



**MUSTAFA KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI**

**GERÇEK GÖRÜNTÜDEN ELDE EDİLEN KOORDİNATLARLA
ROBOT KOL HAREKET OPTİMİZASYONU**

MURAT KERT

YÜKSEK LİSANS TEZİ

ANTAKYA

ŞUBAT-2006

Mustafa Kemal Üniversitesi

Fen Bilimleri Enstitüsü Müdürlüğüne,

Yrd. Doç. Dr. Ersin ÖZDEMİR danışmanlığında, **Murat KERT** tarafından hazırlanan bu çalışma 10.02.2005 tarihinde aşağıdaki jüri tarafından, **Elektrik-Elektronik Mühendisliği** Anabilim Dalında **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Başkan: Yrd.Doç.Dr. Ersin ÖZDEMİR

İmza.....

Üye : Yrd.Doç.Dr. Mustafa ORAL

İmza.....

Üye : Yrd.Doç.Dr. Ali Naci ÇELİK

İmza.....

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Kod No:

İmza
10.02.2006

Prof. Dr. Abdurrahman YİĞİT
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

	Sayfa
ÖZET	I
ABSTRACT.....	II
ÖNSÖZ	III
ÇİZELGELER DİZİNİ	IV
ŞEKİLLER DİZİNİ	V
SİMGELER VE KISALTMALAR.....	VII
1. GİRİŞ	1
2. ÖNCEKİ ÇALIŞMALAR	4
2.1. Derinlik Bulma Tekniklerine Giriş	4
2.1.1. Tek Kamera İle Derinlik Bulma.....	4
2.2. GA Araştırmalarının Tarihi Gelişimi.....	8
2.3. Görüntü İşleme Tekniklerinde GA Kullanılmasının Tarihi Gelişimi	10
2.4. Robotlarda GA Kullanılmasının Tarihi Gelişimi.....	11
3. MATERYAL VE YÖNTEM	13
3.1. Materyal	13
3.2. Yöntem.....	13
3.2.1. Görüntü İşleme ve Çalışmada Kullanılan Görüntü İşleme Teknikleri	14
3.2.1.1. Gri Seviye Resim	15
3.2.1.2. Eşikleme.....	16
3.2.1.3. Yapısal Eleman	17
3.2.1.4. Erozyon	18
3.2.1.5. Sinc Filtreleme	21
3.2.1.6. Sobel Filtreleme	21
3.2.1.7. Genişleme	23
3.2.1.8. Etiketleme	25
3.2.1.9. Engellerin Tespiti.....	27
3.2.1.9.1. Eğim ile Engellerin Tespiti	27
3.2.1.9.2. Çevre ve Alandan Engel Tespiti	29
3.2.1.10. Dairesel Kesitli Nesneleri Etiketleme ve 2 boyutlu Koordinat Bulma	29
3.2.1.11. Renk Sayımı ile Derinlik Bulma.....	30

3.2.2. Genetik Algoritma	33
3.2.2.1. GA'nın Akış Diyagramı.....	33
3.2.2.2. Hazırlık	35
3.2.2.3. Hesaplama ve Uygunluk fonksiyonu	37
3.2.2.4. Seçim.....	38
3.2.2.5. Çaprazlama	40
3.2.2.6. Mutasyon.....	42
3.2.2.7. Elitizm.....	43
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	45
4.1. Test Bölgesi Seçim Analizi.....	45
4.2. Dokuz Nesne İçin Derinlik Analizi.....	47
4.3. Dairesel Kesitli Nesneler Arasında Engel Tanıma	54
4.4. Genetik Algoritma Test sonuçları	59
5. SONUÇ VE ÖNERİLER.....	65
KAYNAKLAR	66
ÖZGEÇMİŞ	70

ÖZET**GERÇEK GÖRÜNTÜDEN ELDE EDİLEN KOORDİNATLARLA
ROBOT KOL HAREKET OPTİMİZASYONU**

Bu çalışmada, deney düzeneğinden tek kamera ile alınan sayısal görüntülere, görüntü işleme teknikleri uygulanarak, dairesel kesitli nesneler ile bu nesneler arasındaki engellerin konumlarının bulunması ve nesnelere erişim sırasının optimizasyonu gerçekleştirilmiştir.

Deney düzeneğinden farklı odak uzaklıklarında çekilen fotoğraflar önce gri seviye resimlere dönüştürülüp ardından uygun bir eşik değeri ile threshold'lanarak ikilik (binary) resimlere dönüştürülmüştür. Binary resimler iki aşamada incelenmiştir. Birinci aşamada, resim içindeki engeller ile dairesel kesitli nesnelerin ayrılması ve engellerin koordinatlarının bulunması için iki ayrı yöntem uygulanmıştır. Birinci yöntemde binary resimlere Sobel x ve Sobel y filtreleri uygulanmış, nesnelerin kenarları tespit edilerek eğimleri bulunmuştur. Eğim değerlerine göre engeller ile dairesel kesitli nesneler birbirinden ayrılmıştır. İkinci yöntemde ise nesnelerin çevre alan bağıntısından yola çıkılarak engeller ile dairesel kesitli nesneler birbirinden ayrılmıştır. İkinci aşamada ise engellerin koordinatlarına bağlı olarak, binary resimler üzerinde engeller zemin ile aynı renge boyanarak sadece dairesel kesitli nesnelerin kalması sağlanmıştır. Dairesel kesitli nesneler etiketlenerek sayıları bulunmuş ve ağırlık merkezlerinin iki boyutlu koordinatları (x, y) tespit edilmiştir. Bu koordinatlara göre nesnenin kenarlarından taşmayacak şekilde bir test bölgesi seçilmiş ve bu bölgede renk sayımları yapılarak dijital fotoğraf makinesinin odak uzaklıklarına göre mesafeleri tespit edilmiştir.

Görüntü işleme sonucu elde edilen nesnelerin üç boyutlu koordinatları, soruna özgü geliştirilen Genetik Algoritma programına aktararak, robot kolun nesnelere erişim sırasının optimizasyonu sağlanmıştır.

2006, 70 Sayfa

Anahtar Kelimeler: Görüntü işleme, Genetik algoritma, Renk sayımı, Mesafe, Engel

ABSTRACT**OPTIMIZATION OF THE ROUTE FOR A ROBOT ARM
BY CAPTURED COORDINATE FROM A REAL IMAGE**

The aim of this study is to localize circular and non-circular objects within an image, taken from an experimental set-up with a single camera, by using image processing, together with route optimization of collecting route of the objects.

The images of a scene that is taken from the experimental setup with varying focus distances of the same scene have been segmented by applying thresholding algorithm with a suitable threshold value. The identification of circular and non-circular objects (for this study, fruits and branches, respectively) have been achieved successively. Two different approaches have been tested for the identification of circular objects. Evaluation of the edge derivatives of the binary objects that was obtained by using Sobel x and Sobel y filters, enabled successful identification of the objects. The other approach that employed compactness value -that is the ratio between the perimeter and the area of an object- yielded successful classifications as well. Non-circular objects were deleted from the scene by assigning the same color value of the background to the non-circular objects. The resulting image contains only the circular objects. After labeling the objects, the center of mass coordinates and the number of circular objects were calculated. An area of interest within a circular object was identified and the number of color shades within this region were counted to calculate the distance of the object from the camera lens.

Three dimensional coordinates of the circular objects that were obtained through image processing were fed into a software that uses a Genetic Algorithms (GAs) that was constructed specially for this task. The GAs produces an optimized route for a robot arm to collect each circular object by avoiding any obstacle.

2006, 70 Page

Key Words: Image processing, Genetic algorithms, Colour count, Distance, Obstacle Detection

ÖNSÖZ

Yüksek Lisans tez konumun belirlenmesinde ve çalışmamın her aşamasında yardımlarını esirgemeyen danışmanım Yrd. Doç. Dr. Ersin ÖZDEMİR'e, bölüm başkanımız Yrd. Doç. Dr. Emin ÜNAL'a ve Yrd. Doç. Dr. Mustafa ORAL'a çok teşekkür ederim. Ayrıca manevi katkılarından dolayı eşime ve beni sabırla bekleyen kızıma teşekkür ederim.

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1. Nesnelere ait eğim değerleri	28
Çizelge 3.2. Farklı odak mesafelerinde çekilen fotoğraflar ve mesafeleri.....	32
Çizelge 3.3. Şehir isimleri düzeninde çözüm grubu	35
Çizelge 3.4. İkilik düzende çözüm grubu	35
Çizelge 3.5. Görüntü işleme sonucu GA'ya aktarılan dairesel kesitli nesnelerin x,y,z koordinatları	36
Çizelge 3.6. Kromozom yapısı.....	36
Çizelge 3.7. On tane örnek kromozom	37
Çizelge 3.8. Örnek Kromozomlar ve uygunluk değeri yüzdeleri	39
Çizelge 4.1. Dokuz nesne için farklı odak mesafelerinde çekilen fotoğraflar ve mesafeleri	53
Çizelge 4.2. Fotoğrafların bulunduğu aralıklara göre mesafeleri	54
Çizelge 4.3. Nesnelerin bulunan mesafeleri ve gerçek mesafe karşılaştırması	54
Çizelge 4.4. Sobel x filtreleme sonucu oluşan resimdeki nesnelerin eğim değerleri.....	57
Çizelge 4.5. Sobel y filtreleme sonucu oluşan resimdeki nesnelerin eğim değerleri.....	58
Çizelge 4.6. Engeller ve koordinatları	59
Çizelge 4.7. Kırksekiz nesne için GA'nın kontrolü ve kontrolsüz oranları	62

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. a) Göz merceği b) Fotoğraf makinesi	1
Şekil 1.2. Beyin ve yapay zeka yöntemleri.....	2
Şekil 2.1. Kameranin yatay düzlem üzerinde hareketi.....	5
Şekil 2.2. Kıyas ile derinlik parametreleri	5
Şekil 2.3. Optik lens ve imajın oluşması.....	6
Şekil 2.4. Lazer ile derinlik.....	7
Şekil 3.1. Deney düzeneği	13
Şekil 3.2. Koordinatlarının (x, y, z) bulunması için akış diyagramı.....	14
Şekil 3.3. Renkli resim.....	15
Şekil 3.4. Gri seviye resim.....	15
Şekil 3.5. Binary resim.....	16
Şekil 3.6. Gri seviye resmin histogramı ve thresholding değeri	17
Şekil 3.7. a)Kare, b)Üçgen, c)Dikey dikdörtgen, d)Yatay dikdörtgen yapısal eleman ..	18
Şekil 3.8. a) Erozyon öncesi b) Erozyon sonrası	19
Şekil 3.9. 3x3 Kare yapısal elemanı ile erozyon işlemi	20
Şekil 3.10. Sinc filtre	21
Şekil 3.11. Sinc filtreleme sonrası	21
Şekil 3.12. Sobel x ve Sobel y filtreleri	22
Şekil 3.13. 3x3 şeklinde örnek piksel komşuluğu.....	22
Şekil 3.14. a) Sobel x sonrası b) Sobel y sonrası	22
Şekil 3.15. Dilation öncesi	23
Şekil 3.16. Dilation sonrası	23
Şekil 3.17. 3x3 Kare yapısal elemanı ile dilation işlemi.....	24
Şekil 3.18. a) Etiketlenmiş resim b) Etiketleme sırası	26
Şekil 3.19. a) Sobel x, b) Sinc filtrelerinin etiketleme sonrası	26
Şekil 3.20. Örnek bir doğru parçası	27
Şekil 3.21. Dairesel kesitli nesnelerin etiketlenmesi	30
Şekil 3.22. Nesne ve seçilen test bölgesi	31
Şekil 3.23. Farklı odak mesafelerinde çekilen resimler ve renk sayım grafiği.....	32
Şekil 3.24. GA'nın akış diyagramı	34
Şekil 3.25. Rulet tekeri modeli	39
Şekil 3.26. a) Tek noktalı çaprazlama yapılacak kromozomlar, b) Tek noktalı çaprazlama sonucu oluşan kromozomlar	40
Şekil 3.27. Çaprazlama yapılacak kromozomlar	41
Şekil 3.28. Tek noktalı çaprazlama sonucu oluşan kromozomlar.....	41
Şekil 3.29. Tamir operatörü ile düzeltilmiş kromozomlar	42
Şekil 3.30. a) İki noktalı çaprazlama yapılacak kromozomlar b) Çaprazlamadan sonra oluşan kromozomlar.....	42
Şekil 3.31. a) Mutasyon öncesi b) Mutasyon sonrası	43
Şekil 3.32. Elitizm eklenmiş GA algoritması	44
Şekil 4.1. İç ve dış test bölgesi.....	46
Şekil 4.2. İç ve dış test bölgelerinin renk sayım karşılaştırmalı grafiği.....	46

Şekil 4.3. 20 cm mesafedeki nesnenin 800x600 ve 640x480 piksellerdeki renk sayım grafikleri.....	47
Şekil 4.4. 50 cm mesafedeki nesnenin 800x600 ve 640x480 piksellerdeki renk sayım grafikleri.....	47
Şekil 4.5. Dokuz nesneli örnek resim	48
Şekil 4.6. Birinci nesne renk sayım grafiği.....	48
Şekil 4.7. İkinci nesne renk sayım grafiği.....	49
Şekil 4.8. Üçüncü nesne renk sayım grafiği.....	49
Şekil 4.9. Dördüncü nesne renk sayım grafiği.....	50
Şekil 4.10. Beşinci nesne renk sayım grafiği.....	50
Şekil 4.11. Altıncı nesne renk sayım grafiği.....	51
Şekil 4.12. Yedinci nesne renk sayım grafiği	51
Şekil 4.13. Sekizinci nesne renk sayım grafiği	52
Şekil 4.14. Dokuzuncu nesne renk sayım grafiği	52
Şekil 4.15. Engeller bulunan binary resim.....	55
Şekil 4.16. Sobel x işlemi sonrası	55
Şekil 4.17. Sobel y sonrası oluşan resim.....	56
Şekil 4.18. Turnuva metodu ile Rulet tekerinin karşılaştırmalı grafiği	59
Şekil 4.19. Engellere göre parçalara ayrılması	60
Şekil 4.20. Nesne sırası ve bölgeler	60
Şekil 4.21. Dört bölgenin GA performans grafiği	61
Şekil 4.22. Bölgelerdeki nesnelerin erişim rotası	61
Şekil 4.23. Başlangıç ve bitiş noktası belirlenen nesnelerin erişim rotası	62
Şekil 4.24. Başlangıç ve bitiş noktaları belirlenen nesnelerin GA performans grafiği...	63
Şekil 4.25. GA için geliştirilen programın ekran çıktısı	64
Şekil 4.26. Derinlik bulmak için geliştirilen programın ekran çıktısı.....	64

SİMGELER VE KISALTMALAR

GA	: Genetik Algoritma
GSP	: Gezgin Satıcı Problemi
GP	: Genetik Programlama
YSA	: Yapay Sinir Ağı
ÇKP	: Çok Katmanlı Perceptron
MLP	: Multi Layer Perceptron
RGB	: Red-Blue-Gren
b	: İki kamera arasındaki mesafe
z	: Kameraların nesneye uzaklığı
D	: Mercekle nesne arasındaki uzaklık
F	: Odak uzaklığı
v	: Odaktaki görüntü ile mercek arasındaki uzaklık
da	: Merceğin yarıçapı
vo	: Görüntü düzlemi ile mercek arasındaki uzaklık
db	: Görüntü düzleminde oluşan görüntünün yarıçapı
σ	: Nokta yayılım fonksiyonu uzaysal sabiti
$G(x,y)$	: Gaussian fonksiyonu
$g(x,y)$	: Binary görüntü değeri
$f(x,y)$	: Gri seviye skala
T	: Threshold (eşik değeri)
m	: Eğim
x_1,y_1	: Başlangıç koordinatı
x_2,y_2	: Bitiş koordinatı
p	: Piksel sayısı
Ç	: Çevre
A	: Alan
r	: Yarıçap

1. GİRİŞ

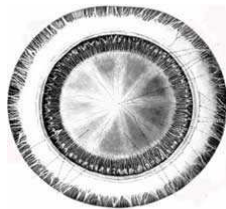
Bilgisayar teknolojisindeki gelişmeler, bilimsel araştırmaların hızını ve hacmini arttırmıştır. Araştırmalardaki amaç insan hayatını kolaylaştırmak ve yeni imkanlar sunmaktır. Araştırma alanlarının bir bölümü insan müdahalesi olmadan görevleri minimum zamanda yerine getirebilecek kendi kendine hareket eden akıllı makineler geliştirmekle ilgilidir. Görüntü işleme ve yapay zeka yöntemleri kullanılarak yapılan akıllı makinelerin de yaşamı kolaylaştırdığını görmekteyiz. Bunlara örnek olarak; tıpta (GOLDBERG, 1989; FORREST ve Ark., 1992), üretim sistemlerinde (ZHENG ve Ark., 2002), savunma sanayinde (HUANG ve Ark., 1998) kullanımları gösterilebilir.

Bu tezde canlılardaki göz-beyin-kol koordinasyonunun benzeşimi kullanılarak robot kol hareketinin optimizasyonu amaçlanmıştır.

Göz-beyin-kol koordinasyonunun sağlanmasında önemli yeri olan görme olayında nesneden yansıyan ışık göze gelir. Göz ışık sinyalinin sinirler aracılığıyla beyne iletir. Beyin hücreleri görüntüyü algılayıp yapılması gereken işlemleri yine sinirler aracılığıyla kola iletir. Böylece göz-beyin-kol koordinasyonu oluşur.

Çalışmada, sayısal fotoğraf makinesi gözün yerini alarak veri girişini yerine getirirken görüntü işleme teknikleri ve yapay zeka yöntemlerinden Genetik Algoritma (GA) ise beynin görevi olan tanıma ve karar verme işlevini yerine getirmektedir.

Canlılardaki göz ile dijital fotoğraf makinesi benzeşiminde, dijital fotoğraf makinelerinde objektife düşen ışığın CCD (Charge-Coupled Device) algılayıcılar üzerinden hafıza birimine aktarılması, gözün ışık sinyallerini sinirler aracılığı ile beyne iletmesine benzemektedir.



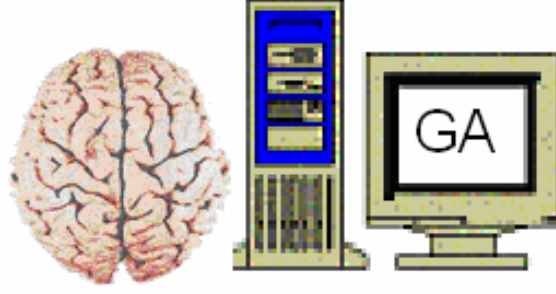
(a)



(b)

Şekil 1.1. a) Göz merceği b) Fotoğraf makinesi

Görüntü işleme teknikleri ve Genetik Algoritma (GA) ile canlılardaki beyin benzetiminde, görüntü işleme teknikleri ile nesnelerin tanımlanması ve nesnelerin koordinatlarının bulunması ile GA'nın nesneler arasındaki optimum uzaklığı bularak robot kola iletmesi, beyinin gözden gelen görüntüdeki cisimleri tanımlaması ve cisimlere en kısa yoldan erişimini kola iletmesine benzemektedir (Şekil 1.2).



Şekil 1.2. Beyin ve yapay zeka yöntemleri

İnsanlardaki görme, tanıma, karar verme yeteneklerinin simülasyonunu oluşturmak için dijital fotoğraf makinesi ve bilgisayar kullanılmıştır.

Günümüzde robotların görme sistemlerinde ses dalgası (OKUYAMA ve Ark. 2002; BORENSTEIN ve KOREN, 1989), lazer ışını (AN ve WANG, 2004) ve kamera (JAFARI ve JARVIS, 2004; WONG ve JARVIS, 2004; GONZALEZ ve Ark. 1999; ALVAREZ ve Ark. 2004; RUICHEK ve Ark. 2004) kullanılmaktadır. Bu çalışmada aynı anda birden fazla cismin tanınması ve cisimlerin üç boyutta koordinatlarının (x, y, z) bulunması için tek kamera kullanılmıştır.

Son yıllarda robot kol yönlenmesinin optimizasyonu için çeşitli yapay zeka yöntemleri kullanılmaktadır. Bu yöntemlerden yaygın olarak kullanılanlar; Yapay Sinir Ağları (YSA), Bulanık Mantık ve GA'dır (GOLDBERG, 1989; MORAVEC, 2000).

Tezde incelenen problemde Üç Boyutlu Gezgin Satıcı Problemine (GSP) benzer şekilde noktalar arasındaki optimum uzaklığın bulunması ile robot kolun en uygun rota ile nesnelere ulaşması gerekmektedir. Bu tür problemler kombinatoriyal optimizasyon problemleri olarak adlandırılırlar. Beasley, kombinatoriyal optimizasyon problemlerinin çözümlerinde GA'nın diğer optimizasyon tekniklerinden daha başarılı çözümler aldığını ayrıntılı olarak incelemiştir (BEASLEY ve Ark., 1993). Bu tezdeki problem,

kombinatorial optimizasyon problemi olduđu için GA kullanılmıştır. GSP'nin olası çözüm kümesinin büyüklüğü n nokta sayısı olmak üzere,

$$\text{Olası Çözüm Sayısı}=(n-1)! \quad (1.1)$$

kadardır (LALENA, 2005). Bu işlem 30 şehri dolaşan ve tekrar başlangıç noktasına dönen bir satıcı için yapıldığında $(31-1)! = 2,65 \times 10^{32}$ sonucunu vermektedir. Saniyede 1 milyon işlem yapabilen bir bilgisayar $2,65 \times 10^{32}$ tane çözüm arasında uygun çözümü yaklaşık olarak 8.000.000.000.000.000.000 yılda bulmaktadır (LALENA, 2005). Bu çalışmada da problemlerin benzerliği nedeniyle yukarıdaki formül ve sayılar geçerlidir.

2. ÖNCEKİ ÇALIŞMALAR

2.1. Derinlik Bulma Tekniklerine Giriş

Günümüz teknolojisindeki hızlı gelişme, hız ve hafıza kapasiteleri çok yüksek boyutlara ulaşan bilgisayarları da beraberinde getirmiştir. Çok yakın bir gelecekte bilgisayarların, insan fonksiyonlarının büyük bölümünü yerine getirebilecek robotlara dönüşmesi beklenen bir sonuçtur. Görüntü işleme, robotik kontrol ve özellikle düşünen bilgisayarlar konusunda yoğunlaşan bilimsel araştırmalar, bu amaca yönelik alt çalışmalardır.

Bu açıdan bakıldığında, robotların görsel yeteneklerinin geliştirilmesine duyulan ihtiyaç bir başka araştırma alanı yaratmaktadır. Çevreyle iletişimli robotlar için görme yeteneğini geliştirmesi konusu, günümüzde üzerinde yoğun bir şekilde çalışılan konular arasında yer almaktadır.

