntüde 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'daki örnekte, yatay tarama ile belirlenen kırmızı noktalardan süreklilik arzedenlerinin 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 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		
Düşey Tarama		

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 mavi ile çizilmiş olan, hatalı vektör doğru ortaya çıkmaktadır. Bu gibi durumlarda önce hatalı doğrular belirlenmekte, ardından 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 elde edilen vektör  $A(x_a, y_a)$  ve  $B(x_b, y_b)$  noktalarının birleştirdiği bir AB doğrusu ise,  $(Y - y_a) / (y_a - y_b) = (X - x_a) / (x_a - x_b)$  olacağından, vektöre ait  $Y = mX + n$  formatındaki doğru denklemi;

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

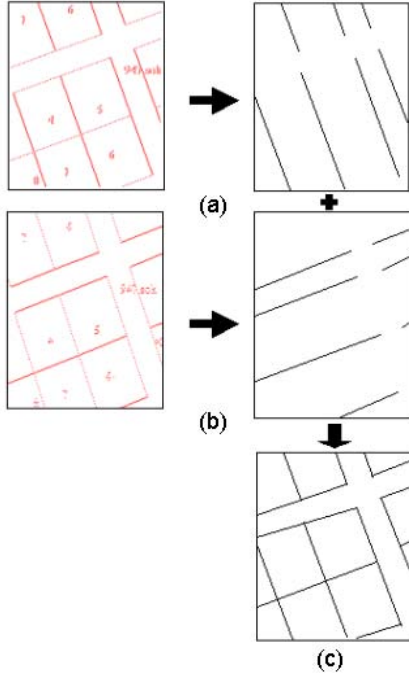
şeklinde. Vektör doğrunun bu şekilde hesaplanan denkleminde, o vektörün elde edildiği kırmızı piksellerin koordinatları yerine koyulduğunda, tüm pikseller eşitliği belirli bir limit dahilinde sağlar ise, elde edilen vektörün doğruluğuna, aksi takdirde hatalı olarak elde edildiğine karar verilmektedir. 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

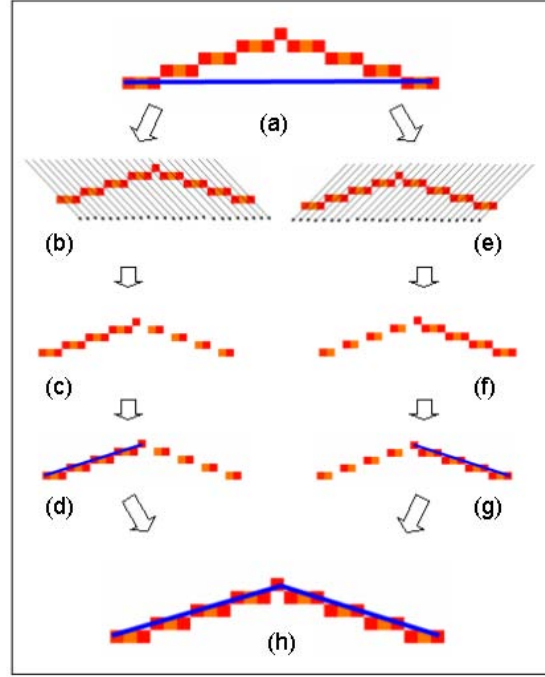
Hatalı doğrulara ait kırmızı piksellerin oluşturduğu raster görüntü 45 derecelik açılarla, önce soldan sağa, ikinci aşamada ise sağdan sola taranmaktadır. Çapraz taramadaki temel yaklaşım, tarama yönündeki ardışık piksellerden birinin iptal edilmesi esasına dayanmaktadır. Şekil 5b'deki örneği inceleyecek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

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

doğruların gözardı edilmesi ile sağ taraftaki ikişer pikselden oluşan doğrular dikkate alınmamaktadır. İkinci aşamada görüntü sağdan sola tarandığında ise aynı yöntemle bu kez sağ taraftaki 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 dikey tarama ile vektör verinin elde edilmesi



Şekil 5. Çapraz taramalarla hatalı doğruların elde edilmesi

### Elde edilen vektör veri üzerinde gerçekleştirilen topolojik düzeltmeler

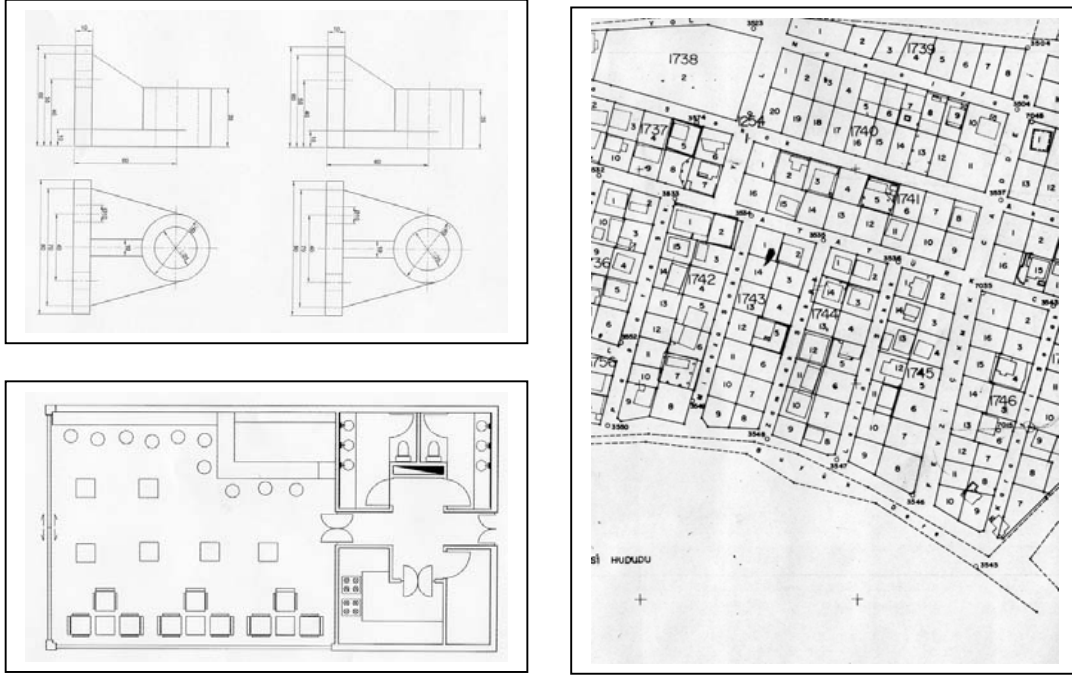
Yatay, dikey ve çapraz taramaların ardından elde edilen vektör veri üzerinde topolojik düzeltmeler gerçekleştirilmektedir. Söz konusu düzeltmeler ile doğru parçalarının uç noktalarının bir araya getirilmesi, sarkan ve erişmeyen doğru uçlarının düzeltilmesi, kesişim noktalarının tanımlanması sağlanmaktadır.

### Elde edilen vektör yapının kayıt ortamında saklanması

Vektörizasyon sonucu elde edilen sonuç ürün, tüm CAD tabanlı yazılımlar tarafından okunabilen DXF formatında saklanarak kullanıma hazır hale getirilmektedir. Tarama işlemleri ve 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 MUSCLE model, Wintopo ve Scan2Cad yazılımları ile test edilerek sonuçları karşılaştırılmıştır. Makine parçası, bir restorana ait kat planı ve imar paftasından oluşan A4 formatındaki sözkonusu görüntüler 200 dpi ile taranmış ve JPG formatında kayıt edilmiştir. Önce raster görüntüdeki, vektörizasyon işleminden sonra da vektör verideki çizgiler sayılmıştır (Tablo 1). Elde edilen sonuçlara bağlı olarak modelin diğer algoritmalarla karşılaştırması yapılmıştır ve sonuçlar Tablo 2'de görülmektedir.



Şekil 6. Test Verileri

Tablo 1. Test sonuçları

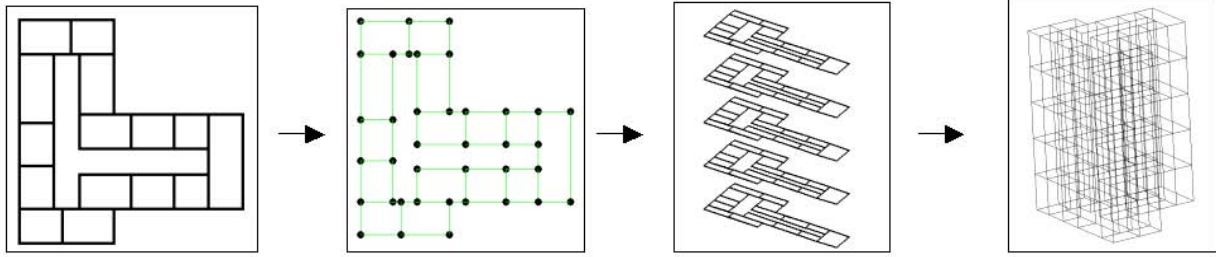
Görüntü No	Elde Edilen Çizgi Sayısı			Süre		
	1	2	3	1	2	3
<b>Raster Görüntü</b>	835 düz, 21 kesikli	130 düz, 50 kesikli, 44 noktalı	342 düz			
<b>MUSCLE</b>	2578 düz	602 düz, 72 kesikli	351 düz	11 dk 20 sn	14 dk 48 sn	5 dk 22 sn
<b>Wintopo</b>	4160 düz, 20 kesikli	653 düz, 20 kesikli, 22 noktalı	3183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