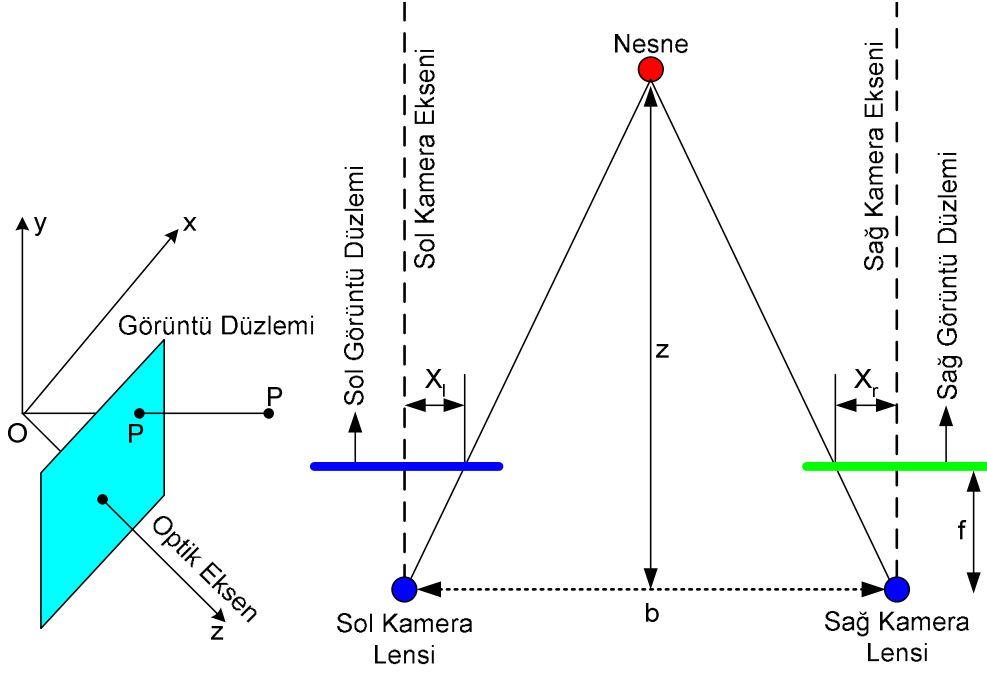
Tek kamera ile derinlik bulma tekniklerinden bazıları şunlardır.

2.1.1. Tek Kamera İle Derinlik Bulma

a) Kamera hareketi ile derinlik: İnsan gözünün karakteristik özelliğine göre iki kamera veya tek kameranın yatay düzlem üzerindeki hareketine bağlı olarak derinlik bulunmuştur. Bu sistemde kameralar nesnenin aynı noktasına odaklanmış ve b iki kamera arasındaki mesafe, f kameranın odak uzaklığı, x_l ve x_r kameraların optik eksenini ile görüntü düzleminde oluşan nesnenin görüntüsü arasındaki mesafe olmak üzere;

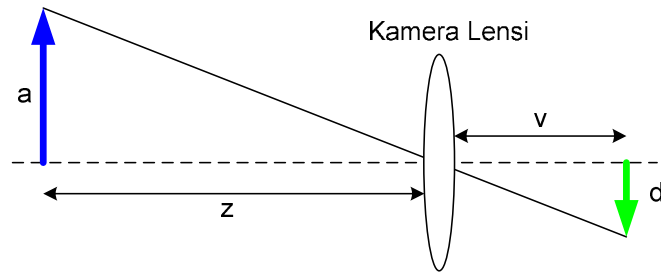
$$z = \frac{b \cdot f}{(x_l - x_r)} \quad (2.1)$$

eşitliği ile derinlik hesaplanmıştır (JAIN ve Ark., 1995)



Şekil 2.1. Kameranın yatay düzlem üzerinde hareketi

b) Kıyas ile derinlik: Büyüklüğü bilinen bir cismin tek kamera ile odak uzaklığına bağlı olarak büyüklüğündeki değişme miktarından derinlik bulunmaya çalışılmıştır (YAMAGUTI, 1997) (Şekil 2.2).



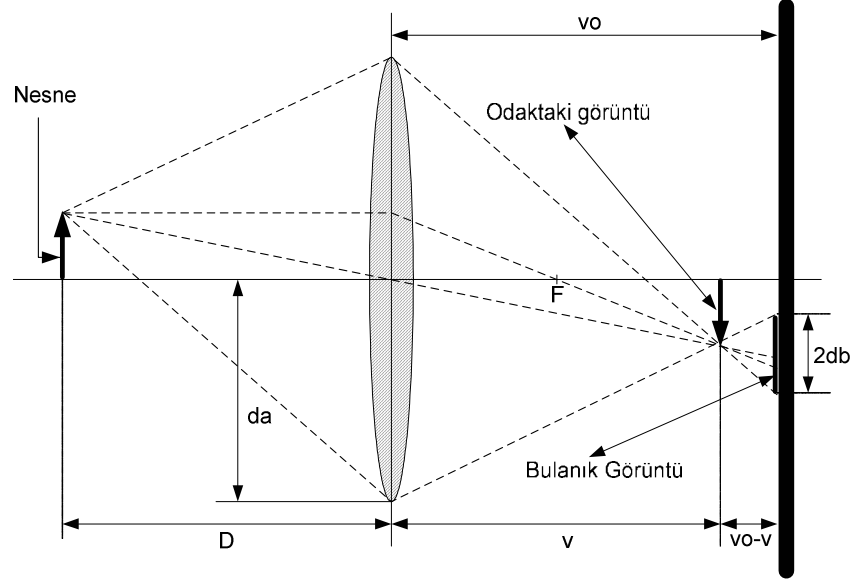
Şekil 2.2. Kıyas ile derinlik parametreleri

Şekil 2.2’de parametrelerden, a nesnenin büyüklüğü, z nesne ile kamera arasındaki uzaklık, d nesnenin görüntü düzleminde oluşan büyüklüğü, v görüntü düzlemi ile kamera lensi arasındaki uzaklık, f odak uzaklığı olmak üzere;

$$\frac{1}{f} = \frac{1}{z} + \frac{1}{v} \quad (2.2)$$

$$d.z = v.a \quad (2.3)$$

lens formüllerinden z çekilerek mesafe hesaplanmıştır. Aynı çalışmada kamera veya nesne yatay düzlem üzerinde hareket ettirilerek nesnenin büyüklüğündeki değişme miktarları da incelenmiştir.



Şekil 2.3. Optik lens ve imajın oluşması

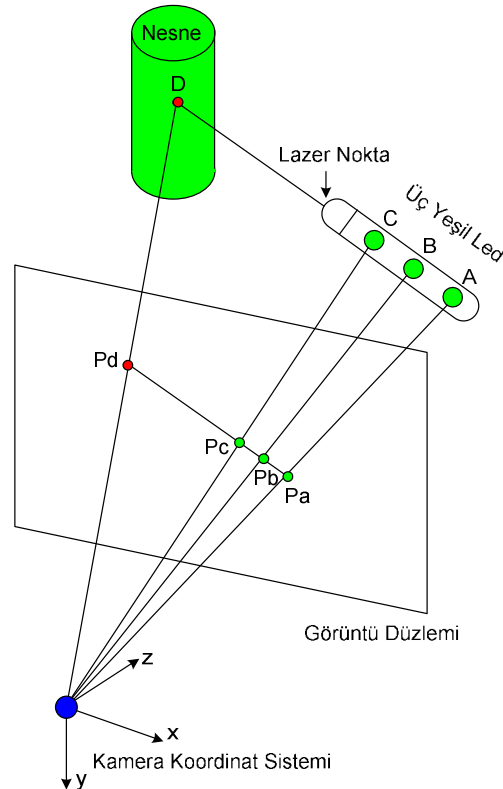
c) Gaussian ile derinlik: Odak dışındaki görüntü için nokta yayılım fonksiyonunun Gaussian şeklinde olduğu kabul edilmiştir (PENTLAND, 1982; SUBBARAO ve GURUMORTHY, 1982). Bu konuda özellikle Subbarao ve Pentland'ın çalışmaları önemli bir yer teşkil eder. Her ikisi de odak dışındaki bulanık görüntü için nokta dağılım fonksiyonunun iki boyutlu Gaussian fonksiyonu şeklinde olduğunu kabul etmiş ve daha sonra ulaşılan bulanık görüntünün bazı özelliklerini ve kamera parametrelerini (Şekil 2.3'te görülen parametreleri) kullanarak derinlik hesabının nasıl yapılabileceği üzerinde durmuşlardır.

Derinlik hesabında kullanılan parametreler F odak uzaklığı, v_o görüntü düzlemi ile mercek arasındaki uzaklık, f mercek sisteminin numarası ve σ nokta yayılım fonksiyonunun uzaysal sabitidir. Uzaysal sabit σ , görüntüleme işlemine göre bir görüntü noktasının nasıl bulanık bir şekil alacağını tanımlar. Nokta yayılım fonksiyonunu, σ uzaysal sabitiyle iki boyutlu Gaussian $G(x,y)$ fonksiyonu olarak eşitlik (2.4) ile tanımlanmış ve eşitlik (2.5) ile derinlik bulunmuştur.

$$G(x, y) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma}} \cdot e^{-\frac{(x^2 + y^2)}{2 \cdot \sigma^2}} \quad (2.4)$$

$$D = \frac{F \cdot v_o}{v_o - F - \sigma \cdot f} \quad (2.5)$$

d) Lazer ile derinlik: Derinliği bulmak için bir çubuk üzerine yerleştirilmiş üç yeşil led, bu çubuğun eksenine yerleştirilmiş lazer ışık kaynağı ve CCD kamera kullanılmıştır. Lazer ışık kaynağından nesne üzerine bir nokta yansıtılmıştır. Nesne üzerindeki bu nokta ile yeşil ledlerin görüntüleri CCD kamera ile alınarak görüntü işleme teknikleri uygulanmıştır. Görüntü işleme teknikleri ile noktalar Şekil 2.4'te gösterilen üçgenlere dönüştürülmüş ve üçgenlerin tüm bilgileri matrislere aktarılarak derinlik bulunmuştur. Bu sistemde nesnenin istenilen noktasındaki mesafeyi bulmak için o noktaya lazer ışık yansıtılmaktadır. (TAKATSUKA ve Ark., 1999).



Şekil 2.4. Lazer ile derinlik

e) Yapay sinir ağı ile derinlik: Derinlik çevrenin iki bulanık resminden hesaplanmıştır. Çok Katmanlı Perceptron (ÇKP) (Multi Layer Perceptron (MLP)) yapısındaki bir yapay sinir ağı (YSA) geri yayılım algoritması ile, girişi bir kıstas fonksiyonu ile elde edilen resim netlikleri, çıkışı ise mesafe olmak üzere eğitilmiştir. Eğitilen YSA farklı mesafelerde alınan çeşitli resimlerle test edilerek mesafe bulunmaya çalışılmıştır (ASLANTAŞ ve TUNÇ, 2004).

2.2. GA Araştırmalarının Tarihi Gelişimi

1961’de Bledsoe, 1962’de Bremermann tarafından evrime dayalı optimizasyon teknikleri kullanılmıştır. Bu teknikler modern GA notasyonuna yakındır. Bledsoe ve Bremermann binary dizi kodlarını kullanmayı önermiş, bireyden bireye jenerasyon, mutasyon ve en iyi seçimin saklanmasıyla oluşan şemanın sonuçlarını sunmuşlardır. Bremermann bu çalışmaları devam ettirmiş ve çalışmalarında rekombinasyon operatörünü kullanmıştır.

1965’te evrime dayalı optimizasyon teknikleri Rechenberg tarafından evrim stratejileri (evolution strategie) adı altında toplanmıştır.

GA, Michigan üniversitesinde araştırmacı grubunun başı olan John Holland ve meslektaşları tarafından geliştirilmiştir. Araştırmalarının hedefi doğal sistemlerin uyum sistemlerini açıklamak ve yapay sistem yazılımlarının doğal sistemlerin önemli mekanizmalarının kaybedilmeden dizayn edilmesi idi. 1975 yılı GA’nın gelişmesi için önemli bir yıl olmuştur. Bu yılda Holland “Adaptation in Natural and Artificial Systems” isimli kitabını yazmış Michigan Üniversitesi Bilgisayar Bilimleri Bölümün’de doktora öğrencisi olan Kenneth A. De Jong, Holland’ın teorileri ile kendi çalışmalarını birleştiren “An Analysis of the Behavior of a Class of Genetic Adaptive Systems” isimli tezini tamamlamıştır (GOLDBERG, 1989).

Bütün bu gelişmelere rağmen Holland’ın öğrencisi David E. Goldberg’in “Genetic Algorithms in Search, Optimization, and Machine Learning” isimli kitabı 1989’da yayımlanıncaya kadar GA’nın önemi anlaşılmamıştır (TURGUT ve ARSLAN, 2001).

1990’da Stanford Üniversitesinden John Koza, GA’ya dayalı genetik programlamayı (GP) tanımlamıştır. GP genetik algoritmayı kullanarak çeşitli görevleri

yerine getiren programlardır. Koza, GP'yi kullanarak yapay sinir ağı dizayn etmiş, matematikteki bazı denklemleri çözmüş ve Kepler'in 3. Kanununu trigonometriden yararlanarak yeniden ispat etmiştir (KOZA, 1990).

Karmaşık problemler hızlı ve en uygun sonucu bularak çözen GA pek çok alanda kullanılmaktadır. GA'nın kullanıldığı alanlar:

Otomatik Programlama ve Bilgi Sistemleri: GA, belirli görevler için bilgisayarlar programlar geliştirilmesinde kullanılabilir. Ayrıca hesaplama gerektiren bilgisayar entegrelerinin tasarımında, ders programı hazırlanmasında, dağıtılmış bilgisayar ağlarının tasarımında da kullanılabilirler (EMEL ve TAŞKIN, 2002; DABAN, 2004).

Mekanik Öğrenme (Makine Öğrenmesi): Makine öğrenmesine dayalı genetik (Genetics-Based Machine Learning (GBML)), sınıflandırma sistemi olarak da adlandırılır. GA, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadırlar (GOLDBERG, 1989).

Ekonomik ve Sosyal Sistem Modelleri: Bir sistemi ölçen, deneysel olarak gözlenmiş değişkenler arasındaki matematiksel ilişkiyi bulmakta kullanılır (KOZA, 1992).

Finans: Hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi ve uluslar arası sermaye tahsisi stratejilerini belirlemede genetik algoritmalar kullanılmaktadır (SCHLOTTMANN, 2001).

Pazarlama: Tüketici profili çıkarma pazarlama stratejisi oluşturmada önemlidir. Tüketici profili çıkarma çok büyük veri tabanlarının kullanılmasını gerektirir. GA bu veri tabanlarından model elde edilmesinde kullanılır (EMEL ve TAŞKIN, 2002).

Montaj Hattı Dengeleme Problemi: Endüstride montaj işleminde her bir iş istasyonundaki toplam işlem zamanlarını minimize etmek için kullanılır (LIT, 2001; ÖZÇAKAR, 1998).

Çizelgeleme Problemi: Siparişe göre üretimin yapıldığı iş atölyelerinin çizelgeleme problemi ve sürekli ve hücresele üretim yapan akış atölyelerinin çizelgeleme probleminin GA ile çözümünde optimuma yakın sonuçlar bulunur (EMEL ve TAŞKIN, 2002; WADHWA ve CHOPRA 2000).

Tesis Yerleşim Problemi: Bir tesisteki araç-gereç ve diğer malzemeleri istenen kriterler doğrultusunda yerleştirilmesi çok karışık olabileceği için tesis yerleşim probleminin çözümünde GA kullanılmıştır (EMEL ve TAŞKIN, 2002).

Atama Problemi: n elemanın n farklı göreve atanması problemlerinde GA'nın kullanıldığı çözümler vardır (ÖZÇAKAR, 1998; YENİAY, 2001).

Gezgin Satıcı Problemi (GSP): GSP'de amaç alınan yolu minimize etmektir. Posta taşıyıcılarının, okul otobüs rotalarının bulunması gibi problemler GSP olarak modellenenmektedir (DONG, 1999; LALENA, 2005; ÖZDEMİR ve ORAL, 2003).

Araç Rotalama Problemi: Temel araç rotalama probleminde tek bir depodan araçlar ayrılmakta ve müşteri talepleri karşıladıktan sonra depoya geri dönmektedir. Problem verilen kısıtlara göre maliyeti en aza indirmektir (TAN ve ark., 2001).

Protein Yapısının Bulunması: DNA ve protein dizilerinin yapısının bulunmasında GA'nın bir uzanımı olan GP kullanılmaktadır (KOZA, 1990).

2.3. Görüntü İşleme Tekniklerinde GA Kullanılmasının Tarihi Gelişimi

GA'nın görüntü işlemede kullandığı bazı konular:

- 1970 yılındaki Cavicchio GA'yı örnek tanıma (Pattern Recognition) problemine uyarlamıştır (GOLDBERG, 1989).
- Tıpta X ışını ile elde edilen görüntülerin karşılaştırılmasında (GOLDBERG, 1989)
- Bağışıklık sistemi için model tanıma (FORREST ve Ark., 1992).
- Yönlendirilmiş erozyon ve dilation işlemlerinin otomatik olarak yapılması (YODA, 1995).
- Yüz tanıma (HUANG ve Ark., 1998).
- GA ile imza tanıma (RAMESH ve MURTY, 1999)
- Metal yüzeylerdeki kusurun otomatik olarak bulunması (ZHENG ve Ark., 2002).
- Sayısal görüntülere veri gizleme yöntemlerinde kullanılmıştır (SHIEH ve Ark., 2004).

2.4. Robotlarda GA Kullanılmasının Tarihi Gelişimi

Robotikte GA kullanılarak optimize edilen birçok parametre vardır. Örnek olarak sensörlerden gelen bilgilerin optimizasyonunda, kontrol parametrelerinin düzenlenmesinde kullanılabilir.

Brooks, mobil robotların kontrol edilmesi için çaprazlama performansına bağlı GP ve GA kullanmıştır (BROOKS, 1992).

Harvey, GA'yı kullanarak görsel yönlendirilmiş robotlarda (visually guided robots) yapay sinir ağlarının kullanılmasını geliştirmiş ve sınırlı sürede maksimum alanda dolaşan robot yapmıştır (HARVEY ve Ark., 1992).

Bessiere ve arkadaşları, engellerin hareket ettiği dinamik bir çevrede robotun gideceği yolu planlayan bir algoritma geliştirdiler (Ariadne's Clew Algorithm). Algoritmaları keşif ve arama olmak üzere iki kısımdan oluşuyordu. Bessiere ve arkadaşları algoritmanın iki kısmında da paralel GA kullanmışlardır (BESSIERE ve Ark., 1993).

Matthies, robotlar için stereo görüntüler kullanarak engel tanıyan, yol bulan, risk belirleyen, arazide yön bulan evrimsel algoritmalar geliştirmiştir. Bu algoritmalar Mars Pathfinder uzay aracında denenmiştir (MATTHIES ve Ark., 1995).

Floreano ve Mondada, yenilenen yapay sinir ağlarının evriminde GA'nın kullanıldığı bir sistem tanımlamışlardır. Tanımladıkları bu sistemle çalışan Khepera isimli bir robot yapmışlardır. Bu robot bir iterasyon süresi 40 dakika süren 100 iterasyon sonunda bir odanın içinde köşelere çarpmadan gezmeyi ve engelden kaçmayı öğrenmiştir (FLOREANO ve MONDADA, 1998).

National Research Council Roma'da araştırmacı olan Nolfi, Khepera robota çöp, duvar ve hedefi ayırt edebilen, çöp toplayan ve çöpleri duvarın dışına atan modüller eklemiştir. Nolfi, hedefler arasındaki uzaklığı hesapladığı modülü GA ve yapay sinir ağlarını kullanarak oluşturmuştur (NOLFI, 1997).

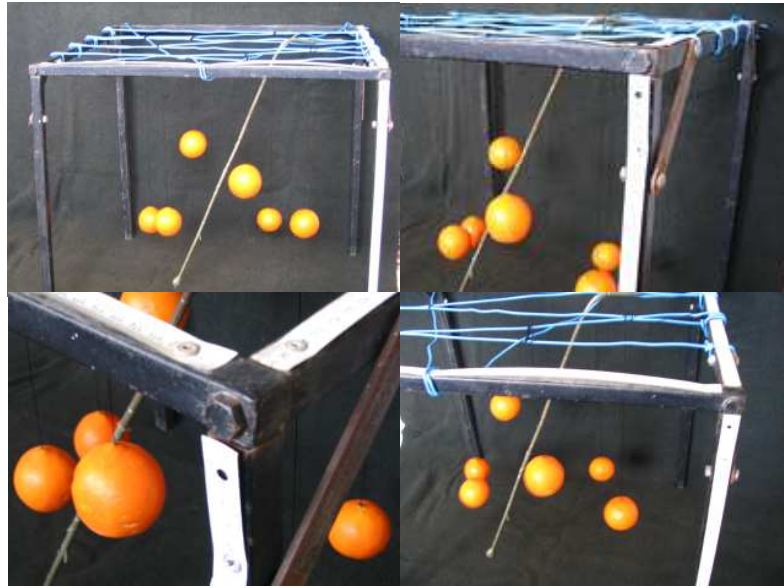
Literatür taramasında bulunabilen çalışmalarda görüntü işleme teknikleri ve GA labirentten çıkış yolunu bulan, engelden kaçan, duvarlarla çevrili bir alanda dolaşabilen ve çöp toplayıp duvarın dışına atan robotların geliştirilmesinde kullanılmıştır (BESSIERE ve Ark., 1993; NOLFI, 1997; KANASUKI ve TAKAHASHI, 2001; MATTHIES ve Ark., 1995; FLOREANO ve MONDADA, 1998). Bu çalışmalarda aynı

anda ya tek nesne üzerinde çalışılmış veya derinlik analizleri sensörler ile yapılmıştır. GA ise nesnelere erişim sırasını belirlerken veya hedefe ulaşmak için en kısa yolu bulmak için kullanılmışlardır. Tezdeki araştırma ise bu çalışmalardan farklı olarak tek kamera ile aynı anda birden fazla nesnenin koordinatları bulunmuş ve nesnelere erişim sırasının optimizasyonu GA ile yapılmıştır.

3. MATERYAL VE YÖNTEM

3.1. Materyal

Deney düzeneği 100x100x100 cm'lik küp şeklinde olup, üzerine nesnelerin yerleştirilmesi için iskelet yapıda dizayn edilmiştir (Şekil 3.1). Yazılım, 800 MHz işlemci, 256 Mbyte RAM, 8 Mbyte ekran kartına sahip PC kullanılarak, Borland Delphi 7.0 programlama dili ile geliştirilmiştir. Deneylerde kullanılan fotoğraflar 800x600 piksel ve 24 bitlik olup 3.2 mega piksellik dijital fotoğraf makinesiyle çekilmiştir. Hazırlanmış deney düzeneğinde nesneler ve engeller birbirleri ile temas halinde olmadan çekilen fotoğraflar 24 bit *.BMP formatında incelenmiştir.