Tablo 2. MUSCLE Model ve diğer algoritmalar

	Avantajları	Dezavantajları
<b>MUSCLE</b>	<ul style="list-style-type: none"> <li>Köşe noktalarında iyi birleştirme yapmaktadır</li> <li>Her bir parça çizgi için yine bir parça vektör oluşmaktadır. Bu durum düzeltme sürecinde büyük zaman avantajı sağlamaktadır.</li> <li>Özellikle düz çizgilerden oluşan görüntülerde çok iyi başarı sağlamıştır.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Algoritma özellikle noktalı çizgilerde başarısız sonuçlar vermektedir. Ayrıca kesikli çizgilerde de başarı oranı düşüktür.</li> </ul>
<b>Wintopo</b>	<ul style="list-style-type: none"> <li>En hızlı işlem yapan programdır</li> <li>Program dâhilinde hiçbir CAD programına ihtiyaç duymadan düzeltme yapabilmek mümkündür.</li> <li>Program eşik değeri istemeden kendi algoritmasıyla gerekli değeri belirlemektedir.</li> </ul>	<ul style="list-style-type: none"> <li>Çizgileri benzerlerine göre çok fazla parçaya bölmektedir. Bu da düzeltme işleminde büyük zaman kaybına yol açmaktadır.</li> <li>Özellikle birbirine yakın doğru parçalarında büyük hatalar yapmaktadır.</li> <li>Çapraz çizgi birleştirmesinde hatalar yapmaktadır.</li> </ul>
<b>Scan2cad</b>	<ul style="list-style-type: none"> <li>Program dâhilinde hiçbir CAD programına ihtiyaç duymadan düzeltme yapabilmek mümkündür.</li> </ul>	<ul style="list-style-type: none"> <li>Eşik değer belirleme seçeneği olmadığı için görüntüler sadece binary yapılarak değerlendirmeye sokulabilmektedir.</li> <li>Birbiriyle çakışan doğrularda bölme yapmadığı için yoğun bir düzeltme işi çıkarmaktadır.</li> </ul>

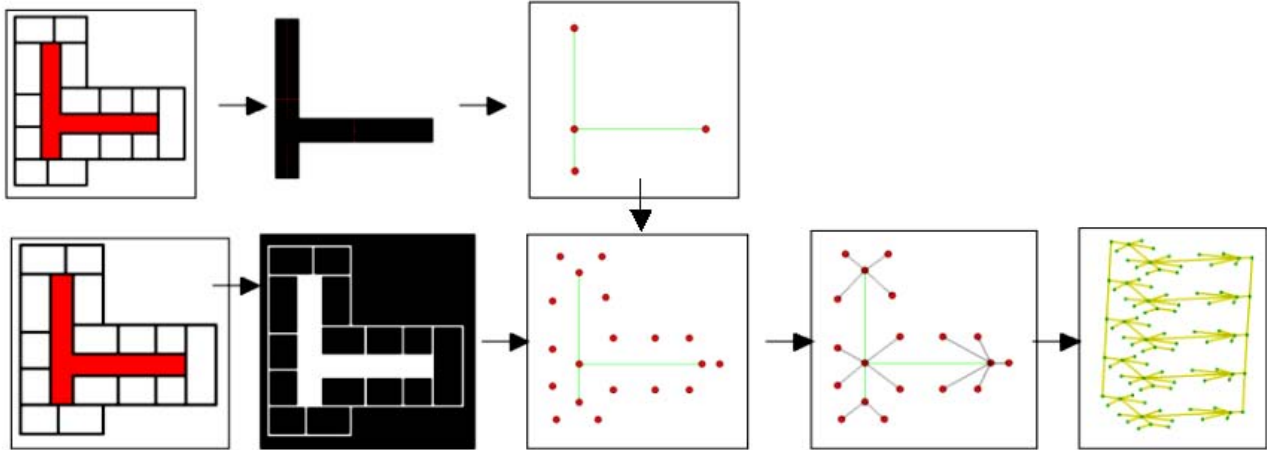
### **MUSCLE model kullanılarak 3 boyutlu ağ analizleri için veri üretimi**

Yapılan testlerde de görüldüğü gibi MUSCLE model özellikle doğrusal çizgilerin elde edilmesinde oldukça etkilidir. Dolayısı ile, modelin kullanımıyla, büyük ölçüde doğrusal çizgilerden oluşan mimari planlardan bina ve ağ modellerinin otomatik olarak elde edilmesi mümkündür. Bu düşünceyle MUSCLE modeli temel alan, üç boyutlu ağ analizi uygulamalarına yönelik bir veri üretim sistemi tasarlanmıştır (Karas vd., 2006). Sistemde ilk aşamada bina modeli elde edilmektedir. Şekil 7'de görüldüğü üzere kat planı, model uygulanarak işlenmekte ve vektör formata dönüştürülmektedir. Ardından kullanıcı tarafından girilmiş olan kat adedi, kat yüksekliği gibi bilgilere bağlı olarak, üç boyutlu bina modeli üretilmektedir.