Şekil 3.1. Deney düzeneği

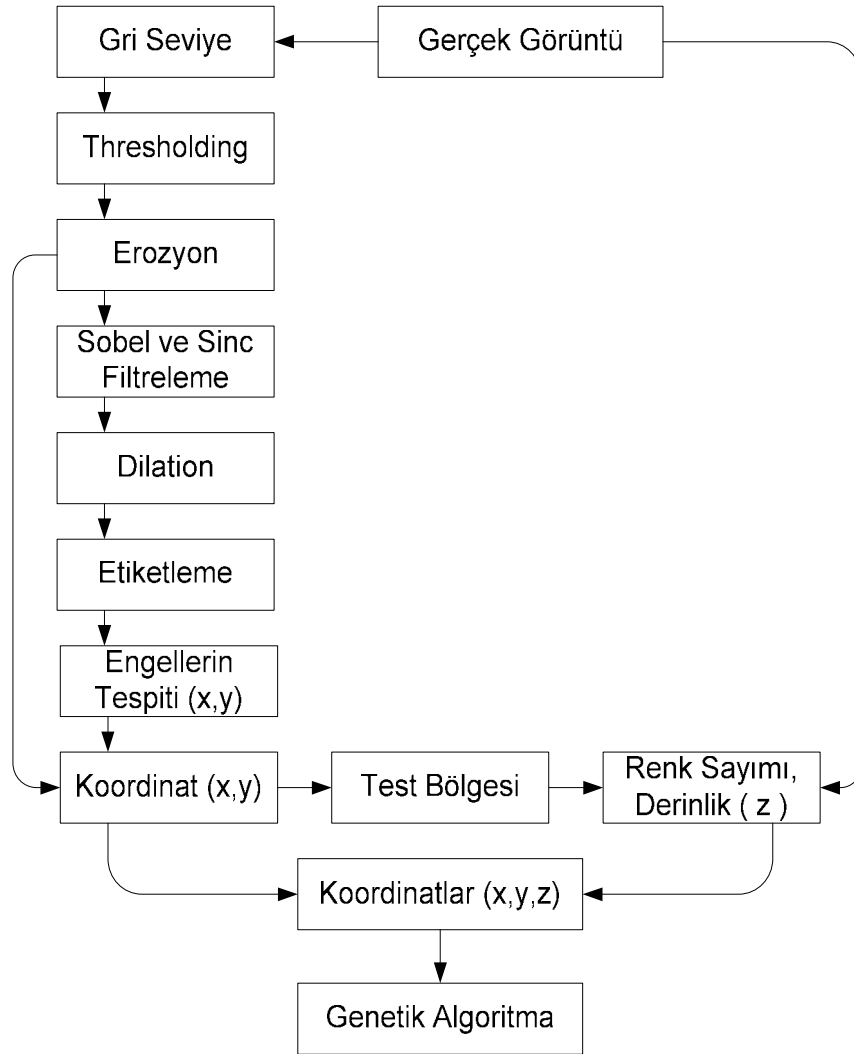
3.2. Yöntem

Yöntem iki ana kısımdan oluşmaktadır. Birinci kısımda görüntü işleme teknikleri kullanılarak nesnelerin koordinatlarının tespiti, ikinci kısımda ise GA ile robot kolun izleyeceği yolun kısaltılarak, daha etkin kullanımı anlatılmaktadır.

3.2.1. Görüntü İşleme ve Çalışmada Kullanılan Görüntü İşleme Teknikleri

Görüntü işleme teknikleri görüntünün gri seviye resme dönüştürülmesi, binary resmin elde edilmesi, nesnelerin sayılması ve görüntüde nesne ve zeminin veya resim içerisindeki ilgilenilen değişik özelliklere sahip bölgelerin birbirinden ayrıştırılması için kullanılmıştır (PITAS, 1995).

Uygulanan görüntü işleme teknikler Şekil 3.2'deki işlem akış sırasında aşağıda anlatılmıştır.



Şekil 3.2. Koordinatlarının (x, y, z) bulunması için akış diyagramı

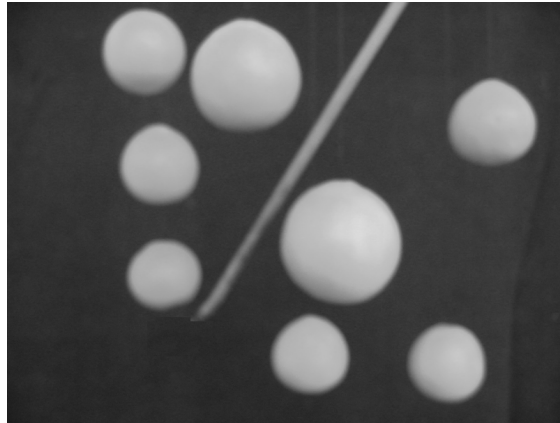
3.2.1.1. Gri Seviye Resim

Resim piksellerindeki RGB (Red-Green-Blue) ile gösterilen Kırmızı, Yeşil, Mavi renk kanallarının (3.1) no'lu bağıntıda verilen katsayılarla çarpımının toplamı o pikselin gri-seviye karşılığını verir. Bu işlem resimdeki tüm piksellere uygulanarak renkli resim gri seviyeye dönüştürülür (GONZALEZ ve WOODS, 1992). Şekil 3.3'de görülen gerçek resime gri seviye dönüşüm uygulanarak Şekil 3.4'deki görüntü elde edilir.

$$Y=0.299R+0.587G+0.114B \quad (3.1)$$



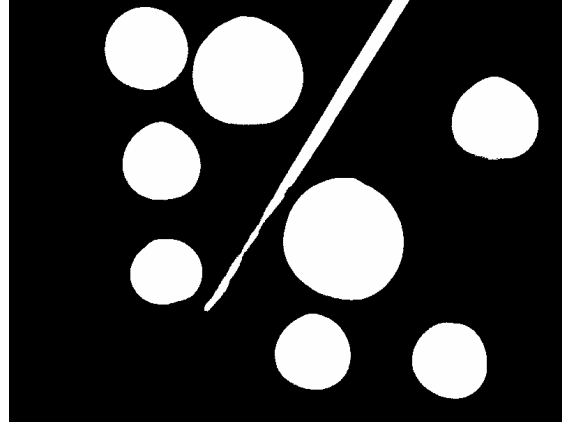
Şekil 3.3. Renkli resim



Şekil 3.4. Gri seviye resim

3.2.1.2. Eşikleme

Gri seviyeye dönüştürülen resimdeki renk değerleri 0 ile 255 arasında değişir. Eşik değeri için 0 ile 255 arasında bir değer önceden seçilir. Resimdeki piksellerin gri seviye değeri eşik değerinden küçük veya eşitse pikselin yeni değeri 0, büyükse 1 (255) olacak şekilde değiştirilir (3.2) ve iki renkli (siyah-beyaz) bir resim elde edilir. Bu işlem ile resim binary resme dönüştürülmüş olur. Dönüştürülen resimde beyaz olarak görülen bölgeler nesneyi, siyah olarak görülen bölgeler ise zemini temsil eder (Şekil 3.5).



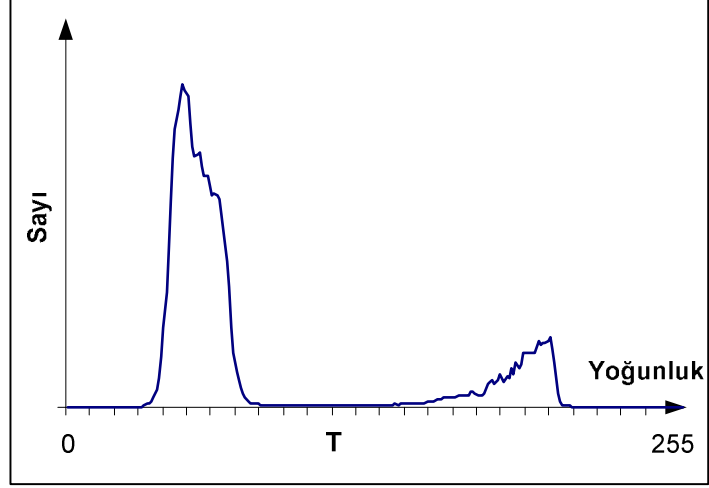
Şekil 3.5. Binary resim

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & \text{aksi takdirde} \end{cases} \quad (3.2)$$

$g(x, y)$ = binary görüntü değeri, $f(x, y)$ = gri seviye skala, T = threshold (eşik değeri)

Şekil 3.6'deki gri seviye resimde koyu bir zemin üzerinde parlak bir nesnenin histogramı görülmektedir. Bu grafik incelendiğinde piksellerin dağılımı iki gruba ayrılmıştır. Gri seviye bir resimde piksellerin renk değerleri (0-255) aralığında değiştiği için, histogramdaki pikseller (0-255) arasında dağılmıştır. Eğer eşik değeri olarak T değerini seçersek ve bu eşik değerinden küçük gri seviye değerine sahip pikselleri 0, T eşik değerinden büyük ve eşit olanları ise 1 (255) ile değiştirilerek iki renkli (siyah-

beyaz) bir resim elde edilir. Bu resimde beyaz olarak görülen bölgeler nesneyi, siyah olarak görülen bölgeler ise zemini temsil eder (Şekil 3.5).



Şekil 3.6. Gri seviye resmin histogramı ve thresholding değeri

Basit thresholding algoritmasının pseudo kodu aşağıdadır:

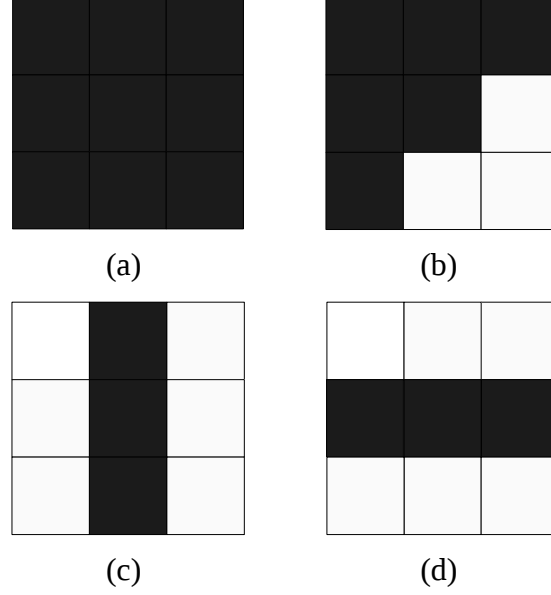
- 1) Eşik değerini seç, x ;
- 2) Görüntünün başlangıç ve bitiş koordinatlarını bul, (X_0, Y_0) , (X_n, Y_n) ;
- 3) Başlangıç koordinatından bitiş koordinatına kadar piksellerin gri seviye değerine bak;
- 4) Gri seviye değeri eşik değerinden küçükse yeni değeri 0, büyükse 1 yap;

3.2.1.3. Yapısal Eleman

Erozyon ve dilation (genişleme) işlemlerinin gerçekleşmesi için en önemli öğe yapısal elemandır. Yapısal eleman çeşitli geometrik şekillerden biri olabilir. En sık kullanılanlar; kare, dikdörtgen ve üçgendir (GONZALEZ ve WOODS, 1992; JAIN ve Ark., 1995). Şekil 3.7’de çeşitli yapısal elemanlar gösterilmektedir.

Yapısal elemanın merkez pikselinin binary bir resim içerisindeki nesnelere temas etmesiyle ortaya çıkan nesne ile yapısal elemanın kesişim kümesi nesneden çıkartılarak veya nesneye eklenerek bir çok morfolojik işlem gerçekleştirilir ve

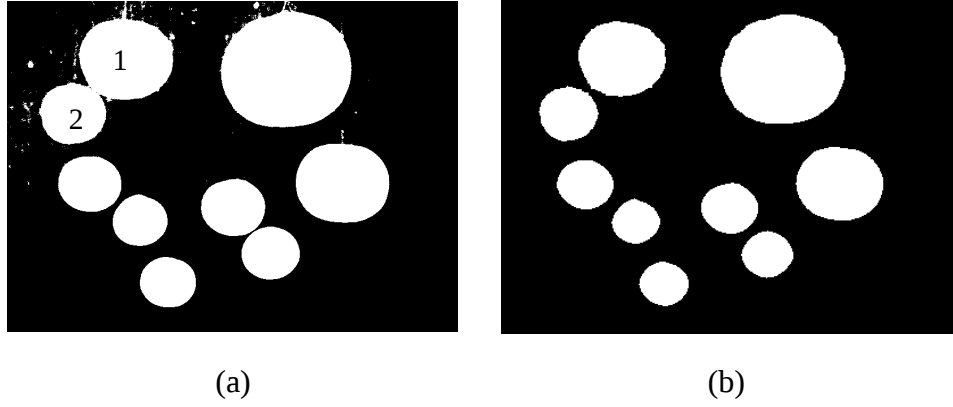
nesneden çıkartılan veya eklenen kısmın yapısal özelliği yapısal eleman tarafından belirlenir. Çalışmalarımızda Şekil 3.7.a'da görülen 3x3 şeklindeki kare yapısal elemanı seçilmiştir.



Şekil 3.7. a) Kare, b) Üçgen, c) Dikey dikdörtgen, d) Yatay dikdörtgen yapısal eleman

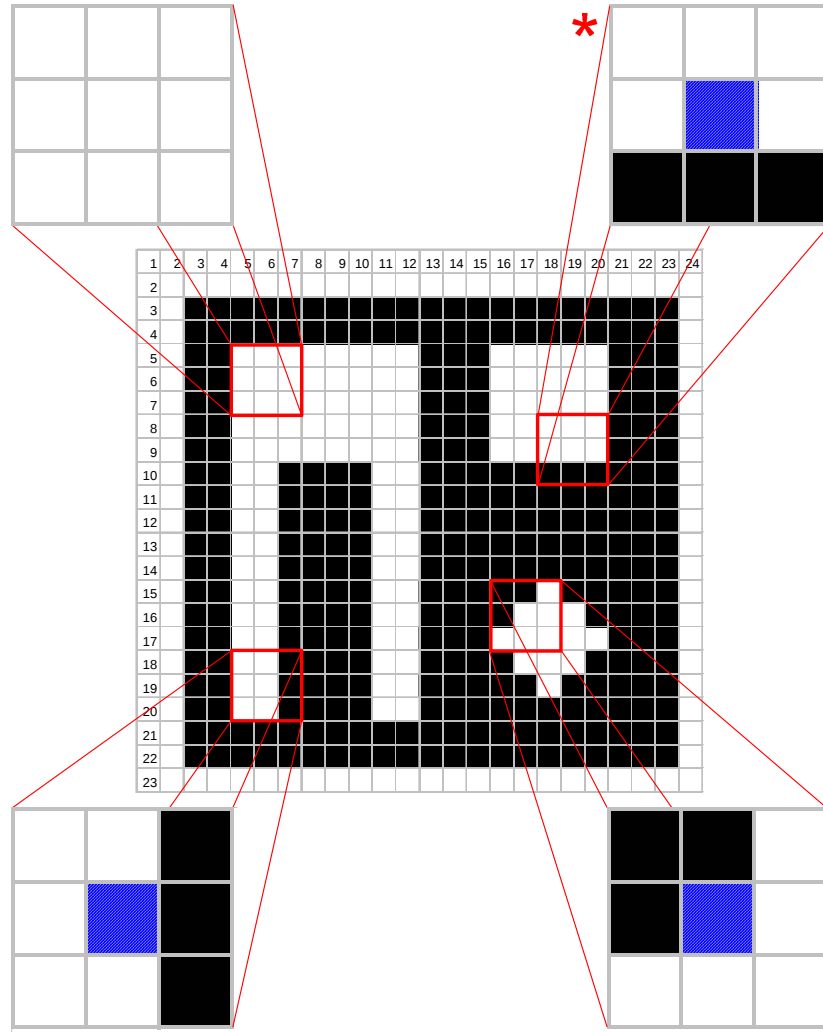
3.2.1.4. Erozyon

Gri seviye resim thresholding işlemi ile binary resime dönüştürüldükten sonra binary resim içindeki birbirine yakın olan nesneleri uygun bir yapısal eleman kullanarak birbirinden ayırmak ve resimdeki gürültüden kaynaklanan bozulmaları yok etmek amacı ile binary resme erozyon uygulanır. Şekil 3.8.a'da 1 ve 2 nolu nesnelerin birbirlerine yapışmış olduğu ve nesnelerin etrafında gürültüden oluşan noktalar görülmektedir. Şekil 3.8.b'de ise erozyon işlemi ile 1 ve 2 nolu nesnelerin birbirinden ayrıldığı ve gürültüden oluşan noktaların yok olduğu görülmektedir.



Şekil 3.8. a) Erozyon öncesi b) Erozyon sonrası

Erozyon işleminde resim içerisindeki nesnelerin en ve boyları aynı oranda azaltılmak istendiğinden kare yapısal eleman kullanılmıştır. Kare yapısal eleman ile nesne karşılaştırılarak erozyon işlemi uygulanır (GONZALEZ ve WOODS, 1992; JAIN ve Ark., 1995). Karşılaştırılan bölümlerden tümü aynı ise hiçbir değişiklik yapılmaz, fakat karşılaştırmada bir fark oluşuyorsa nesnede karşılaştırılan kısmın merkezi 0 yapılı. Şekil 3.9'da örnek bölgelerdeki erozyon sonuçları görülmektedir. Mavi olarak gösterilen bölge erozyon işlemi ile 0 yapılacak pikselleri göstermektedir. Şekil 3.9'da * ile gösterilen örnek bölge incelendiğinde 6 pikselin nesne üzerinde 3 pikselin ise zemin üzerinde bulunduğu ve mavi ile gösterilen merkez pikselin 0 olacağı gösterilmektedir.



Şekil 3.9. 3x3 Kare yapısal elemanı ile erozyon işlemi

Basit erozyon algoritmasının pseudo kodu aşağıdadır;

- 1) Görüntünün başlangıç ve bitiş koordinatını bul, (X_0, Y_0) , (X_n, Y_n) ;
- 2) Başlangıç koordinatından bitiş koordinatına kadar pikselleri yapısal eleman ile karşılaştır;
- 3) Yapısal elemanın karşılaştığı pikseller yapısal eleman ile aynı ise değiştirme, farklı ise yapısal elemanın orta hücresinin karşılaştığı pikseli 0 yap;

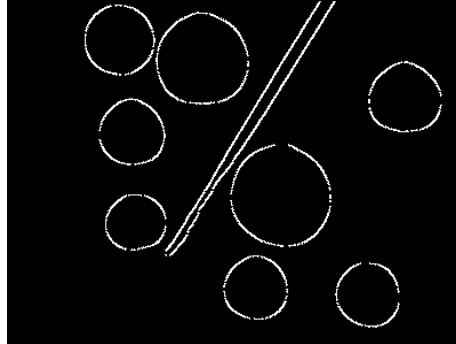
3.2.1.5. Sinc Filtreleme

Binary resimde kenarları ortaya çıkarmak için kullanılan filtrelerden biri Sinc filtresidir.

$$S = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Şekil 3.10. Sinc filtre

Matematikteki sinc fonksiyonunun sayısal resimlere uygulanması ile gerçekleşir. Şekil 3.10'da sinc filtesi görülmektedir. Şekil 3.11'de binary resme sinc filtesi uygulandıktan sonra oluşan resim görülmektedir.



Şekil 3.11. Sinc filtreleme sonrası

3.2.1.6. Sobel Filtreleme

Binary resim içindeki istenilen ayrıntıları (kenarları) ortaya çıkarmak için Sobel kenar bulma filtreleri kullanılır. Sobel x ve Sobel y olmak üzere iki filtre kullanılmıştır. Sobel x filtreleme ile nesnelerin x düzlemindeki kenarları, Sobel y filtreleme ile de y düzlemindeki kenarları bulunur.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Şekil 3.12. Sobel x ve Sobel y filtreleri

Şekil 3.12’de Sobel x ve Sobel y filtreleri görülmektedir. Resmin bir bölgesinden örnekleme olarak alınan Şekil 3.13’de görülen 3x3 matris şeklinde örnek piksel komşulukları ile filtreler ile üst üste bindirilerek yapılan toplama işleminde Sobel x filtresi için (3.3) bağıntısı, Sobel y için de (3.4) bağıntısı elde edilir. Bağıntı sonuçları [i,j] pikseline yazılarak filtreleme yapılır (JAIN ve Ark., 1995).

a_0	a_1	a_2
a_7	[i,j]	a_3
a_6	a_5	a_4

Şekil 3.13. 3x3 şeklinde örnek piksel komşuluğu

$$S_x = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6) \quad (3.3)$$

$$S_y = (a_0 + 2a_1 + a_2) - (a_6 + 2a_5 + a_4) \quad (3.4)$$

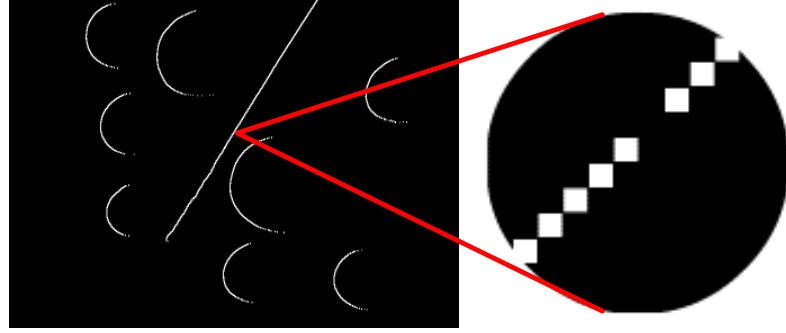


Şekil 3.14. a) Sobel x sonrası b) Sobel y sonrası

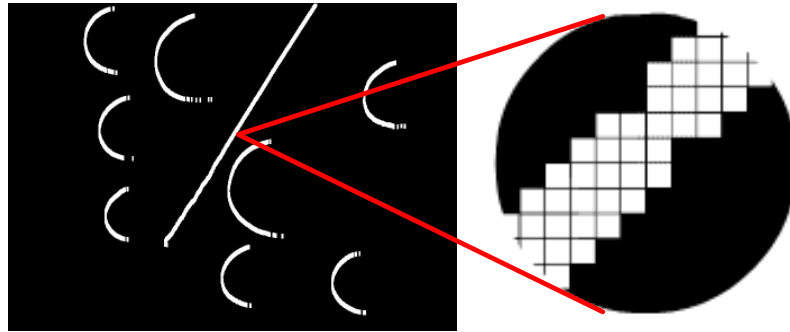
Şekil 3.14.a'da Sobel x ve Şekil 3.14.b'de Sobel y kenar bulma filtreleme işleminden sonraki resimleri görülmektedir.

3.2.1.7. Genişleme

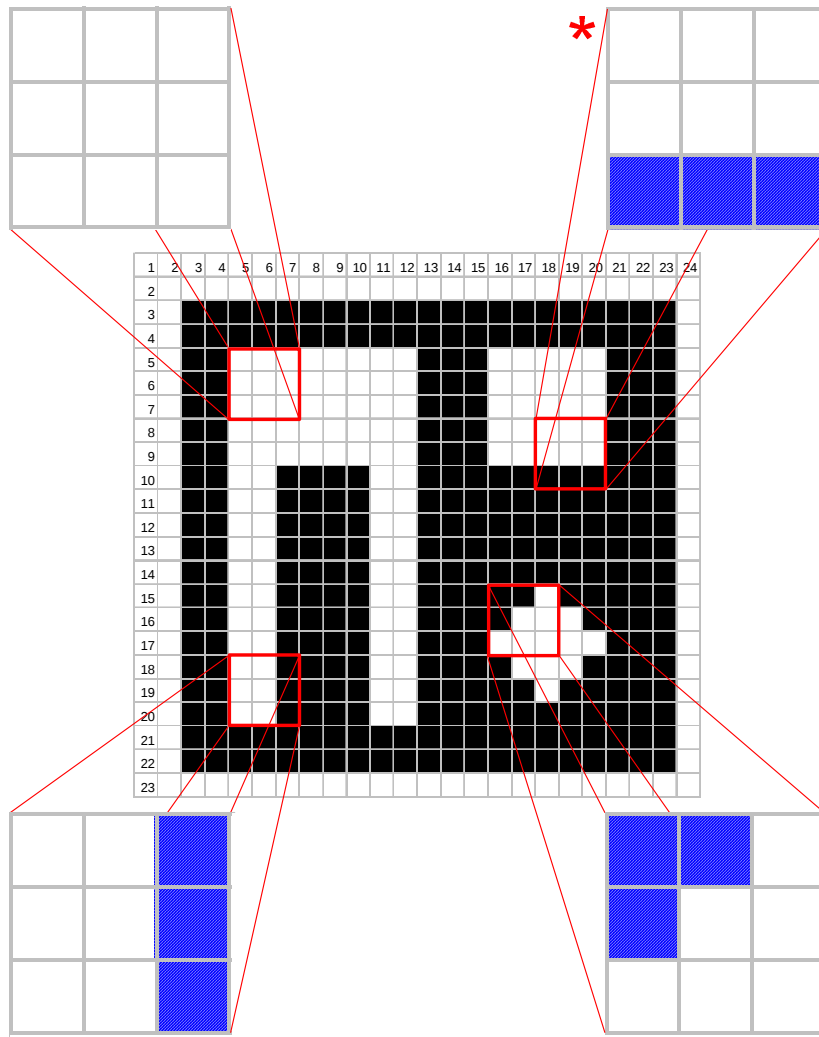
Sobel kenar bulma operatörleri ile bulunan kenarlar içindeki küçük kopuklukları (boşlukları) tamamlamak amacıyla genişletme (dilation) uygulanır. Genişletme işlemi için 3x3 kare yapısal elemanı kullanılmıştır. Şekil 3.15'de dilation öncesi resimdeki çizginin kopuk olduğu ve Şekil 3.16'da dilation uygulandıktan sonraki çizgideki kopukluğun düzeltildiği görülmektedir.