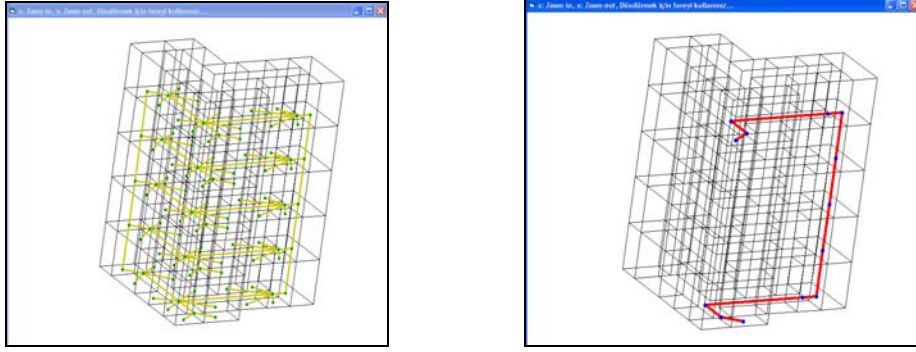
*Şekil 7. MUSCLE model kullanılarak üç boyutlu bina modelinin elde edilmesi*

İkinci aşamada ise ağ modeli üretilmektedir. Odaların binadaki diğer tüm birimlerle bağlantısını sağladığı için kat planında koridor, ana omurgadır. Dolayısı ile koridorun belirlenmesi ve modellenmesi oldukça önem arz etmektedir. Kullanıcı tarafından koridor gösterildikten sonra, algoritma görüntüde sadece koridoru bırakmakta, ardından MUSCLE model kullanılarak koridor orta hattı belirlenmekte, böylece koridorun geometrik ve topolojik modeli elde edilmektedir. Ardından bu kez görüntüden koridor çıkarılmakta, sadece odalar bırakılmaktadır ve modelin uygulanması ile oda orta noktaları birer düğüm olarak elde edilmektedir. Koridor ve odaların kullanıcı etkileşimli olarak birleştirilmesinin ardından, bina modelinde olduğu gibi kat bilgilerine bağlı olarak üç boyutlu ağ modeli üretilmektedir (Şekil 8).



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

Sonuç olarak, bu uygulamayla elde edilen bina ve ağ modelleri üzerinde (en kısa yol vb.) üç boyutlu ağ analizlerini gerçekleştirmek mümkün olmaktadır (Şekil 9). Böylece MUSCLE model kullanılarak tasarlanan sistem sayesinde modelin uygulamada da etkin ve başarılı bir şekilde kullanılabileceği gösterilmiştir.



Şekil 9. Elde edilen bina ve ağ modelleri üzerinde yapılan ağ analizleri

## 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 vb.) değil, kağıt ortamından sayısal hale getirilecek (teknik resim, makina parçaları, mimari çizimler vb.) doğrusal hatlardan oluşan her türlü dökümanın dönüşümünde kullanılabilir. Test sonuçlarında görüldüğü üzere yöntem özellikle doğrusal çizgilerinin vektörizasyonunda diğer algoritmalara göre belirgin bir üstünlük göstermektedir. İşlem zamanı açısından bakıldığında diğer algoritmalara göre oldukça yavaştır. Bunun nedeni ise 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, klasik 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 üç 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 üç boyutlu coğrafi bilgi sistemleri ve üç boyutlu ağ analizi uygulamalarına da katkı yapıldığı düşünülmektedir. Modelin özellikle bu alanda kullanılması yönünde çalışmalara devam edilecektir.

## Kaynaklar

- Dori, D., Wenyn, L. (1999). Automated CAD Conversion with the Machine Drawing Understanding Syst: Concepts, Algorithms, Performance. IEEE Trans. Sys., Man & Cybernetics – Part A, 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.
- Karas, I. R., Batuk, F., Akay, A. E., Baz, I. (2006). *Automatically Extracting 3D Models and Network Analysis for Indoors* in Abdul-Rahman, A., Zlatanova, S., Coors, V., eds, *Innovations in 3D Geoinformation Science (Lecture Notes in Geoinformation and Cartography Series)*, Springer Verlag, 395 - 404, Berlin
- 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
- Wenyn, 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.