Şekil 3.15. Dilation öncesi



Şekil 3.16. Dilation sonrası



Şekil 3.17. 3x3 Kare yapısal elamanı ile dilation işlemi

Dilation işlemi erozyonun tam tersidir. Dilation işleminde resim içerisindeki nesnelerin en ve boyları aynı oranda arttırmak istendiğinden kare yapısal eleman kullanılmıştır. Kare yapısal eleman ile nesne karşılaştırılarak dilation işlemi uygulanır (GONZALEZ ve WOODS, 1992; JAIN ve Ark., 1995). Karşılaştırılan bölümlerden yapısal elemanın orta noktası, nesnenin herhangi bir pikseli üzerindeyse yapısal elemanın karşılaştığı tüm kısımlar 1 (255) yapılır. Eğer yapısal elemanın orta noktası nesne üzerinde değilse herhangi bir değişiklik yapılmaz. Şekil 3.17’de örnek bölgelerdeki dilation sonuçları görülmektedir. Mavi olarak gösterilen bölge dilation işlemi ile 1 (255) yapılacak pikselleri göstermektedir. Şekil 3.17’de * ile gösterilen örnek bölge incelendiğinde merkez pikselin nesne üzerinde bulunduğu buna bağlı olarak

ta yapısal elemanın üzerinde bulunduğu zemine ait olan piksellerin 1 yapılmak üzere mavi ile belirtildiği görülmektedir.

Basit genişleme algoritmasının pseudo kodu aşağıdadır;

- 1) *Görüntünün başlangıç ve bitiş koordinatını bul, (X_0, Y_0) , (X_n, Y_n) ;*
- 2) *Başlangıç koordinatından bitiş koordinatına kadar pikselleri yapısal eleman ile karşılaştır;*
- 3) *Yapısal elemanın orta hücresi nesnenin üzerinde ise yapısal elemanın üzerinde bulunduğu tüm pikselleri 255 yap;*

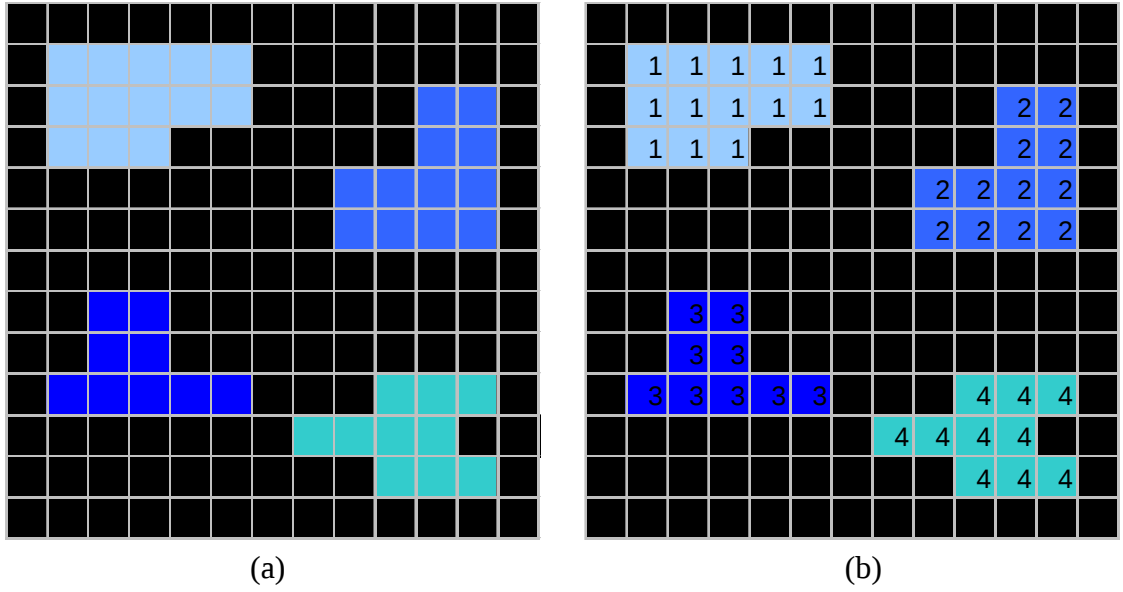
3.2.1.8. Etiketleme

Binary bir resim içerisinde birbiriyle hiçbir komşuluğu olmayan nesnelerin değişik renklere boyanarak birbirlerinden ayrılmasına etiketleme denir (JAIN ve Ark., 1995; ORAL, 1996).

Etiketleme işlemi, binary resmin (0,0) noktasından başlayarak piksel piksel taranması ile yapılır. Binary resim içinde nesneler beyaz, zemin siyah renktedir. Tarama sırasında beyaz piksele (nesneye) rastlandığında bu pikselin 3x3 matris şeklindeki piksel komşularına bakılır. Bu piksel komşuları arasında önceden etiketlenmiş bir piksele rastlanırsa beyaz piksele de aynı etiket değeri verilir. Beyaz pikselin komşulukları arasında önceden etiketlenmiş bir piksel değeri yoksa, son etiketleme değeri bir arttırılarak beyaz piksele verilir ve etiketleme tüm resme uygulanır (Şekil 3.18). Etiketleme işlemi sonucunda resimdeki nesneler farklı renge boyanır ve resimdeki renkler sayıldığında nesne sayısı bulunur. Bir pikselde Red-Green-Blue üç renk katmanı olduğu ve her renk katmanı 0-255 arasında değer aldığına göre bir resimde toplam olarak,

$$\begin{matrix} R & G & B \\ 255 \times 255 \times 255 & = & 16.581.375 \end{matrix} \quad (3.5)$$

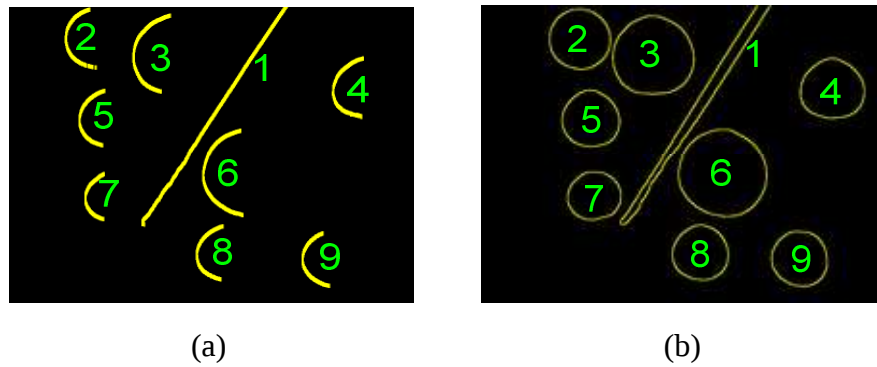
adet nesne sayılabilir.



Şekil 3.18. a) Etiketlenmiş resim b) Etiketleme sırası

Basit etiketleme algoritmasının pseudo kodu aşağıdadır;

- 1) Görüntünün başlangıç ve bitiş koordinatını bul, (X_0, Y_0) , (X_n, Y_n) ;
- 2) Başlangıç koordinatından bitiş koordinatına kadar pikselleri tara;
- 3) Eğer piksel değeri 255 ise, komşu piksel değerlerine bak;
- 4) Komşu pikseller arasında etiketlenen varsa, pikselide aynı değer ile etiketle, komşu pikseller arasında etiketlenen yoksa en son etiket değerini piksele ver;
- 5) Etiket değerini bir arttır;



Şekil 3.19. a) Sobel x, b) Sinc filtrelerinin etiketleme sonrası

Etiketleme yapılan resimdeki renkler sayılarak birbirine piksel komşuluğu olmayan nesnelerin sayısı bulunur. Etiketleme işleminde R, G, B değerleri sırasıyla 1, 1

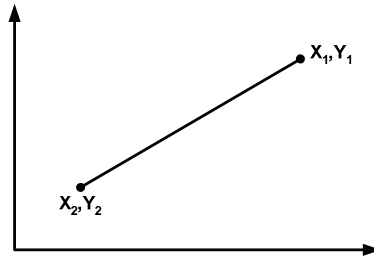
ve 0 verilerek başlandığından nesneler sarı rengin tonları olarak görülmektedir (Şekil 3.19). Bu işlem üst satırdan alt satıra ve soldan sağa doğru yapıldığından nesnelerin numaralandırılması bu sıraya göre belirlenmiştir.

3.2.1.9. Engellerin Tespiti

Dairsel kesitli nesneler arasındaki engelleri tespit etmek için iki yöntem denenmiştir. Birinci yöntem sobel filtreleri sonucu oluşan resimlerdeki eğimlerden engellerin tespiti diğer yöntem ise sinc filtreleme sonucu oluşan resimlerden engellerin tespitidir.

3.2.1.9.1. Eğim ile Engellerin Tespiti

Nesnelerin eğim değerleri, görüntü işleme algoritmalarında dairesel ve çizgisel nesnelerin bulunmasında kullanılır. Etiketleme işlemi sonunda farklı renklerdeki nesnelerin belirlenen noktalar arasındaki eğim değerleri yatay ve dikey düzleme bulunur (GONZALEZ ve WOODS, 1992; JAIN ve Ark., 1995). Sobel x ve Sobel y filtrelerinden geçirilerek etiketlenen resimdeki her nesne yatay ve dikey düzlemde eğimleri eşitlik (3.6) ile bulunur.



Şekil 3.20. Örnek bir doğru parçası

$$m = \frac{x_1 - x_2}{y_1 - y_2} \quad (3.6)$$

m=eğim, (x₁,y₁)=başlangıç koordinatı, (x₂,y₂)=bitiş koordinatı

Şekil 3.20’de örnek bir doğru parçası görülmektedir. Bu doğru üzerinden alınan 2 noktanın koordinatları (3.6) eşitliğinde yerine yazılarak eğim değeri bulunur.

Çizelge 3.1. Nesnelere ait eğim değerleri

		1.Nesne	2.Nesne	3.Nesne	4.Nesne	5.Nesne	6.Nesne	7.Nesne	8.Nesne	9.Nesne
+10 Piksellik Artışlar	1	-60,95	-20,56	-22,25	-24,23	-21,37	-19,09	-29,36	-25,35	-19,80
	2	-56,31	-39,29	-32,74	-36,87	-39,29	-30,96	-41,99	-36,87	-41,99
	3	-56,31	-56,31	-45,00	-45,00	-52,13	-45,00	-52,13	-52,13	-56,31
	4	-60,95	-60,95	-48,37	-56,31	-56,31	-52,13	-66,04	-60,95	-60,95
	5	-56,31	-71,57	-56,31	-66,04	-71,57	-60,95	-77,47	-71,57	-77,47
	6	-60,95	-83,66	-66,04	-71,57	-77,47	-66,04	0,00	-77,47	-83,66
	7	-56,31	0,00	-77,47	-83,66	0,00	-71,57	77,47	0,00	83,66
	8	-56,31	83,66	-77,47	0,00	83,66	-77,47	60,95	77,47	71,57
	9	-60,95	77,47	-83,66	77,47	66,04	-77,47	52,13	66,04	60,95
	10	-56,31	60,95	0,00	71,57	60,95	-83,66	30,96	52,13	52,13
	11	-56,31	52,13	77,47	48,37	45,00	0,00		32,74	34,70
	12	-60,95	39,29	71,57	39,29	27,90	77,47			18,43
	13	-56,31	17,24	66,04	17,82		71,57			
	14	-56,31		66,04			60,95			
	15	-66,04		56,31			56,31			
	16	-56,31		36,87			48,37			
	17	-60,95		17,24			36,87			
	18	-60,95					34,70			
	19	-56,31					16,19			
	20	-56,31								
	21	-60,95								
	22	-60,95								

Her nesne için bulunan eğimler incelendiğinde bazı nesnelerin eğim değerlerinde işaret (+, -) olarak değişim görülür ve bu nesneler dairesel kesitli nesne olarak tanımlanır. Eğim değerlerinde işaret olarak değişim görülmeyen nesneler ise engeldir. Bu yöntemle engeller ile dairesel kesitli nesneler birbirinden ayrılarak engellerin başlangıç ve bitiş koordinatları tespit edilir. Çizelge 3.1’de nesnelere ait 10 piksel aralıkta bulunan eğim değerleri incelendiğinde 1. nesne hariç tüm nesnelerin eğim değerler işaretlerinin değiştiği görülmekte ve bu durum 1. nesnenin engel olarak tanımlanmasını sağlar. Örneğin dairesel kesitli nesne olan 2. nesnenin verilerine bakıldığında – değerden + değere doğru bir değişim görülür. Çizelge 3.1’de nesnelerin eğim sayılarının farklı olması nesnelerin boyutları ile doğru orantılı olarak

değişmektedir. 3. nesnenin 17 tane eğim değeri bulunurken 7. nesnenin 10 tane eğim değeri bulunuyor. Bu da 3. nesnenin çapının 7. nesneye göre daha büyük olduğunu gösterir.

3.2.1.9.2. Çevre ve Alandan Engel Tespiti

Engelleri tespit etmek için kullanılan diğer bir yöntemde nesnelerin çevre ve alan bağıntısında yapılır.

Dairesel kesitli nesnelerin çevre ve alanları sırasıyla eşitlik (3.7) ve (3.8)'de verilmiştir. Çevre ve alan eşitlikleri (3.9)'da yerine yazıldığında bu oranın dairesel kesitli nesnelerde yaklaşık olarak 1'dir. Diğer nesnelerde ise 1'den farklı sonuç çıkmaktadır.

$$Ç = 2\pi r \quad (3.7)$$

$$A = \pi r^2 \quad (3.8)$$

$$K = \frac{Ç^2}{4\pi A} = \frac{(2\pi r)^2}{4\pi r^2} = \frac{4\pi^2 r^2}{4\pi^2 r^2} \cong 1 \quad (3.9)$$

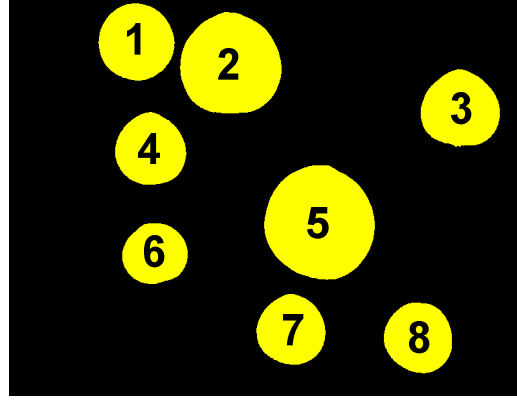
Ç= çevre, A=alan, r = yarıçap

Sinc filtrelemesi ile oluşan resme etiketleme işlemi uygulanarak her nesnenin çevresi bulunur. Erozyon işleminde sonra oluşan binary resime uygulanan etiketleme ile de nesnelerin alanları bulunur. Bulunan bu çevre ve alan değerleri eşitlik (3.9)'da yerine yazılır ve sonucu 1'den büyük olan nesneler engel olarak tanımlanır.

3.2.1.10. Dairesel Kesitli Nesneleri Etiketleme ve 2 boyutlu Koordinat Bulma

Çevre-Alan oranı veya eğim değerleri ile başlangıç ve bitiş koordinatları bulunan engeller, erozyon işlemi sonucu elde edilen binary resim içinde, zemin rengine boyanır ve tekrar yapılan etiketleme ile resim içindeki dairesel kesitli nesnelerin sayısı

bulunur. Şekil 3.21’de görüldüğü gibi nesneler, sarı rengin tonları kullanılarak etiketleme yapılmıştır. Etiketleme işlemi üst satırdan alt satıra doğru yapıldığından nesnelerin numaralandırılması satır sırasına göre belirlenmiştir.



Şekil 3.21. Dairesel kesitli nesnelerin etiketlenmesi

Etiketlemeleri yapılan dairesel kesitli nesnelerin iki boyutlu koordinatlarının bulunması (3.10) nolu eşitlikte gösterildiği gibi nesnenin x ve y koordinatların toplamı, nesnenin toplam piksel sayısına (kapladığı alana) bölünerek ağırlık merkezi koordinatları hesaplanır (JAIN ve Ark., 1995).

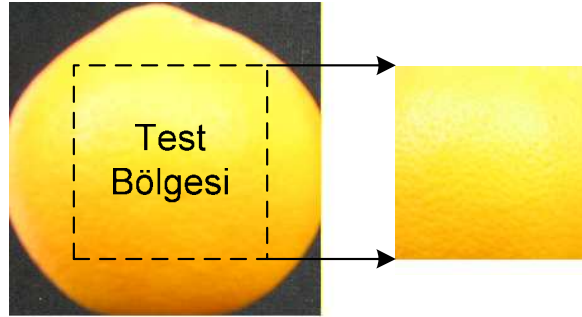
$$x = \frac{\sum_i^n \sum_j^m jB[i, j]}{\sum_i^n \sum_j^m B[i, j]} \quad y = \frac{\sum_i^n \sum_j^m iB[i, j]}{\sum_i^n \sum_j^m B[i, j]} \quad (3.10)$$

i,j = Nesne başlangıç koordinatları, B = Etiketlenen resim,
n,m = Nesne bitiş koordinatları

3.2.1.11. Renk Sayımı ile Derinlik Bulma

Bir resimde bulunan renk sayısı resmin netliği hakkında bilgi verir. Resim ne kadar net olursa o kadar fazla renk bulunur (SHARMA ve TRUSSELL, 1997;

DEODHAR ve TRAGOUDAS, 2001; BALA ve SHARMA, 2005). 24 bit derinliğe sahip bir resimde $2^{24}=16,7$ milyon renk bulunabilir. Çalışmada derinliği bulmak için odak mesafesi manuel olarak ayarlanabilen bir dijital fotoğraf makinesi kullanılarak farklı odak mesafelerinde resimler çekilmiştir. Çekilen resimlere görüntü işleme tekniklerinden gri seviye dönüşüm, eşikleme, erozyon ve etiketleme işlemleri sırası ile uygulanarak nesnelerin ağırlık merkezlerinin (x, y) koordinatları bulunmuştur. Sonra çekilen resimlerdeki nesnelerin ağırlık merkezinde (x, y) bir test bölgesi seçilmiştir. Seçilen bu test bölgesi nesnelerin kenarlarından taşmayacak şekilde ve her nesnenin alanı ile orantılı olarak değişen bir karedir (Şekil 3.22’de görüldüğü gibi). Her nesne için alınan test bölgesi ile farklı odak uzaklıklarında çekilmiş resimler üzerinde renk sayıları kontrol edilmiş ve renk sayısı en fazla olan resimdeki nesnenin çekilen odak mesafesine göre derinliği (z) hesaplanmıştır. Şekil 3.23’de n adet farklı odak uzaklıklarında çekilen fotoğraflar, 1. nesnenin seçilen bir test bölgesindeki renk sayım grafiği ve bu grafiğe bağlı olarak en fazla renk sayısının 11. fotoğrafa ait olduğu görülmektedir.

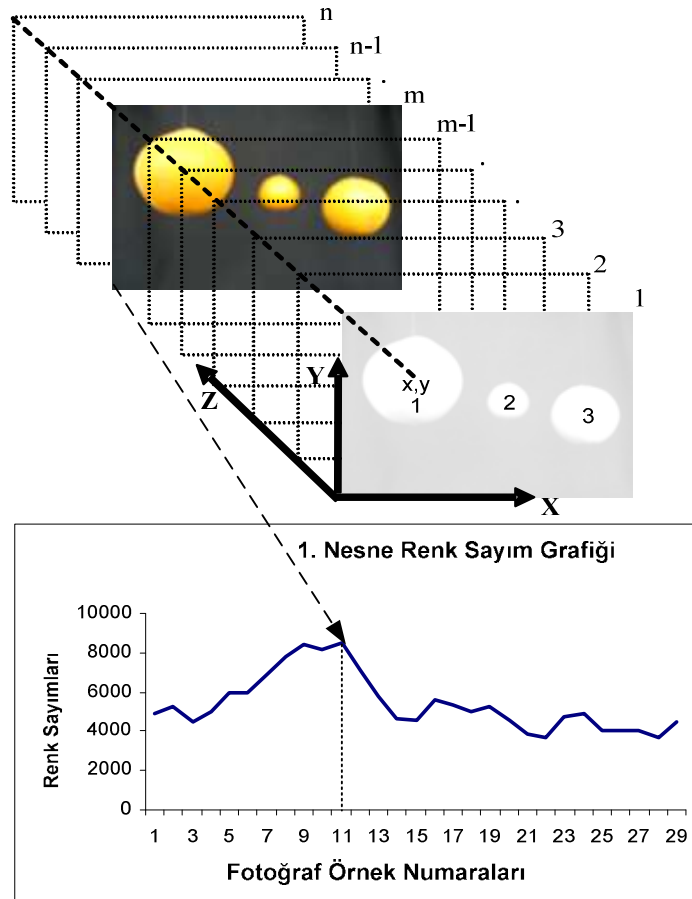


Şekil 3.22. Nesne ve seçilen test bölgesi

Çizelge 3.2’deki odak mesafelerine göre 11. fotoğrafın hangi aralıkta olduğuna bakılarak nesnenin derinliği bulunur. Bu işlem resimdeki tüm nesnelerin test bölgelerine ayrı ayrı uygulanarak her nesnenin derinliği hesaplanır.

Çizelge 3.2. Farklı odak mesafelerinde çekilen fotoğraflar ve mesafeleri

Fotoğraf No	Mesafe
1	5 cm
5	10 cm
11	20 cm
17	50 cm
22	100 cm
24	200 cm
26	500 cm
28	1000 cm
29	Sonsuz



Şekil 3.23. Farklı odak mesafelerinde çekilen resimler ve renk sayım grafiği

3.2.2. Genetik Algoritma

GA, biyolojik organizmaların genetik işlemlerine dayanır ve bu işlemleri taklit eder. GA, Darwin'in en iyinin hayatta kalması (survial of the fittest) fikrine dayalıdır.

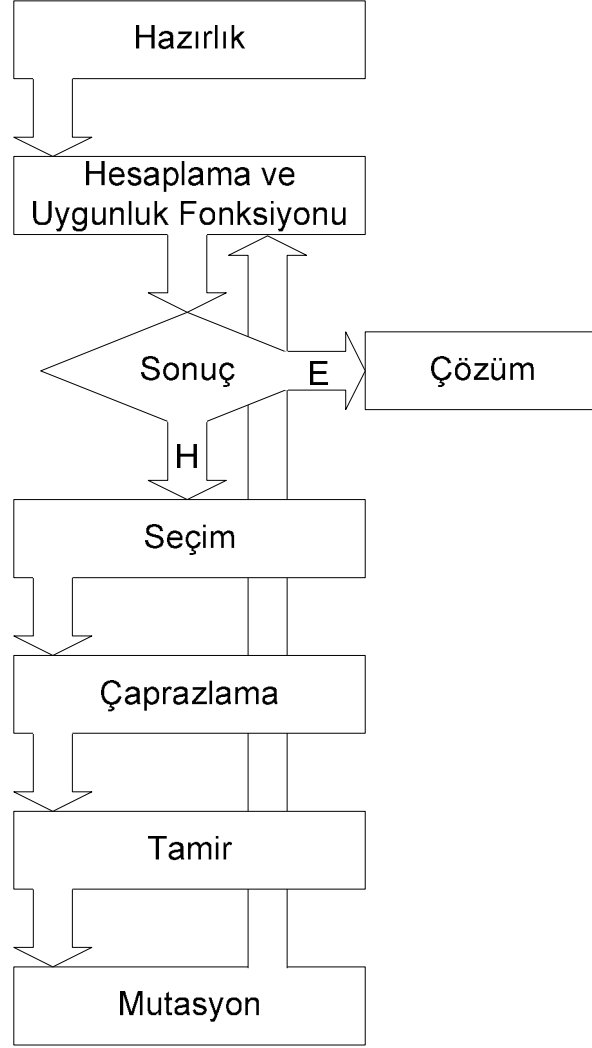
GA'nın diğer optimizasyon ve arama yöntemlerinden farklılıkları şu şekilde sıralana bilir:

- 1) GA optimize edilecek parametrelerin kodlarıyla çalışır, parametrelerin kendilerini kullanmaz.
- 2) GA, arama uzayında bireylerin uygunluk değerini bulmak için sadece sonuç bilgisini (amaç fonksiyonu) kullanır, türevleri veya diğer yardımcı bilgileri kullanmaz.
- 3) GA, sadece bir arama noktasını değil, bir popülasyon grubundaki noktaları arar. Bir tek yerden değil bir grup çözüm içinden arama yapar.
- 4) GA ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir.
- 5) Olasılık kurallarına göre çalışırlar. Programın ne kadar iyi çalıştığı önceden kesin olarak belirlenemez.
- 6) GA, çözümlerden oluşan popülasyonu eş zamanlı incelediği için lokal maksimumlara takılmamasıdır.

Bu farklılıkları nedeniyle GA etkin arama yaparak çözüme çabuk ulaşır.

3.2.2.1. GA'nın Akış Diyagramı

Koordinatları bulunmuş ve erişim sırası belirlenecek nesnelerin, üç boyutlu Gezgin Satıcı Problemindeki gibi noktalar arasında en kısa rotanın bulunması gerekmektedir. Bu tür problemler kombinatorial optimizasyon problemleri olarak adlandırılırlar ve GA ile diğer optimizasyon tekniklerinden daha başarılı çözümler ürettiği için robot kol hareket optimizasyonunda GA kullanılmıştır (FLOREANO ve MONDADA, 1998; MAN ve Ark.,1996).



Şekil 3.24. GA'nın akış diyagramı

GA Darwin'in, en iyinin hayatta kalması fikrine dayanarak sürekli iyileşen çözümler üretir. (GOLDBERG, 1989). Bunun için "iyi" nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için çaprazlama, mutasyon gibi operatörleri kullanır. GA'nın akış diyagramı Şekil 3.24'de gösterilmiştir

GA, hazırlık ile başlayan hesaplama işlemi sonrasında belirlenen hedefe ulaşım ulaşmadığını test eder. Belirlenen hedefe ulaşılmamış ise seçim algoritmasını, çaprazlama ve mutasyon operatörlerini çalıştırır. Sonuca ulaşıncaya kadar bu işlemler devam eder.

3.2.2.2. Hazırlık

Bu bölümünde görüntü işleme teknikleriyle elde edilen x, y, z koordinat bilgileri GA'nın kromozom bilgilerini ve çözüm grubunu oluşturmaktadır. Her bir çözüm grubu bir kromozomdur. Kromozomlar ihtiyaca göre istenilen sayı tabanında, reel sayılarla veya alfabedeki harflerle oluşturulabilir.

Aşağıda GSP ile tezdeki problemin benzemesi nedeni ile örnek olarak 8 şehir için 4 kromozomdan oluşan çözüm grupları isim ve ikilik gösterilmiştir (Çizelge 3.3, 3.4, 3.5)

Çizelge 3.3. Şehir isimleri düzeninde çözüm grubu

1.Kromozom	Mersin	Ankara	İstanbul	Kayseri	Hatay	Adana	Antalya	İzmir
2.Kromozom	Kayseri	İzmir	Hatay	Adana	Antalya	Mersin	İstanbul	Ankara
3.Kromozom	Antalya	Adana	Mersin	İzmir	Kayseri	Hatay	Ankara	İstanbul
4.Kromozom	Hatay	Ankara	İzmir	Kayseri	Antalya	İstanbul	Mersin	Adana

Çizelge 3.4. İkilik düzende çözüm grubu

1.Kromozom	111	001	101	100	011	000	010	110
2.Kromozom	100	110	011	000	010	111	101	001
3.Kromozom	010	000	111	110	100	011	001	101
4.Kromozom	011	001	110	100	010	101	111	000

Çizelge 3.5. Görüntü işleme sonucu GA'ya aktarılan dairesel kesitli nesnelerin x,y,z koordinatları

	X	Y	Z
1. Nesne	27	417	20
2. Nesne	127	428	12
3. Nesne	95	345	16
4. Nesne	540	251	50
5. Nesne	362	386	20
6. Nesne	164	300	25
7. Nesne	70	379	16
8. Nesne	84	467	30
9. Nesne	431	670	45
10. Nesne	240	370	18

Çizelge 3.5’de görüntü işleme sonucu GA’ya aktarılan dairesel kesitli nesnelerin üç boyutlu (x, y, z) koordinatları gösterilmiştir. Tezde kullanılan örnek çözüm grubundaki kromozomlar nesnelerin üç boyutlu koordinatlarından oluşturulmuştur. Her nesnenin (x, y, z) koordinatı kromozomda bir bilgiyi (geni) göstermektedir (Çizelge 3.6). Çizelge 3.5’te verilen 10 tane nesnenin koordinatları ile bir kromozom oluşturulmaktadır. Çizelge 3.7’de tezde kullanılmak üzere hazırlanmış örnek kromozomlar görülmektedir.

Çizelge 3.6. Kromozom yapısı

1. Gen	2. Gen	3. Gen	4. Gen		n. Gen
95 345 16	540 251 50	240 370 18	84 467 30		X _n Y _n Z _n

Çizelge 3.7. On tane örnek kromozom

1.Kromozom	3	4	10	8	1	9	6	5	7	2
2.Kromozom	5	6	9	4	8	7	3	1	2	10
3.Kromozom	2	4	1	9	3	6	10	7	5	8
4.Kromozom	1	5	6	2	7	8	9	3	10	4
5.Kromozom	9	6	4	3	7	8	2	10	1	5
6.Kromozom	3	7	8	10	2	1	5	6	4	9
7.Kromozom	2	5	8	1	10	9	3	7	6	4
8.Kromozom	4	8	7	2	3	6	10	1	5	9
9.Kromozom	9	7	5	1	3	2	8	10	4	6
10.Kromozom	1	10	2	4	9	5	6	8	3	7

3.2.2.3. Hesaplama ve Uygunluk fonksiyonu

Her kromozom için noktalar arasındaki toplam mesafenin hesap edildiği fonksiyondur. Uygunluk fonksiyonu ile çözümün, amaca ne kadar yakın olduğu ölçülür. Çalışmada üç boyutlu koordinatlar arasındaki toplam mesafenin bulunması için i tane genden oluşan bir kromozomun Uygunluk fonksiyonu (3.11) nolu eşitlikle hesaplanmaktadır. 4 genden oluşan kromozom için uygunluk fonksiyonunun açılmış şekli (3.12) eşitliğinde gösterilmiştir.

Uygunluk fonksiyonu=

$$\sum_{n=1}^{i-1} \sqrt[3]{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2 + (z_{n+1} - z_n)^2} + \sqrt[3]{(x_1 - x_i)^2 + (y_1 - y_i)^2 + (z_1 - z_i)^2} \quad (3.11)$$

Uygunluk Fonksiyonu=

$$\sqrt[3]{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} + \sqrt[3]{(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2} + \sqrt[3]{(x_3 - x_4)^2 + (y_3 - y_4)^2 + (z_3 - z_4)^2} + \sqrt[3]{(x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2} \quad (3.12)$$

Bu fonksiyonların hassas ve verimli çalışması GA'nın sonuçlarını belirler. Bu yüzden GA'nın başarısı uygunluk fonksiyonunun iyi çalışmasına bağlıdır.

3.2.2.4. Seçim

Hesaplama basamağında uyumlulukları hesaplanan kromozomlardan bazıları seçim yöntemiyle elenirler. Seçim yöntemleri her zaman en kötü sonuçları elemeyebilir. Bu işlemin amacı, daha iyi özelliğe sahip yeni nesiller üretmek ve arama algoritmasının alanını genişletmektir.

Başlıca iki seçim yöntemi vardır.

Turnuva Metodu: Turnuva metodunda kromozomların özellikleri karşılaştırılır ve en iyi kromozom seçilir. Birbiriyle yarıştırlan kromozom sayısı azaldıkça GA daha verimli çalışır. En iyi sonucu elde etmek için kromozomlar birbirleriyle n sayıda ($n=2,3,4,\dots$) yarıştırlırlar, n sayısı arttıkça çözümdeki çeşitlilik azalır. Bu durumun oluşmaması için n sayısı genellikle 2 seçilir.

Basitçe çalışma sistemi 3 adımdan oluşur.

- 1) *Rasgele n tane kromozom seçilir.*
- 2) *Seçilen bu n kromozomdan en iyi uygunluk fonksiyonuna sahip kromozom seçilir.*
- 3) *1. ve 2. adımlar yeni jenerasyondaki kromozom sayısı eski jenerasyona ulaşınca kadar devam edilir (REN ve Ark, 2000).*

Yeni oluşan jenerasyonun içinde aynı kromozomdan birden fazla gelebilir. Turnuva metodunda uygunluk değeri yüksek olan kromozom seçilir (REN ve Ark., 2000).

Rulet Teker Metodu: En kötüye de şans tanıyan seçim yöntemine örnek olarak rulet tekeri modelini gösterilir. Rulet tekeri modelinde çözümler uygunluklarına göre bir dairenin dilimlerine yerleştirilirler ve rulet tekerinin dönmesine benzer biçimde gelişigüzel olarak dairenin dilimlerinin üstündeki çözüm gruplarının seçilme olasılığı hesaplanır.

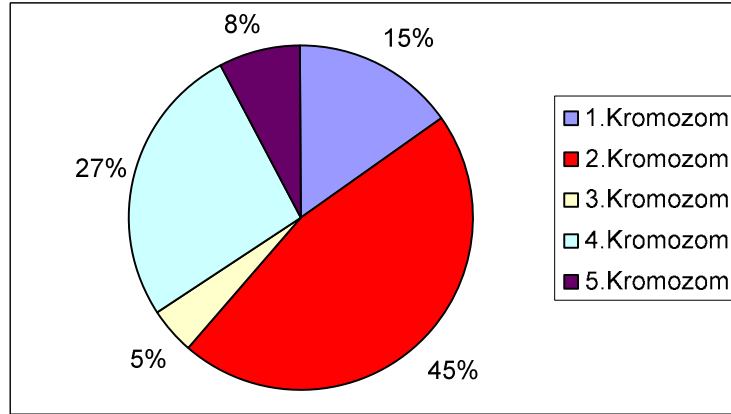
Rulet tekeri seçim algoritması aşağıdaki gibidir:

- 1) *Çözüm grubundaki bütün kromozomların uygunluk değerini hesapla,*

- 2) (0, toplam uygunluk değeri) sayı aralığında rasgele bir sayı seç,
- 3) Seçilen sayıyı kromozomların uygunluk değerleriyle karşılaştır, eğer kromozomun uygunluk değeri rasgele seçilen sayıdan büyükse o kromozomu seç.
- 4) Yeni jenerasyondaki kromozom sayısı eski jenerasyona ulaşınca kadar 2. adımdan devam edilir.

Çizelge 3.8. Örnek Kromozomlar ve uygunluk değeri yüzdeleri

	Uygunluk	%
1.Kromozom	209	15
2.Kromozom	621	45
3.Kromozom	64	5
4.Kromozom	361	27
5.Kromozom	106	8
Toplam	1361	100



Şekil 3.25. Rulet tekeri modeli

Çizelge 3.8’de 5 kromozoma ait uygunluk değerleri ve uygunluk değerlerinin toplamı görülmektedir. Bu toplama göre her kromozomun %’lik değerleri hesaplanmıştır. Şekil 3.25’de beş kromozomun seçilme olasılıkları yüzde olarak

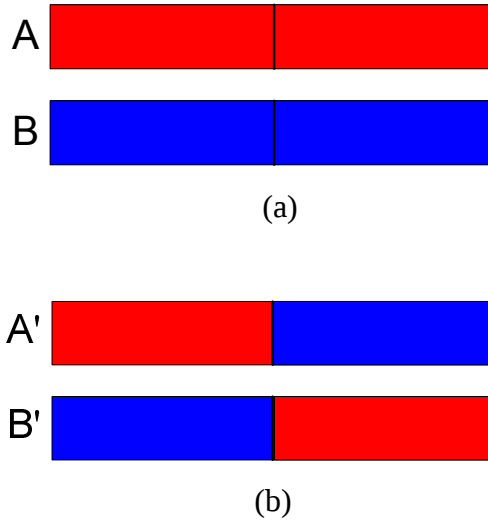
verilmiştir. Rulet tekerinde seçilme ihtimali en yüksek olan kromozom 2. kromozom olmasına rağmen bu kromozom seçilmeyebilir ve seçilme olasılığı en düşük olan 3. ve 5. kromozomlar seçilebilirler (GOLDBERG, 1989).

3.2.2.5. Çaprazlama

Çaprazlamada, kromozom genlerinin yerini değiştirerek çocuk (child) kromozomlar üretmek ve böylece varolan uygunluk değeri yüksek olan kromozomlardan, uygunluk değeri daha yüksek olan kromozomlar elde etmek amaçlanmaktadır.

Seçim basamağında elenmeyen kromozomlar çaprazlama (crossover) yapmak için seçilirler. İki tür seçim yöntemi vardır: kromozomlar ikişer ikişer seçilirler veya seçim tüm popülasyondan seçilerek yapılır. Çaprazlama önceden elde edilmiş bilgilerin yeni gruplara aktarılmasını sağlar.

Tek Noktalı Çaprazlama: Kromozomlar rasgele seçilen bir kesme noktası ile ayrılıyorsa buna tek noktalı çaprazlama denir. Şekil 3.26'de tek noktalı çaprazlama yapan A, B kromozomları ve çaprazlama yaptıktan sonra oluşan A', B' kromozomları gösterilmiştir.



Şekil 3.26. a) Tek noktalı çaprazlama yapılacak kromozomlar, b) Tek noktalı çaprazlama sonucu oluşan kromozomlar

						*				
A Kromozomu	3	4	10	8	1	9	6	5	7	2
B Kromozomu	5	6	9	4	8	7	3	1	2	10

Şekil 3.27. Çaprazlama yapılacak kromozomlar

Bu çalışmada tek noktalı çaprazlama kullanılmıştır. Çaprazlanacak kromozomlar turnuva veya rulet tekeri ile seçilmiştir. Örnek seçilen kromozomlar Şekil 3.27’de görülmektedir. A ve B kromozomları rastgele seçilen bir noktadan (Şekil 3.27’de * ile gösterilen noktadan) ikiye bölünmüştür. A kromozomun birinci parçasında 3, 4, 10, 8, 1, 9 ikinci parçasında ise 6, 5, 7, 2 olacak şekilde ikiye ayrılmıştır. B kromozomu da birinci parçasında 5, 6, 9, 4, 8, 7 ve ikinci parçasında 3, 1, 2, 10 şeklinde bölünmüştür. Bu rastgele seçilen noktadan bölünen kromozomlar çaprazlanarak Şekil 3.28’te görülen yeni kromozomlar oluşturulmuştur.

A' Kromozomu	3	4	10	8	1	9	3	1	2	10
B' Kromozomu	5	6	9	4	8	7	6	5	7	2

Şekil 3.28. Tek noktalı çaprazlama sonucu oluşan kromozomlar

Şekil 3.28’te yeni oluşturulan A' kromozomu incelendiğinde 3, 10, 1 genlerinin B' kromozomunda ise 5, 6, 7 genlerinin iki defa tekrarlandığı ve A' kromozomunda 5, 6, 7 genlerinin, B' kromozomunda ise 3, 10, 1 genlerinin bulunmadığı görülmektedir. Yeni oluşan kromozomlardaki bu tekrarlar ve eksiklikler Gezgin Satıcı Probleminde bir şehre iki defa gitme ve bazı şehirlere ise hiç gidilmemesi istenmeyen bir sonuçtur ve bunun düzeltilmesi gerekmektedir.

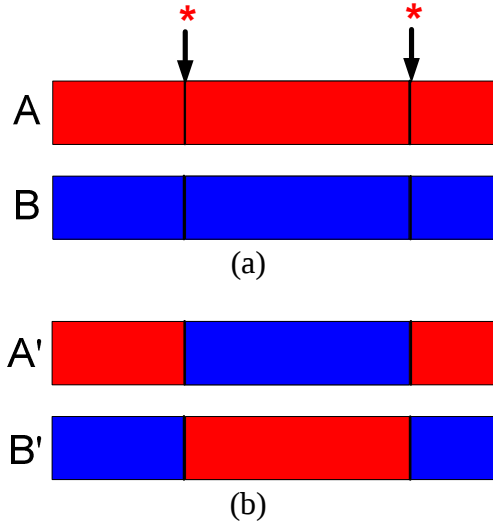
Bu sonuçları düzeltmek ve GSP problemine uygun hale getirmek için tamir operatörü kullanılmıştır.

A' Kromozomu	3	4	10	8	1	9	5	6	2	7
B' Kromozomu	5	6	9	4	8	7	1	3	10	2

Şekil 3.29. Tamir operatörü ile düzeltilmiş kromozomlar

Şekil 3.29'da tamir operatörü ile düzeltilmiş kromozomlar görülmektedir. Tamir işlemine iki defa tekrar edilen genlerin yerine eksik olan genlerin en küçük sıradan başlayarak değiştirilmesi ile yapılmıştır.

Çok Noktalı Çaprazlama: Kromozomlar rastgele seçilmiş iki veya daha fazla noktadan parçalanarak yapılan çaprazlamaya çok noktalı çaprazlama denir. Şekil 3.30.a'da rastgele seçilmiş, * ile gösterilen iki noktadan çaprazlama yapılacak A ve B kromozomu, Şekil 3.30.b'de ise çaprazlama sonucu oluşan kromozomlar görülmektedir.

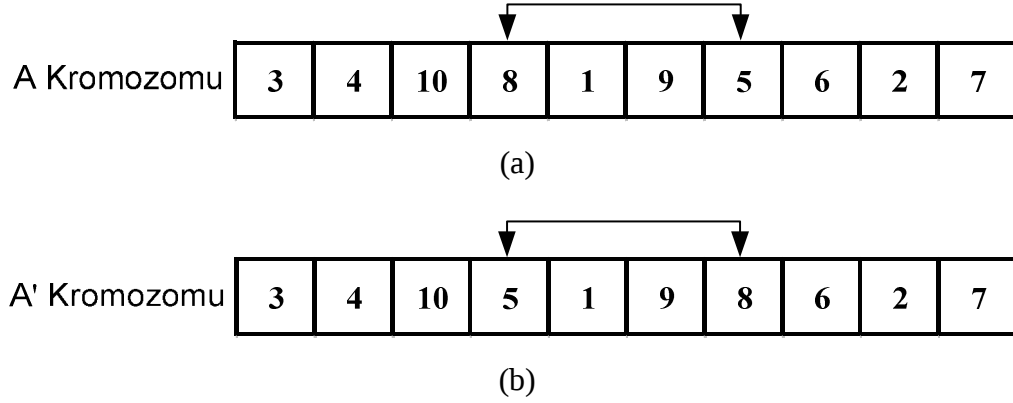


Şekil 3.30. a) İki noktalı çaprazlama yapılacak kromozomlar b) Çaprazlamadan sonra oluşan kromozomlar

3.2.2.6. Mutasyon

Çaprazlama işlemi sonucunda bireyler birbirine benzemeye başlayabilir. Bu durumdan kurtulabilmesi için Mutasyon uygulanması gerekir. Mutasyon işlemi topluluktaki, problemi çözme yeteneği iyi olmayan ve birbirine benzer bireyleri değiştirerek yeni alt çözümler elde etmek için kullanılır (GOLDBERG, 1989; MAN ve

Ark., 1996; KOZEK ve Ark., 1993). Mutasyon işlemi, rastgele seçilen kromozomda rastgele seçilen iki genin yerleri değiştirilerek yapılmıştır. Şekil 3.31.a'da gösterildiği gibi A kromozomda rasgele seçilen oklarla belirtilen genlerin yerleri değiştirilerek Şekil 3.31.b'deki yeni kromozomu A' elde edilmiştir.

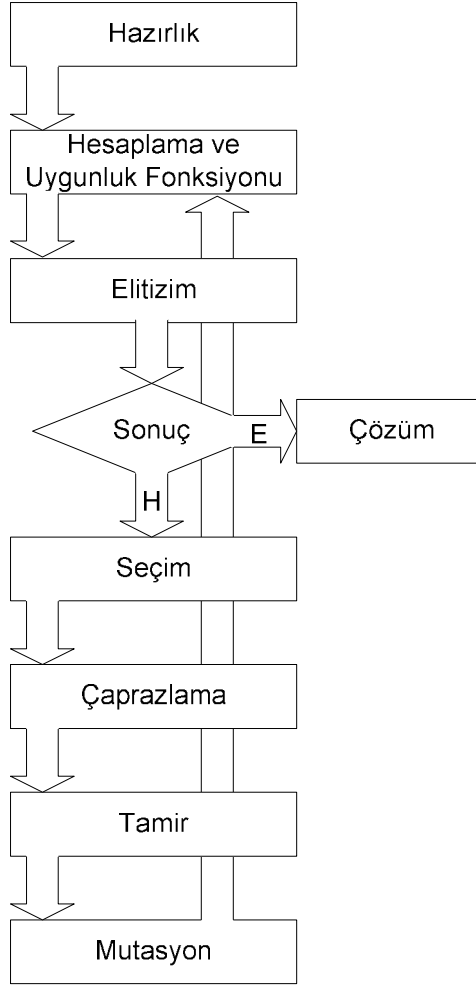


Şekil 3.31. a) Mutasyon öncesi b) Mutasyon sonrası

3.2.2.7. Elitizm

Çaprazlama ve mutasyon sonucunda en iyi uyumluluğa sahip birey seçilmeyebilir. Bunu önlemek için, çaprazlama ve mutasyon işlemlerinden sonra oluşan yeni popülasyonun uygunluk değeri düşük herhangi bir kromozomu ile bir önceki popülasyonun en iyi (elit) kromozomu değiştirilir. Buna elitizm adı verilir. GA'ya elitizm eklendiğinde elde edilen akış diyagramı Şekil 3.32'de gösterilmiştir. Şekil 3.32'deki algoritmada hesaplama ve uyumluluk fonksiyonu tarafından elde edilmiş uyumluluğu en fazla olan kromozom elitizm operatörü tarafından saklanır. Saklanan kromozom çaprazlama ve mutasyon operatörlerinin çalışmasıyla oluşan kromozomlarla karşılaştırılır. Elitizm operatörü tarafından saklanan kromozom uyumluluğu az olan kromozom ile değiştirilerek elitizm yapılır.

Elitizm, GA'nın en iyi bulunan çözümü kaybetmesini önlediğinden performansı artırır.



Şekil 3.32. Elitizm eklenmiş GA algoritması

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu çalışmada resimdeki nesnelere görüntü işleme teknikleri uygulanarak koordinatların bulunması, renk sayımı metodu ile tek kameradan alınan görüntülerden derinliğin hesaplanması ve bulunan üç boyutlu koordinatlar optimizasyon tekniklerinden olan GA'ya aktarılarak nesnelere en kısa yoldan erişim sırasının belirlenmesi yapılmıştır.

Renk sayımı yapılacak test bölgesinin belirlenmesi için araştırmalar yapılmış ve 640x480 ile 800x600 çözünürlükte çekilmiş fotoğraflardaki nesnelerin test bölgelerindeki renk sayıları karşılaştırılmıştır.

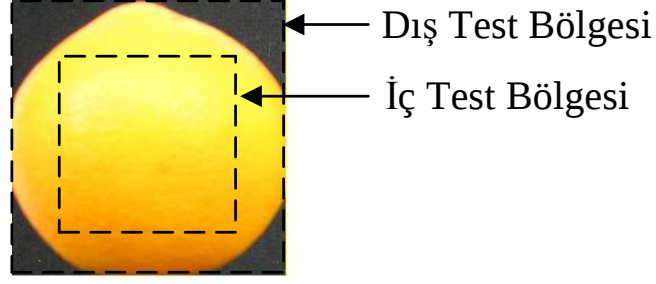
Derinlik analizi aynı resimdeki birden fazla nesne için yapılabilirliği araştırılmıştır.

Bulunan üç boyutlu koordinatlar GA'ya aktarılmış ve en uygun hareket optimizasyonu için elde edilen veriler grafik haline getirilmiştir.

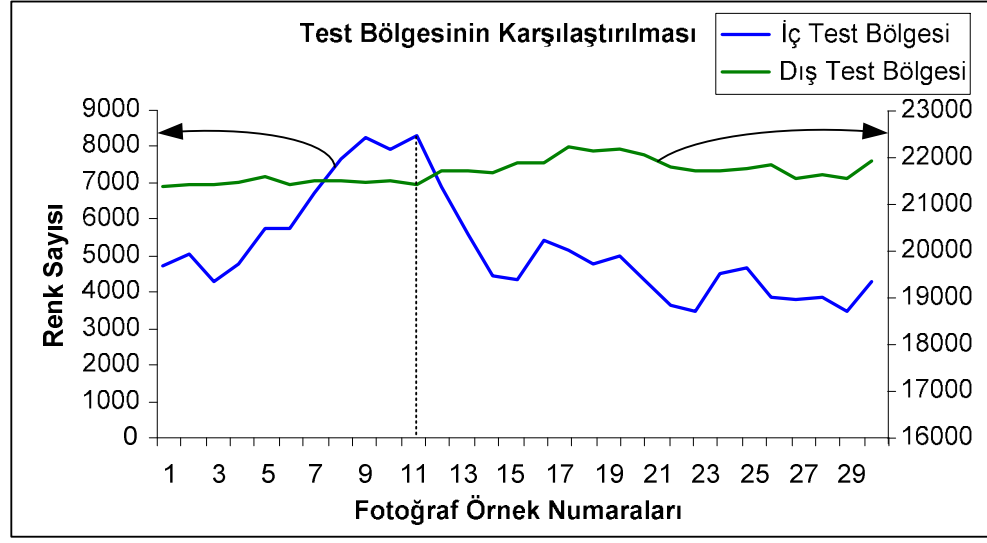
4.1. Test Bölgesi Seçim Analizi

Dijital fotoğraf makinesi ile farklı odak uzaklıklarında çekilen resimlere görüntü işleme teknikleri uygulanarak resimlerdeki dairesel kesitli nesnelerin ağırlık merkezlerinin x, y koordinatları bulunmuştur.

İlk deneylerde nesnenin kenarlarından teğet geçen test bölgesi belirlenerek bu alan içinde renk sayımı yapılmıştır. Alınan sonuçlar incelendiğinde nesnenin kenarlarındaki arka alana ait taşmaların renk sayımına olumsuz etki ettiği görülmüştür. Bu sakıncanın giderilmesi amacıyla nesnenin alanı ile orantılı olarak değişen ve nesnenin kenarından taşmayacak şekilde bir iç test bölgesi belirlenmiştir (Şekil 4.1). Her nesne için seçilen iç test bölgesindeki renk sayımlarında en çok renk sayısına sahip olan fotoğrafın odak mesafesine göre derinliğinin bulunabileceği görülmüştür. Şekil 4.2'de iç ve dış test bölgelerinin renk sayımına etkisinin karşılaştırmalı grafiği görülmektedir.



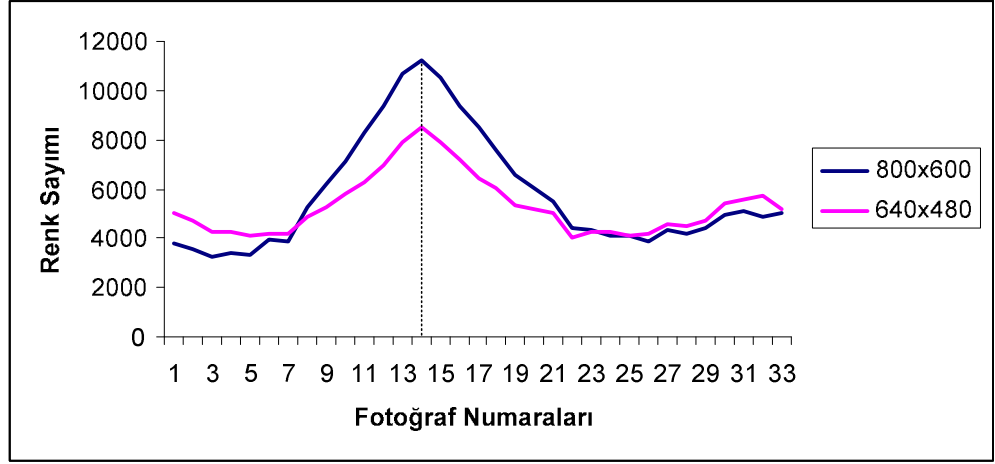
Şekil 4.1. İç ve dış test bölgesi



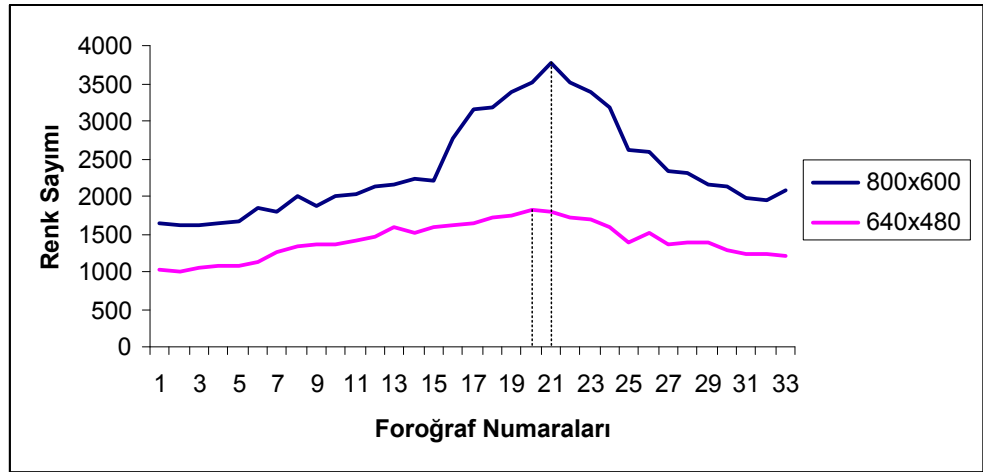
Şekil 4.2. İç ve dış test bölgelerinin renk sayım karşılaştırmalı grafiği

Çalışmada 800x600 ile 640x480 çözünürlükteki resimlerin test bölgelerinde renk sayımları yapılmıştır. Şekil 4.3’de 20 cm mesafede bulunan nesnenin 800x600 ile 640x480 piksellerdeki test bölgelerindeki renk sayıları karşılaştırmalı grafiği görülmektedir. Şekilde görüldüğü gibi test bölgesindeki alana düşen piksel sayısı azaldıkça renk sayısı da azalmaktadır. Bunun sonucu olarak daha küçük çaplı nesnelerde yapılan renk sayımlarına hata oranı artmaktadır. Şekil 4.4’te 50 cm’deki nesnenin 800x600 ve 640x480 çözünürlükteki renk sayım grafiği görülmektedir.

640x480 çözünürlükte çekilen resimler sonucu etkilediği görüldüğünden çalışmada 800x600 çözünürlükteki resimler kullanılmıştır (Şekil 4.4).



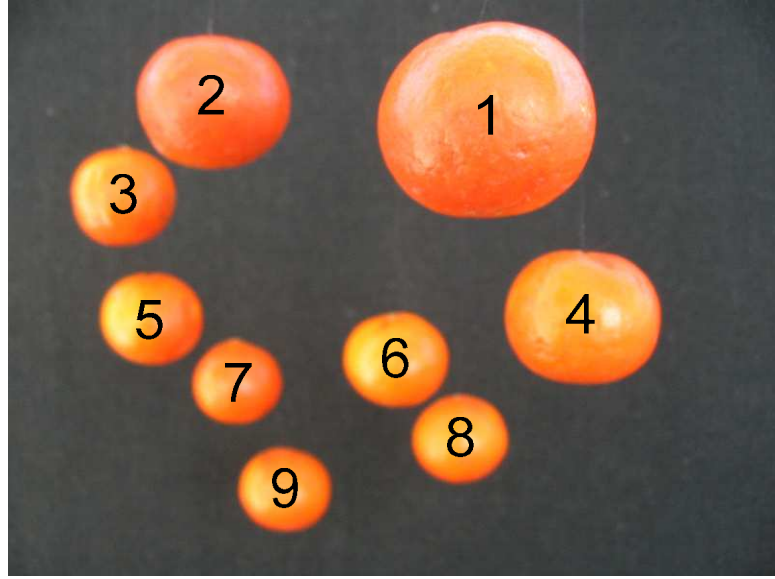
Şekil 4.3. 20 cm mesafedeki nesnenin 800x600 ve 640x480 piksellerdeki renk sayım grafikleri



Şekil 4.4. 50 cm mesafedeki nesnenin 800x600 ve 640x480 piksellerdeki renk sayım grafikleri

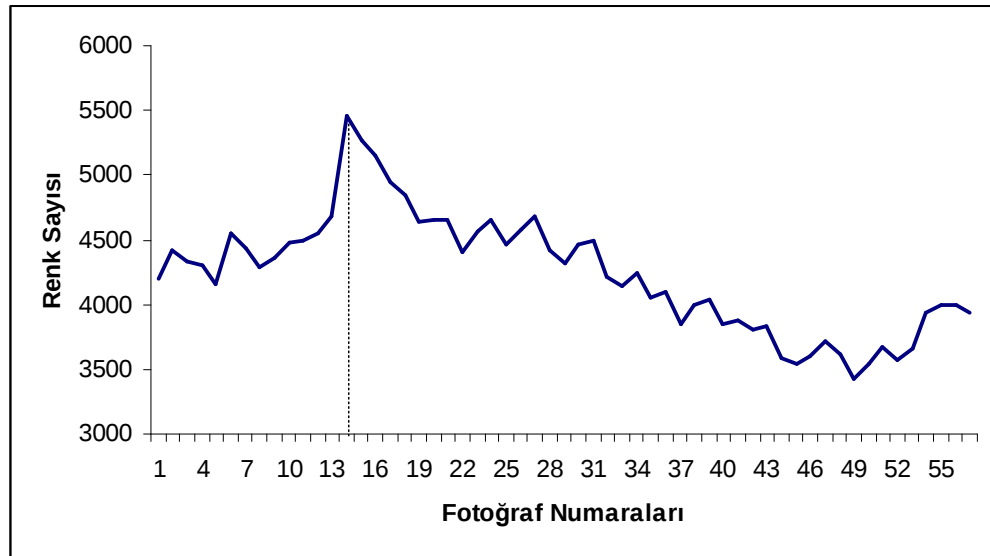
4.2. Dokuz Nesne İçin Derinlik Analizi

Dokuz nesnenin aynı anda derinlik analizi yapılmıştır. Derinlik analizi için farklı odaklarda 57 tane fotoğraf çekilmiştir. Şekil 4.5'teki nesne numaraları satır taraması yapılarak etiketleme ile belirlenmiştir.



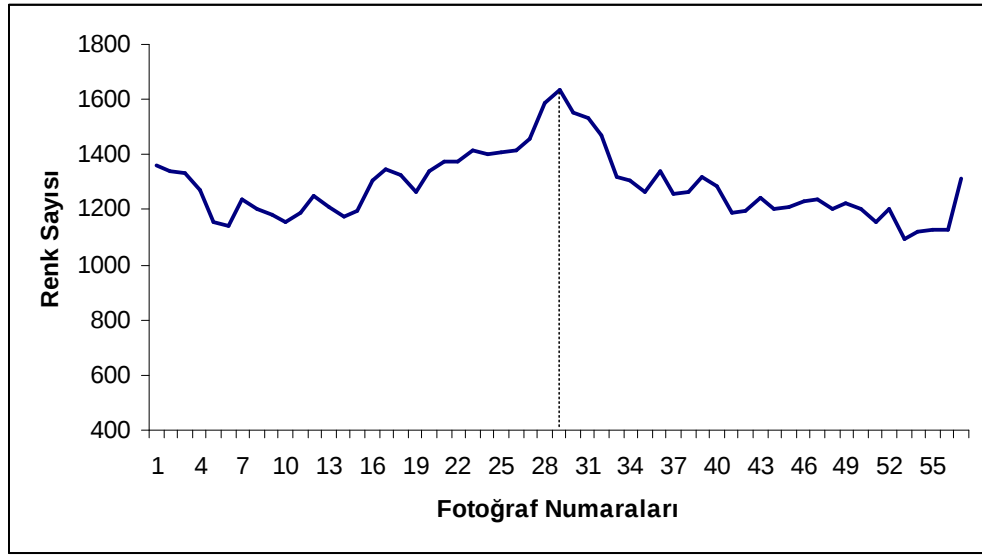
Şekil 4.5. Dokuz nesneli örnek resim

Şekil 4.6’da birinci nesnenin renk sayım grafiği görülmektedir. Renk sayım grafiği incelendiğinde birinci nesnenin 14 nolu fotoğrafta en fazla renk sayısına sahiptir.



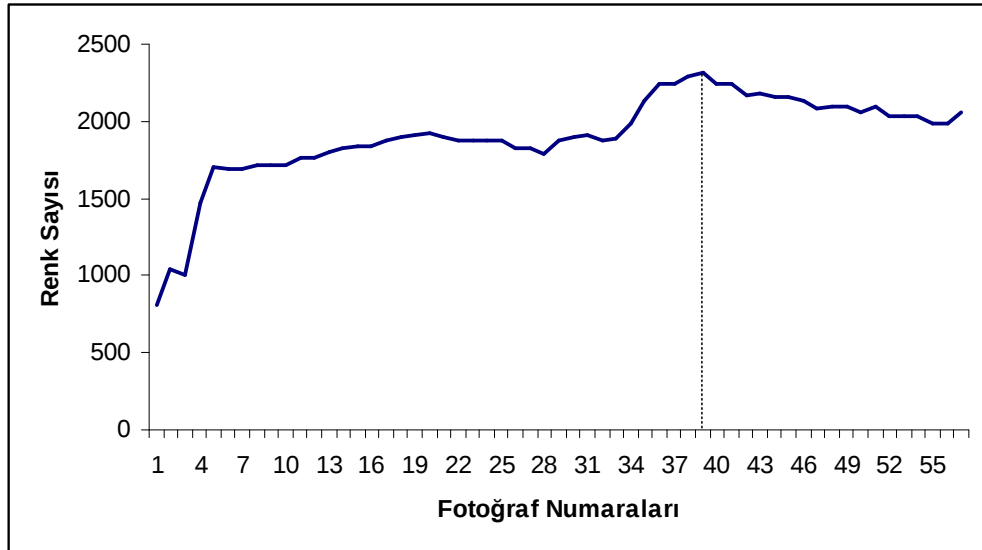
Şekil 4.6. Birinci nesne renk sayım grafiği

Şekil 4.7’de ikinci nesnenin renk sayım grafiği görülmekte ve 29 nolu fotoğrafta en fazla renk sayısına sahiptir.



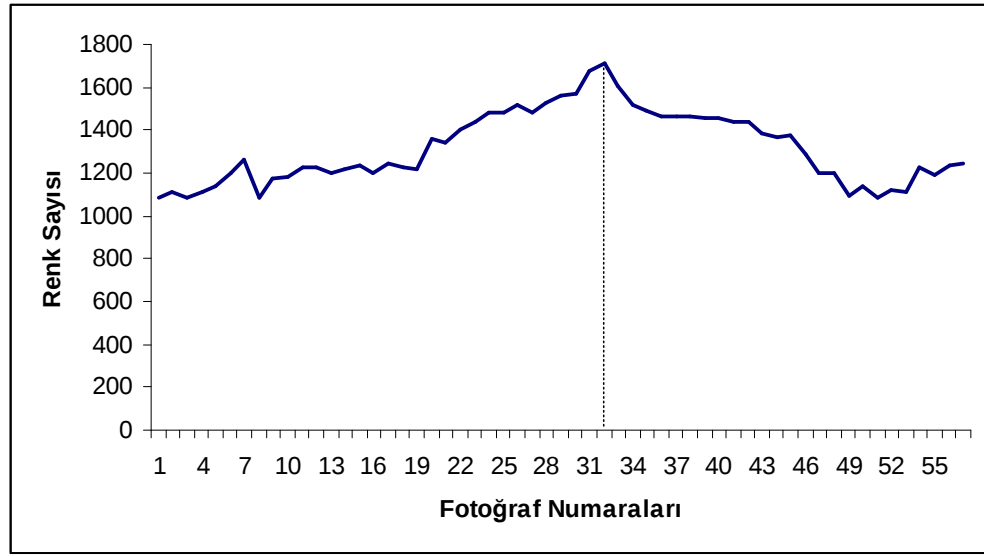
Şekil 4.7. İkinci nesne renk sayım grafiği

Şekil 4.8’de üçüncü nesnenin renk sayım grafiği görülmekte ve 39 nolu fotoğrafta en fazla renk sayısına sahiptir.



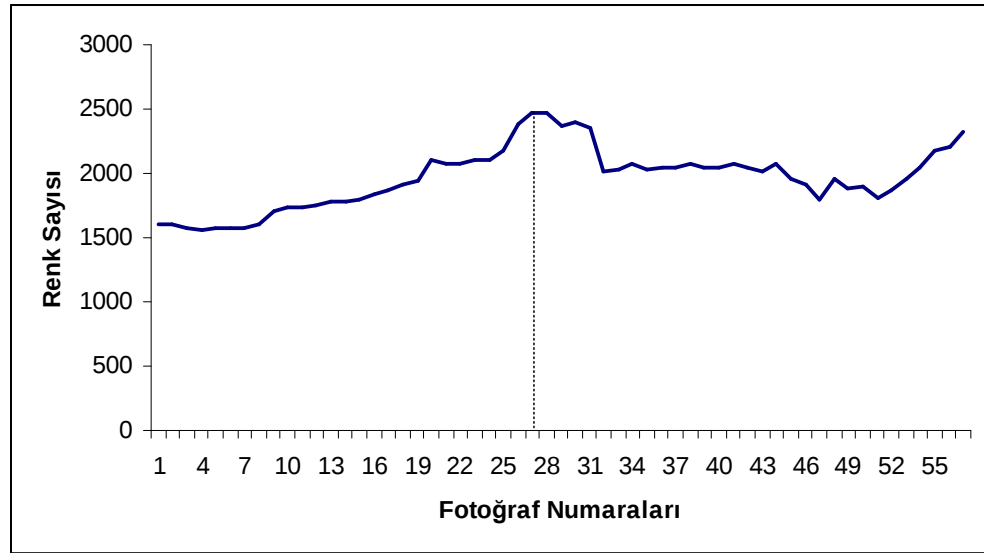
Şekil 4.8. Üçüncü nesne renk sayım grafiği

Şekil 4.9’da dördüncü nesnenin renk sayım grafiği görülmekte ve 32 nolu fotoğrafta en fazla renk sayısına sahiptir.



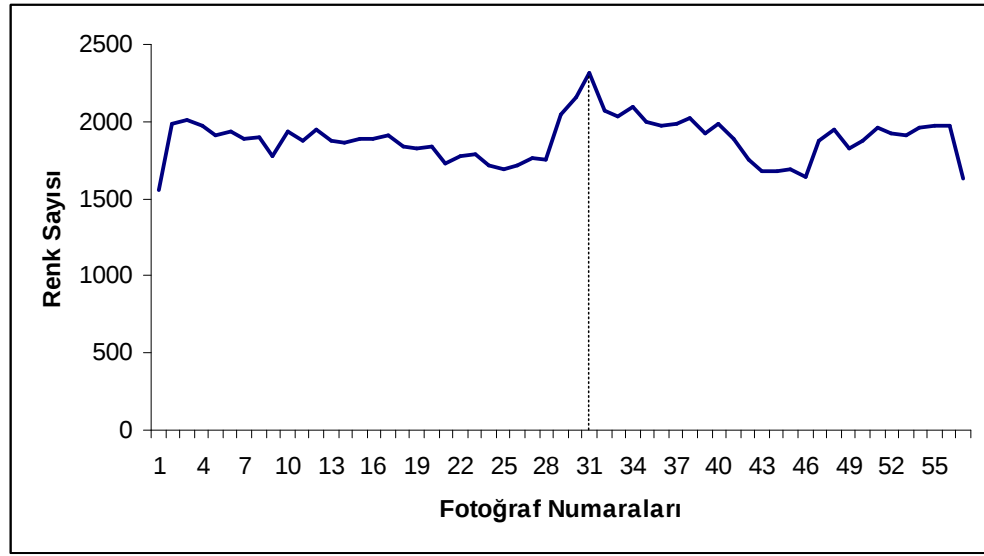
Şekil 4.9. Dördüncü nesne renk sayım grafiği

Şekil 4.10'da beşinci nesnenin renk sayım grafiği görülmekte ve 27 nolu fotoğrafta en fazla renk sayısına sahiptir.



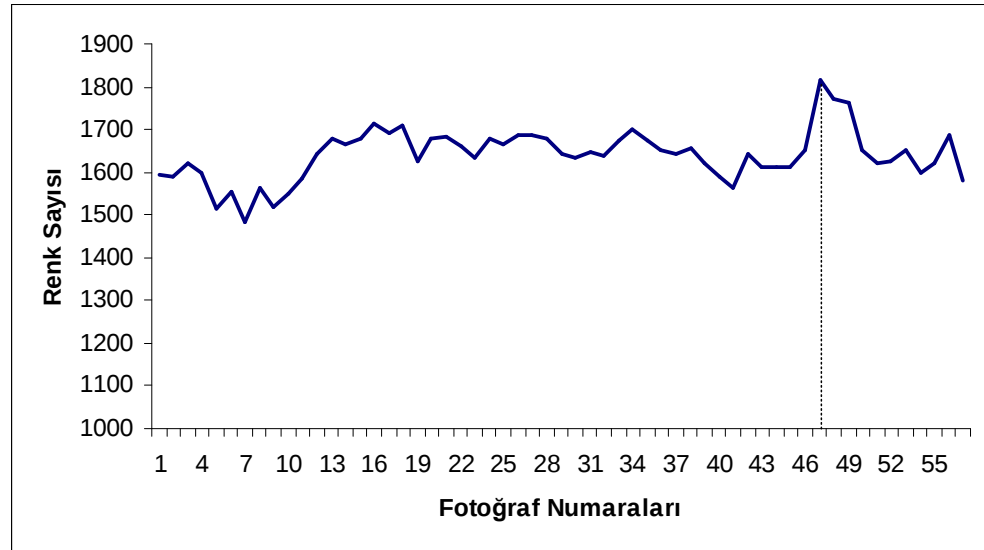
Şekil 4.10. Beşinci nesne renk sayım grafiği

Şekil 4.11'de altıncı nesnenin renk sayım grafiği görülmekte ve 31 nolu fotoğrafta en fazla renk sayısına sahiptir.



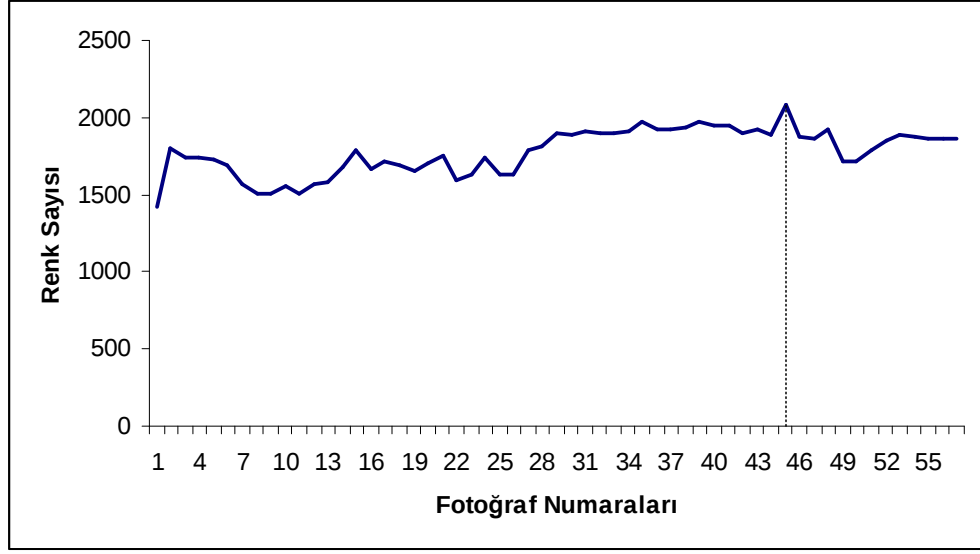
Şekil 4.11. Altıncı nesne renk sayım grafiği

Şekil 4.12’de yedinci nesnenin renk sayım grafiği görülmekte ve 47 nolu fotoğrafta en fazla renk sayısına sahiptir.



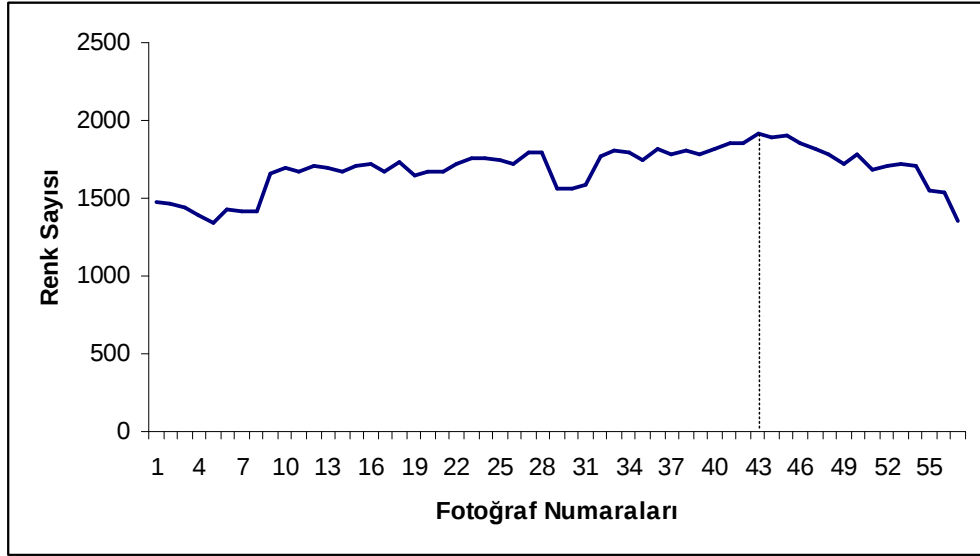
Şekil 4.12. Yedinci nesne renk sayım grafiği

Şekil 4.13’de sekizinci nesnenin renk sayım grafiği görülmekte ve 45 nolu fotoğrafta en fazla renk sayısına sahiptir.



Şekil 4.13. Sekizinci nesne renk sayım grafiği

Şekil 4.14'te dokuzuncu nesnenin renk sayım grafiği görülmekte ve 43 nolu fotoğrafta en fazla renk sayısına sahiptir.



Şekil 4.14. Dokuzuncu nesne renk sayım grafiği

Dokuz nesnenin, renk sayım grafiklerindeki en fazla renk sayısı bulunan fotoğraflar ile Çizelge 4.1'de verilen fotoğrafların odak uzaklıkları karşılaştırılarak

derinlikleri tespit edilmiştir. Örneğin 1 nolu nesnenin renk sayım grafiğinde 14. fotoğrafta en fazla renk sayısına sahiptir ve Çizelge 4.1'e göre 10-20 cm aralığında bulunuyor. Bu aralıkta 28 fotoğraf bulunduğuna göre bu aralıktaki bir fotoğrafın mesafesi bağıntı (4.1) göre 0,357 cm bulunur.

$$\text{fotoğraf mesafesi} \cong \frac{\text{odak aralığı}}{\text{odaktaki fotoğraf sayısı}} \quad (4.1)$$

Örnek vermek gerekirse Şekil 4.6'da görülen grafikte birinci nesne 14. fotoğrafta en fazla renk sayısına sahiptir. 14. fotoğraf, Çizelge 4.1'e göre 10-20 cm kademe aralığında bulunmaktadır ve Çizelge 4.2'ye göre de bu aralıktaki fotoğraflar bağıntı (4.1) ile 0,36 cm olarak hesaplanmıştır. Bu durumda birinci nesnenin $14 \times 0,36 = 5,02$ cm mesafede olduğu bulunur. Çizelge 4.2'de fotoğrafların bulunduğu aralığa göre mesafeleri verilmiştir.

Dokuz nesne için yapılan renk sayım grafikleri incelendiğinde, bazı nesnelerin farklı odak uzaklarında çekilen fotoğraflardaki renk sayılarının birbirlerine çok yakın olduğu görülmüştür. Örneğin Şekil 4.13'teki sekizinci nesnenin renk sayım grafiği incelendiğinde 45 nolu resim ile 2 nolu resimdeki renk sayılarının çok yakın olduğu görülmektedir. Grafiklerdeki bu yakınlığı, nesne büyüklüğü ile üzerine düşen ışık miktarı ve bazı nesnelerin fotoğraflar çekilirken kendi eksenlerinde hafifte olsa hareketlerinin etkilediği tespit edilmiştir.

Çizelge 4.1. Dokuz nesne için farklı odak mesafelerinde çekilen fotoğraflar ve mesafeleri

Fotoğraf No	Mesafe (cm)
1	10
28	20
46	50
53	100
56	200
57	500

Çizelge 4.2. Fotoğrafların bulunduğu aralıklara göre mesafeleri

Aralık(cm)	Fotoğraf (cm)
10-20	0,36
20-50	1,67
50-100	7,14
100-200	33,33
200-300	300,00

Çizelge 4.3'te dokuz nesnenin renk sayım yöntemi ile bulunan mesafeleri ve gerçek mesafeleri görülmektedir. Çizelge incelendiğinde 7. nesne dışındaki diğer nesnelerin gerçek mesafe ile aynı veya yakın olduğu görülmektedir. 7. nesnenin 50 cm mesafe dışında olması ve fotoğraf makinesinin 50 cm mesafeden sonra kademe aralığının azalması hata oranını arttırdığı tespit edilmiştir.

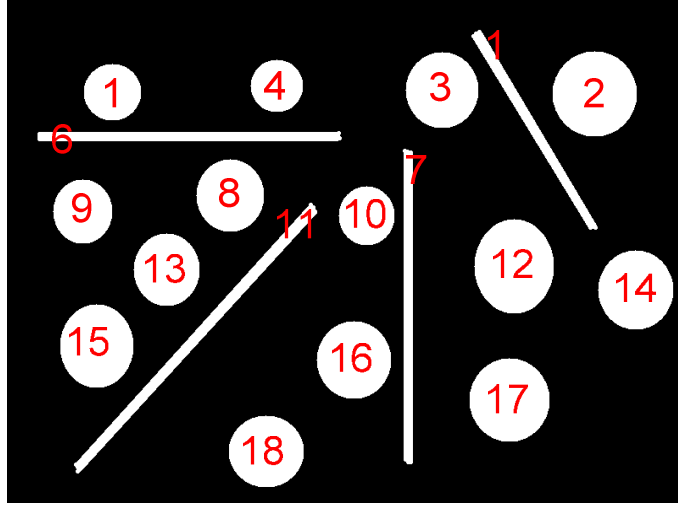
Çizelge 4.3. Nesnelerin bulunan mesafeleri ve gerçek mesafe karşılaştırması

	Hesaplanan Mesafe(cm)	Gerçek Mesafe(cm)	Fark	Oran (%)
1.Nesne	15,02	14	+1,02	93,2
2.Nesne	21,67	21	+0,67	96,9
3.Nesne	38,33	38	+0,33	99,1
4.Nesne	26,67	26	+0,67	97,5
5.Nesne	19,64	21	-1,36	93,5
6.Nesne	25,00	25	0	100,0
7.Nesne	57,14	52	+5,14	91,0
8.Nesne	46,67	48	-2,67	97,2
9.Nesne	43,33	44	-0,77	98,5

4.3. Dairesel Kesitli Nesneler Arasında Engel Tanıma

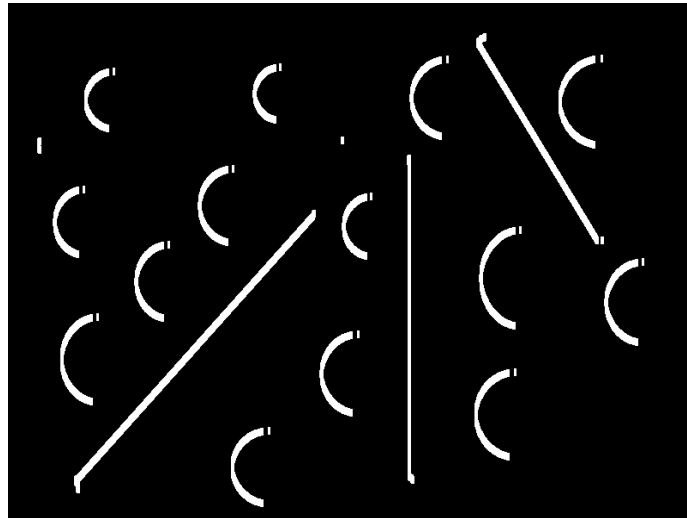
Thresholding ile binary resme dönüştürülen fotoğrafa Sobel x, Sobel y ve eğim işlemleri uygulanarak dairesel kesitli nesneler arasındaki engellerin başlangıç ve bitiş koordinatları tespit edilmiştir.

Deney düzeneğinde, değişik şekillerde engeller oluşturulamamakta, sadece bir engel konumlandırılmaktadır. Bu nedenlerden dolayı engellerin tespiti için yapılan görüntü işleme programını test etmek amacı ile Şekil 4.15'teki resim, paint resim işleme programıyla hazırlanmıştır. 4'dü engel olan 18 nesneli binary resim Şekil 4.15'teki gibi etiketlenmiştir.

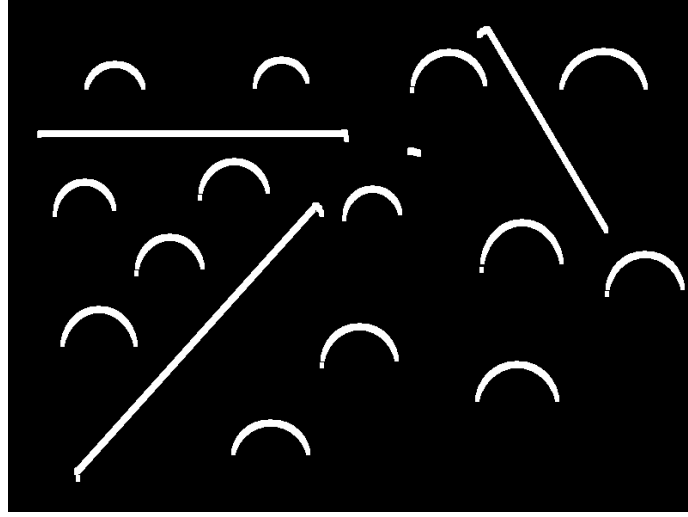


Şekil 4.15. Engeller bulunan binary resim

Şekil 4.16'da Sobel x, Şekil 4.17'de Sobel y filtrelemesi sonucu oluşan resimlere 3x3 kare yapısal eleman ile dilation (genişleme) uygulanmış hali görülmektedir. Bu resimler incelendiğinde Sobel x filtresi uygulanan resimde 6 nolu nesnenin, Sobel y filtresi uygulanan resimde ise 7 nolu nesnenin bulunmadığı görülmektedir.



Şekil 4.16. Sobel x işlemi sonrası



Şekil 4.17. Sobel y sonrası oluşan resim

Çizelge 4.4'te Sobel y, Çizelge 4.5'te Sobel y filtreleri uygulanan resimdeki nesnelerin 10 piksel aralıktaki eğim değerleri görülmektedir. 10 pikselden küçük aralıklar ile bulunan eğimlerin değişme miktarları azalmakta ve sonucu etkilemektedir. Çalışmada kullanılan renk sayımı yönteminde çapı 50 pikselden küçük dairesel kesitli nesnelerde yapılan derinlik analizlerinde hata oranının yüksek olduğu görüldüğünden bu nesneler dikkate alınmamıştır. Bu nedenle en küçük nesne boyu 50 piksel olarak kabul edilmiş ve 10 piksellik aralıklarla en az 5 eğim değerleri hesaplanmıştır.

Örnekleme için Çizelge 4.4'te 11 tane nesnenin eğim değerleri verilmiştir. Çizelge 4.4 incelendiğinde işaret olarak değişim olmayan nesnelerin 1., 7. ve 11. nesneler olduğu görülmektedir. 6. nesne Sobel x filtrelemesi sonucu dikey düzlemde kenarının bulunmaması nedeni ile eğim değeri yoktur. 7. nesnenin tüm noktalarda (dikey düzleme göre) eğim değeri 0'dır.

Çizelge 4.5. incelendiğinde işaret olarak değişim olmayan nesnelerin 1., 6. ve 11. nesneler olduğu görülmektedir. 7. nesne Sobel y filtrelemesi sonucu yatay düzlemde kenarının bulunmaması nedeni ile eğim değeri yoktur. 6. nesnenin tüm noktalarda (yatay düzleme göre) eğim değeri 0'dır.

Bu iki çizelgeler karşılaştırılarak yatay veya dikey kenarları olmayan nesnelerin bulunması sağlanmıştır.

Çizelge 4.6'da tespit edilen engellerin iki boyutta başlangıç ve bitiş koordinatları görülmektedir.

Çizelge 4.4. Sobel x filtreleme sonucu oluşan resimdeki nesnelerin eğim değerleri

	Nesneler										
	1	2	3	4	5	6	7	8	9	10	11
10 Piksellik Artışlar	52,13	-17,82	-19,80	-23,20	-21,37		0,00	-20,56	-22,25	-23,20	-52,13
	56,31	-45,00	-45,00	-56,31	-56,31		0,00	-48,37	-52,13	-52,13	-48,37
	60,95	-56,31	-60,95	-71,57	-66,04		0,00	-60,95	-66,04	-66,04	-48,37
	56,31	-66,04	-71,57	0,00	0,00		0,00	-77,47	-83,66	0,00	-48,37
	60,95	-77,47	-83,66	71,57	77,47		0,00	0,00	0,00	83,66	-48,37
	60,95	0,00	83,66	52,13	60,95		0,00	83,66	71,57	66,04	-48,37
	56,31	83,66	71,57		41,99		0,00	71,57	60,95	48,37	-48,37
	60,95	66,04	60,95				0,00	56,31	39,29		-48,37
	56,31	60,95	45,00				0,00	39,29			-45,00
	60,95	45,00					0,00				-48,37
	56,31	27,90					0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
	56,31						0,00				-48,37
	60,95						0,00				-48,37
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31						0,00				-48,37	
60,95						0,00				-48,37	
56,31											

Çizelge 4.5. Sobel y filtreleme sonucu oluşan resimdeki nesnelerin eğim değerleri

[illegible]

Çizelge 4.6. Engeller ve koordinatları

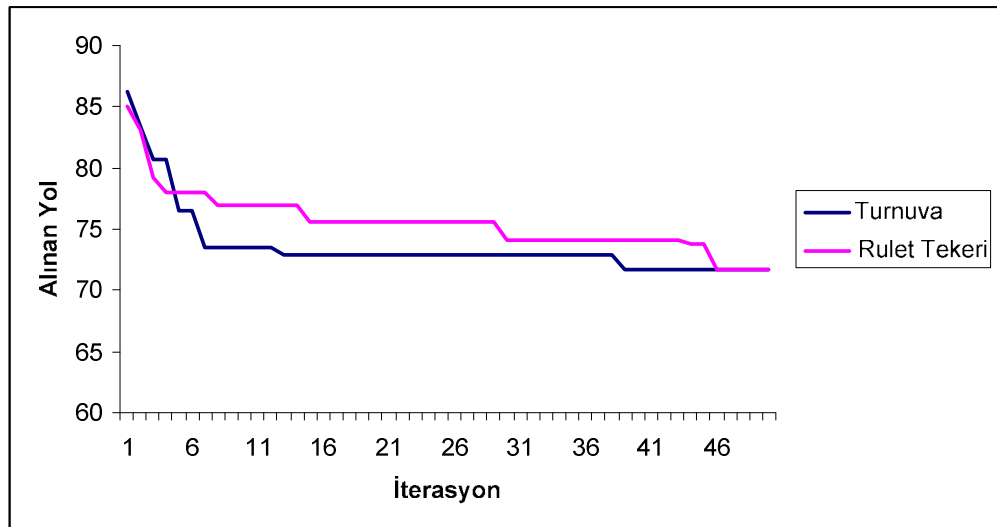
		Engeller			
		1.Nesne	6.Nesne	7.Nesne	11.Nesne
Başlangıç	x	556	36	468	357
	y	39	161	180	244
Bitiş	x	686	387	468	82
	y	273	161	549	559

Engeller çevre-alan bağıntısında tespit edilmiş ve Çizelge 4.6 ile aynı sonuçlar bulunmuştur.

4.4. Genetik Algoritma Test sonuçları

Çalışmamızda dairesel kesitli nesne koordinatları, optimizasyon tekniklerinden GA'ya aktarılarak nesnelere en kısa erişim rotası bulunmuştur.

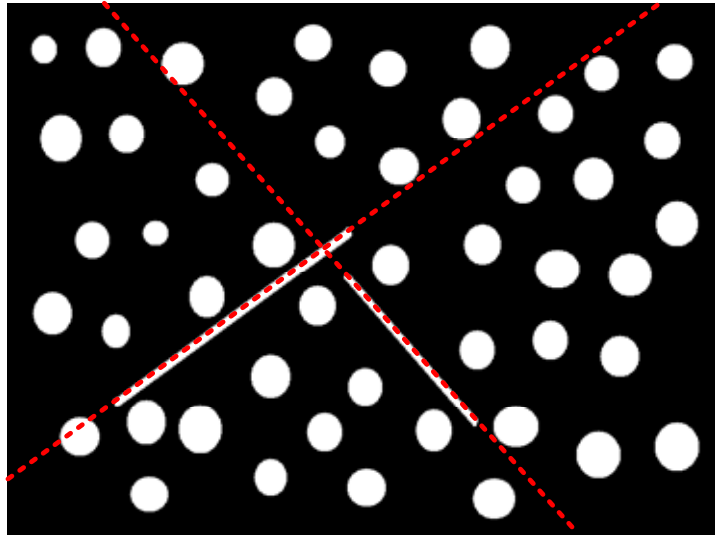
Şekil 4.18'de 10 nesne için 50 döngüde Turnuva metodu ile Rulet tekerinin karşılaştırılmış grafiği görülmektedir. 50 döngü sonunda iki metodunda aynı sonucu verdiği görülmektedir.



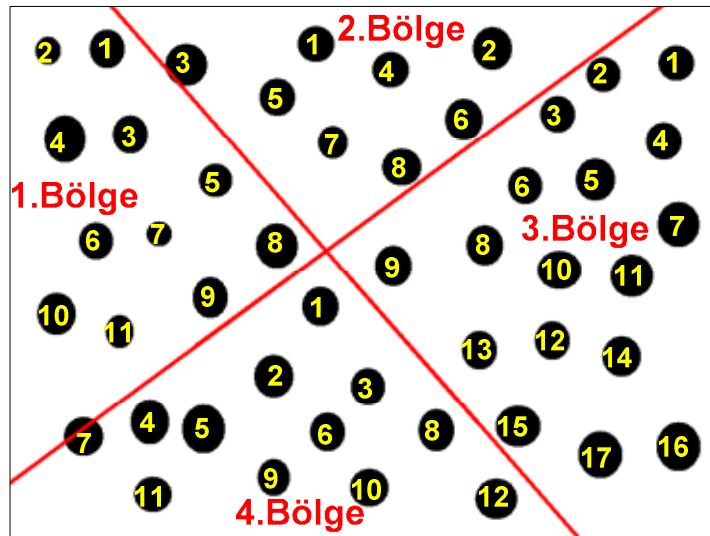
Şekil 4.18. Turnuva metodu ile Rulet tekerinin karşılaştırmalı grafiği

Görüntü işleme yöntemleri ile başlangıç ve bitiş koordinatları bulunan engeller Şekil 4.19 görüldüğü gibi dairesel kesitli nesneleri 4 bölgeye ayırmıştır. Bu bölgelerdeki dairesel kesitli nesnelerin üç boyutlu koordinatları GA'ya aktarılarak işlenmiştir. Ayrılan bölgeler kendi içinde satır sırasına göre numaralandırılmış ve bölge numaraları saat yönünde verilmiştir (Şekil 4.20).

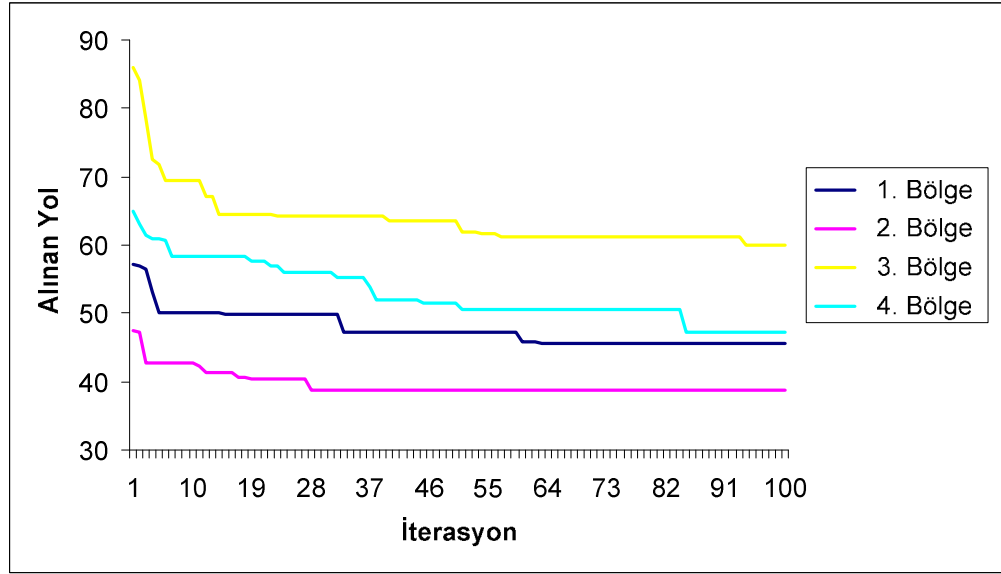
Dört bölge ayrı ayrı GA'ya aktarılarak performansları incelenmiştir. Şekil 4.21'de dört bölgenin birbirlerinden bağımsız işlenmiş performans grafikleri görülmektedir.



Şekil 4.19. Engellere göre parçalara ayrılması

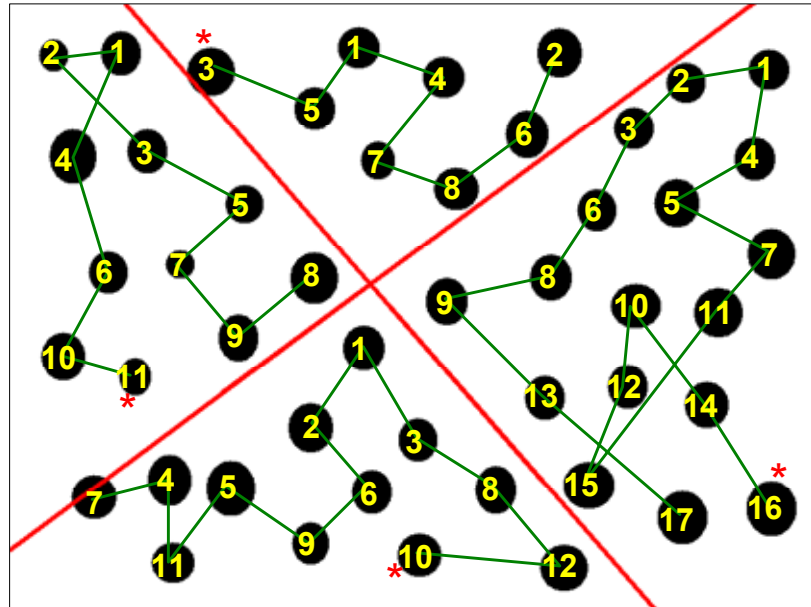


Şekil 4.20. Nesne sırası ve bölgeler



Şekil 4.21. Dört bölgenin GA performans grafiği

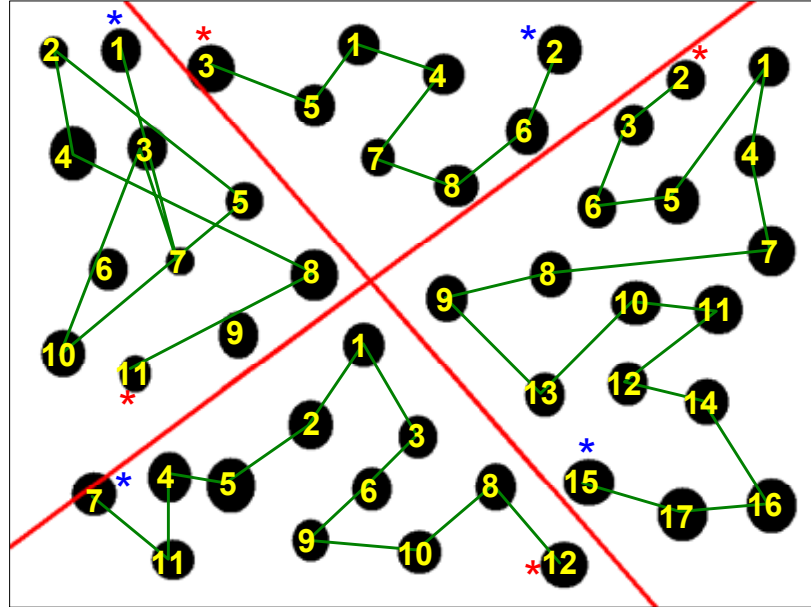
Şekil 4.22’de bölgelerdeki nesnelerin erişim rotası görülmektedir. * ile belirtilen nesneler her bölgenin başlangıç noktasını belirtmektedir. Başlangıç noktaları GA tarafından belirlendiğinden bölgeler arasında geçiş noktaları birbirlerine uzak olarak görülmektedir.



Şekil 4.22. Bölgelerdeki nesnelerin erişim rotası

Şekil 4.23'te başlangıç ve bitiş noktaları belirlenen nesnelerin GA sonucu görülmektedir. Kırmızı * ile işaretli noktalar gölgelerin başlangıç noktalarını, mavi * ile işaretli noktalarda gölgelerin bitiş noktalarını göstermektedir.

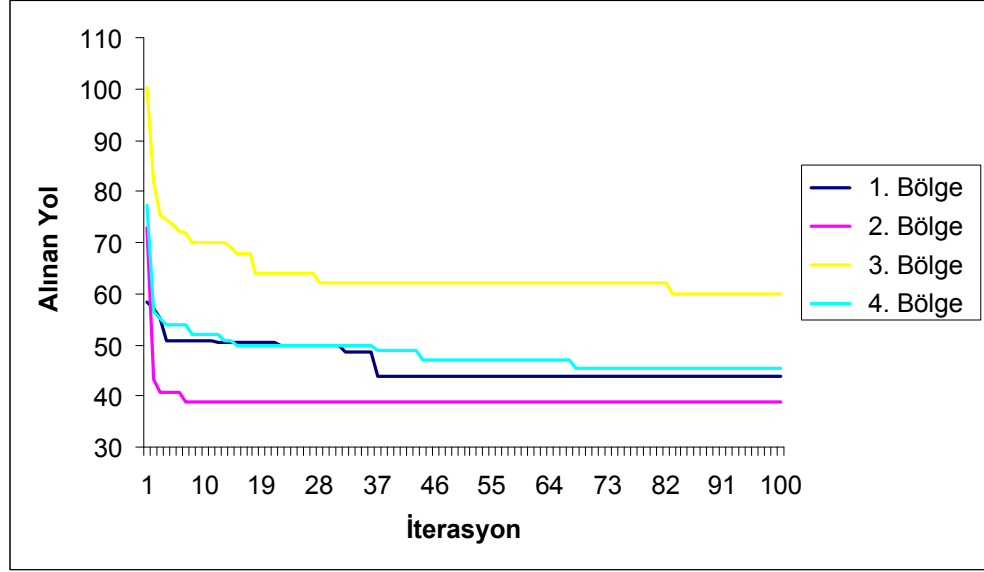
Şekil 4.22 ile Şekil 4.23'deki erişim rotaları bölge olarak karşılaştırıldığında Çizelge 4.7'de elde edilen değerler incelendiğinde kontrollü GA'nın (Şekil 4.23'teki rotanın) %3 oranında daha kısa olduğu hesaplanmıştır. Bölgelerin birbirlerine geçiş noktalarındaki mesafelerde hesaba katıldığında, Çizelge 4.7'deki geçiş noktalarındaki toplam alınan yolda kontrollü GA ile (Şekil 4.23'teki rota) %16 oranında kısa olduğu bulunmuştur. Bölgelerdeki nesne sayıları arttıkça bu oranların arttığı incelenmiştir.



Şekil 4.23. Başlangıç ve bitiş noktası belirlenen nesnelerin erişim rotası

Çizelge 4.7. Kırksekiz nesne için GA'nın kontrolü ve kontrolsüz oranları

	Bölgelerdeki Toplam	Geçiş Noktaları ile Toplam
Kontrollü GA	186,67 birim	205,216 birim
Kontrolsüz GA	192,40 birim	244,287 birim
Oran	%3	%16



Şekil 4.24. Başlangıç ve bitiş noktaları belirlenen nesnelerin GA performans grafiği

Şekil 4.24 ile Şekil 4.21 grafikleri incelendiğinde;

11 nesne bulunan birinci bölgede toplam hesaplanan mesafe Şekil 4.24'te daha kısadır ve sonuca daha erken ulaşmıştır.

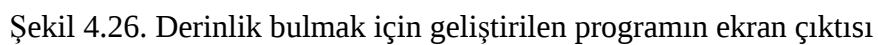
8 nesne bulunan ikinci bölgede toplam hesaplanan mesafe aynı olmasına rağmen Şekil 4.24'te sonuca daha erken ulaşmıştır.

17 nesne bulunan üçüncü bölgede toplam hesaplanan mesafe Şekil 4.24'te biraz daha kısadır ve sonuca daha erken ulaşmıştır.

12 nesne bulunan dördüncü bölgede toplam hesaplanan mesafe Şekil 4.24'te daha kısadır ve sonuca daha erken ulaşmıştır.

Şekil 4.25'te GA için geliştirilen programın ekran çıktısı görülmektedir.

Şekil 4.26'te nesnelerin koordinatlarını bulmak için geliştirilen programın ekran çıktısı görülmektedir.



5. SONUÇ VE ÖNERİLER

Bu çalışmada tek kameradan alınan dijital resim ile nesnenin konumları ve bu konumlara erişim için gerekli sıra bulunmuştur. Robot kol hareket optimizasyonunu gerçekleştirmek için önce dijital fotoğraflar görüntü işleme teknikleri ile işlenmiş ve dairesel kesitli nesnelerin ağırlık merkezlerinin x, y koordinatları ile engellerin başlangıç ve bitiş x, y koordinatları bulunmuştur.

Farklı odak uzaklıklarında çekilmiş resimlerdeki dairesel kesitli nesnelerin, her biri için ayrı ayrı renk sayımı yapılmıştır. En fazla renk sayısına sahip olan resmin odak mesafe değerine göre dairesel kesitli nesnelerin mesafeleri bulunmuştur. Bulunan bu mesafenin gerçek mesafeye oranı 20 cm’de %95, 30 cm’de %92, 50 cm’de %71 olduğu görülmüştür.

Nesnelerin koordinat verilerinin GA aktarılması ile, nesnelere en uygun erişim sırasının belirlenmesinde, %32-%48 oranlarında nesneler arasında alınan yolun kısaldığı görülmüştür.

Engellerin bulunduğu fotoğraflarda engellerin koordinatına göre bölgelere ayrılmıştır. Bu bölgelerdeki nesneler arasından başlangıç ve bitiş nesneleri belirtilerek alınan toplam yolun %11-%23 oranlarında kısaldığı görülmüştür.

Derinlik bulmada 50 cm’den uzaktaki nesnelerde hata oranının artmasına, kullanılan dijital fotoğraf makinesinin manuel odak mesafe ayarının 50 cm’den sonra mesafe kademelerinin azalması neden olmaktadır. Bu nedenle 50 cm kadar olan mesafede çalışılmıştır.

Çalışmalar güneş ışığında yapılmıştır. Işık ve çözünürlük sonucu etkilediğinden farklı dalga boylarındaki ışık kaynakları ve farklı özellikte kameralar ile çalışmalar yapılacaktır. Çalışmadaki resimler dijital fotoğraf makinesinden *.jpg formatında alınmış ve sonra *.bmp formatına çevrilmiştir. *.jpg formatında sıkıştırma kayıpları olduğundan fotoğraf makinesinden *.raw formatında resimlerin alınması üzerine araştırmalar yapılacaktır. *.raw formatından *.bmp formatına dönüştürmede kayıpların *.jpg’ye göre az olacağından ve gelecekte yapılması planlanan çalışmalara ön hazırlık olarak *.bmp formatında çalışılmıştır.

KAYNAKLAR

- ALVAREZ, J.C., GONZALEZ, R.C., ALVAREZ, D., SHKEL, A., LUMELSKY, V., 2004. Accounting for limited sensing in real-time obstacle avoidance for mobile robots, **Proceedings of the 2004 IEEE International Conference on Robotics and Automation**, New Orleans.
- AN D., WANG H., 2004. A New Laser Radar Based Obstacle Avoidance Method for Intelligent Mobile Robots, **Proceeding of the 5th World Congress on Intelligent Control and Automation**, P.R.China.
- ASLANTAŞ V., TUNÇ, M. 2004. Resim Netliğinden Yapay Sinir Ağları İle Mesafe Tespiti. **Akıllı sistemlerde yenilikler ve uygulamaları Sempozyumu**.
- BALA R., SHARMA G., 2005, System Optimization in Digital Color Imaging, **IEEE Signal Processing Magazine**, January 2005 p:55-63.
- BEASLEY, D., BULL, D.R and MARTIN, R. R., 1993. An Overview of Genetic Algorithms: Part 1. Fundamentals. **University Computing**, 58-69.
- BESSIERE, P., AHUACTZIN, J. M., TALBI, E. G. AND. MAZER E., 1993. The “Ariadnes Clew” Algorithm: Global Planning with Local Methods. **<http://citeseer.nj.nec.com/bessiere93ariadnes.html>**.
- BORENSTEIN J., KOREN Y., 1989. Real-Time Obstacle Avoidance for Fast Mobile Robots, **IEEE Transactions on System, Man and Cybernetics** Vol 19, No 5, P 1179, September-October.
- BROOKS R. A. 1992, Artificial life and real robots. **Proceedings of the First European Conference on Artificial Life**, pages 3-10, Cambridge, MA, The MIT Press.
- DABAN F., 2004. **Genetik Algoritma Kullanarak Çok Kriterli Ders Programları Hazırlanması**, Yüksek Lisans Tezi (Basılmamış)., Mustafa Kemal Üniversitesi, Hatay.
- DEODHAR, J., TRAGOUDAS, S., 2001, Color Counting And Its Application To Path Delay Fault Coverage, **IEEE Communications Magazine**. p:373-378.
- DONG Z. 1999, “Genetic Algorithm Applications.” **http://www.me.uvic.ca/~zdong/courses/mech620/GA_App.PDF**.
- EMEL G. G. ve TAŞKIN Ç., 2002. Genetik Algoritmalar Ve Uygulama Alanları. **Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi** Cilt XXI, Sayı 1.
- FLOREANO D. AND MONDADA F., 1998. Evolutionary Neurocontrollers for Autonomous Mobile Robots. **Neural Networks**, Vol. 11, No. 7, pp. 1461-1478, **<http://lis.epfl.ch/publications/nm2.pdf>**.
- FORREST, S. , JAVORNİK, B., SMİTH, R. E. AND PEERELSON A.S., 1992. Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. **<http://citeseer.nj.nec.com/15549.html>**.
- GOLDBERG, D. E., 1989. **Genetic Algorithms in Search, Optimization, and Machine Learning**, Addison-Wesley Publishing Comp., ISBN 020-115-767-5.
- GONZALEZ, R. C., CANCELAS, J., ALVAREZ, J. C., FERNÁNDEZ A. AND ENGUİTA J. M., 1999. Fast Stereo Vision Algorithm For Robotic Applications, **71. IEEE International Conference on Emerging Technologies and Factory Automation**, Barcelona, Spain.

- GONZALEZ R. C. AND WOODS R. E. 1992. **Dijital Image Processing**. Addison-Wesley Publishing Company Inc. UK.
- JAFARI S. AND JARVIS, R., 2004, Robotic hand-eye coordination: from observation to manipulation **Hybrid Intelligent Systems, 2004. Proceedings. HIS 2004. Fourth International Conference.**
- JAIN, R.C., KASTURI, R., SCHUNCK B.G., 1995. **Machine Vision**, ISBN 007-0320-187, Mc Graw-Hill.
- HARVEY I., HUSBANDS P., AND CLIFF D., Issues in evolutionary robotics. **Proceedings of the Second International Conference on Simulation of Adaptive Behavior**, Honolulu, Hawaii, pages 364--373. The MIT Pres.
- HOFFMAN, A.J. AND P. WOLFE, 1985, "History," in: Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds, **The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization** ,Wiley, New York, pp. 1-36.
- HUANG, J., CHENGJUN, L. AND HARRY, W. 1998. Eye detection and Face Recognition using Evoluationalary Computation. **Proceeding of NATO-ASI on Face Recognition: From Theory to Applications.**
- KAHRAMAN, F., 2001. **Sayısal Görüntü İşleme İçin Bir Yazılım Paketi Geliştirilmesi ve Sayısal Resimlerde İnsan Yüzünün Yer Tespitinde Kullanımı**. Yüksek Lisans Tezi (Basılmamış). Mustafa Kemal Üniversitesi,108 s,Antakya.
- KANASUGI A. AND J. TAKAHASHI. 2001. A Greedy Router Based on Mase Router and Genetic Algorithm. **2001 International Conference on Electronics Packaging (ICEP 2001).**
- KERT M., ÖZDEMİR E., 2005. Üç Boyutlu Sistemde Tek Kamera İle Dairesel Kesitli Nesnelerin Konumlarının Belirlenmesi, **1. Uluslararası Mesleki ve Teknik Eğitim Tek.Kongresi (METET 2005)**, İstanbul.
- KOZA, J.R. 1990, Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems, **Stanford University Computer Science Department technical report STAN-CS-90-1314.**
- KOZA, J. R. 1992. "A Genetic Approach to Econometric Modelling", in Bourguine, P. and B. Walliser (eds.), **Economics and Cognitive Science**, Pergamon Press, s. 57-75.
- KOZEK T., ROSKA T., CHUA O. L. 1993. Genetic Algorithm for CNN Template Learning, **IEEE Trans on Circ. And Sys.** Vol.40, no.6 392-402.
- LALENA, M., 2005. Travelling Salesman Problem Using Genetic Algorithms. <http://www.lalena.com/ai/tsp/>.
- LIT P., LATİNNE P. , REKİEK B. VE DELCHAMBRE A. 2001, Assembly Planning With An Ordering Genetic Algorithm, **International Journal of Production Research**, Volume: 39, Number: 16, s. 3623-3640.
- MAN K. F., TANG K. S., KWONG S. 1996. Genetic Algorithms: Concepts and Applications. **IEEE Trans. Ind. Electronics**, vol. 43, no 5, 519-534.
- MATTHİES L., GAT E., HARRİSON R., WİLCOX B., VOLPE R., and LİTWİN T. 1995, Mars Microrover Navigation:Performance Evaluation and Enhancement., [www-robotics.jpl.nasa.gov/ publications/Richard_Volpe/iros95b.pdf](http://www-robotics.jpl.nasa.gov/publications/Richard_Volpe/iros95b.pdf).
- MIRZABAKİ M. 2004. Depth Detection Through Interpolation Functions. **International Society for Photogrammetry and Remote Sensing Conferans (ISPRS)**, İstanbul.

- MORAVEC, H., 2000. Darpa Mars Program Research Progress, **Carnegie Mellon University, Technical Report**. <http://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/2000/>.
- NOLFI S. 1997. Using Emergent Modularity to Develop Control Systems for Mobile Robots. <http://gral.ip.rm.cnr.it/nolfi/papers/nolfi.emodular.pdf>.
- OKUYAMA F., TAKAYAMA J.Y., OHYAMA S. AND KOBAYASHI A., 2002, A study on determination of a sound wave propagation direction for tracing a sound source, **Proceedings of the 41st SICE Annual Conference**.
- ORAL, M. 1996. Classification and analysis of Nailfold Cappillaries by Using Image Processing and Neural Nets. **Computer Studies Research Report**, CS-96-1, University of Glamorgan.
- ÖZDEMİR E. VE ORAL M., 2003, Genetik Algoritma ve Görüntü İşleme Teknikleri Kullanarak Servis Araçlarının Şehir Trafiğinde İzlediği Yolum Kısaltılması, **1. Uluslararası Yerel Yönetimler Üniversite ve Sanayi İşbirliği, Gazi Üniversitesi**, Ankara.
- ÖZÇAKAR N. 1998, Genetik Algoritmalar, **İstanbul Üniversitesi İşletme Fakültesi Dergisi**, Cilt: 27, Sayı: 1, s. 69-82.
- PENTLAND, A. P. 1982. A new Sense for Depth of Field. **IEEE Communications Magazine**. p:523-531.
- PITAS I. 1995. **Digital Image Processing Algorithms**. Prentice Hall, Europe.
- RAMESH V. E. AND MURTY M. N., 1999, Off-Line Signature Verification Using Geneticall Optimized Weighted Features, **The Jurnal Of The Pattern Recognition Society**. v 32, s 217-233.
- REN Q., JIN Z., ZHONGXING Y., 2000, Analysis of common genetic operators. **5th International Conference-Signal Processing Proceedings WCCC-ICSP**.
- RUICHEK, Y., ISSA, H., POSTAIRE, J.G., BURIE, J.C., 2004, Towards Real-Time Obstacle Detection using a Hierarchical Decomposition Methodology for Stereo Matching with a Genetic Algorithm, **Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)**.
- SCHLOTTMANN F. 2001, A Hybrid Genetic Quantitative Method For Risk Return Optimisation Of Credit Portfolios, <http://www.aifb.uni-karlsruhe.de/CoM/pub/hgqm.pdf>.
- SHARMA G., TRUSSELL H. J., 1997, Digital Color Imaging, **IEEE Transactions On Image Processing**, Vol. 6, No. 7, July.
- SHIEH, C. S., HUANG, H. C., WANG, F. H. and PAN, J. S., 2004. **Genetic Watermarking Based on Transform Domain Techniques**, The Journal of the Pattern Recognition, Vol: 37, P: 555-565.
- SUBBARAO M., GURUMORTHY N. 1988. Depth Recovery from Blurred Edges. In **Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition**.
- TAKATSUKA M., WEST G. A. W., VENKATESH S., CAELLI T.M., 1999, **Low-Cost Interactive Active Monocular Range Finder**, 1999. IEEE Computer Society Conference on, 23-25 June.
- TAN K.C., LEE L.H., ZHU Q.L. VE OU K. 2001, Heuristic Methods For Vehicle Routing Problems With Time Windows, **Artificial Intelligence in Engineering. Elsevier** (05/05/2002). s. 281-295.

- UYANIK B. VE ÖZDEMİR E. 2003. Görüntü İşlemeye Dayalı Robot Kol Hareket Optimizasyonu, **International Conference on Signal Processing ICSP**, Çanakkale.
- UYANIK B. 2003. **Görüntü İşlemeye Dayalı Robot Kol Hareket Optimizasyonu**. Yüksek Lisans Tezi (Basılmamış). Mustafa Kemal Üniversitesi. 65 s.,Antakya.
- WADHWA S. VE CHOPRA A. 2000, A Genetic Algorithm Application: Dynamic Reconfiguration in Agile Manufacturing Systems, **With Emphasis on Useful Applications of Advanced Technology**. December 2000. volume 9, number 4, www.ici.ro/ici/revista/sic2000_4/art01.htm.
- WONG, E.T.P., JARVIS R., 2004, Real Time Obstacle Detection and Navigation Planning for a Humanoid Robot in an Indoor Environment, **Proceeding of the 2004 IEEE Conference on Robotics, Automation and Mechatronics**. Singapore.
- YAMAGUTI N., OE SH. AND TERADA K. 1997, A Method of Distance Measurement by Using Monocular Camera, **The 36th SICE Annual Conference**.
- YENİAY Ö., 2001, An Overview of Genetic Algorithms, **Anadolu Üniversitesi Bilim ve Teknoloji Dergisi**, Cilt: 2, Sayı: 1, s. 37-49.
- YODA I., 1995, Automatic Generation of Directional Erosion and Dilation Sequence by Genetic Algorithms, **Proc. of Second Asian Conf. on Computer Vision (ACCV'95)**, December 1995, pp.II-103-107.
- ZHENG H., KONG L. X. AND NAHAVANDİ S., 2002, Automatic Inspection of Metallic Surface Defects Using Genetic Algorithms. **Journal of Materials Processing Technology**, 125-126:427-433.

ÖZGEÇMİŞ

1974 İskenderun'da doğdum. İlk, orta ve lise öğrenimimi İskenderun'da tamamladım. 1993 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi, Bilgisayar öğretmenliğini kazandım ve 1997 yılında mezun oldum. 1997 yılında İskenderun Anadolu Teknik lisesine bilgisayar öğretmeni olarak atandım. 2003 yılında Mustafa Kemal Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Anabilim dalında yüksek lisansa başladım. Halen İskenderun Anadolu Teknik Lisesi bilgisayar bölümü, bölüm şefi olarak görev yapmaktayım. Evli ve bir çocuk babasıyım.