



**YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BİLGİSAYAR PROJESİ

ROBOTİK SİSTEMLERDE ENGEL TANIMA VE YÖN BELİRLEME

Proje Yöneticisi : Yrd. Doç. Dr. SİRMA YAVUZ

Proje Grubu
03011504 : Aytunç BEKEN
03011503 : Necmi ÖZTÜRK

İstanbul, 2006-2007

İÇİNDEKİLER

Şekil Listesi	2
Tablo Listesi	2
Önsöz	3
Abstract	4
1- Giriş	
1-1 – Robot Sistemleri	4
1-2 – Görüntünün İşlenerek Nesnelerin Tanınmasının Önemi	5
2- Fizibilite Analizi	
2-1 – Teknik Fizibilite Çalışması	
2-1-1- Sistemde kullanılacak olan Donanım Araçları.....	6
2-1-2- Sistemde kullanılacak olan Yazılım Araçları.....	6
2-1-3- Kullanılan İşletim Sistemi	6
2-2 – Ekonomik Fizibilite Çalışması	
2-2-1- Maliyet Hesaplaması	6
2-3 – Legal Fizibilite Çalışması	6
2-4 – Kullanılabilirlik Fizibilitesi	6
2-5 – Alternatif Fizibilite.....	7
2-4 – Gant Diyagramı.....	7
3- Sistemin Yapısı	
3-1- Resim üzerinde bulunan parazitlerin yok edilmesi.....	7
3-1-1- Median Filter	7
3-2-Resim üzerinde tanınacak olan nesnenin sınırlarının belirlenmesi.....	8
3-2-1- Yöntemin Açıklaması.....	9
3-2-2- Yöntemin Akış Diyagramı.....	10
3-2-3- Yöntemin Sonuçları.....	11
3-3- Resim üzerinde tanınacak olan nesnenin sınırlarının belirlenmesi(Yöntem2)	13
3-4-Şekil Tanıma	14
3-4-1- Şekil Tanıma Yöntemi(1).....	14
3-4-2- Şekillerin Uzunluklarının Belirlenmesi.....	
3-4-3- Şekil Tanıma Yöntemi(2).....	18
3-4-3-1- Şeklin Resimden Ayrılması İşlemi (Segmentation).	18
3-4-3-2- Şeklin Tanınması.....	20
3-5- Genel Anlamda Engel Tanıma	22
3-6- Sümilatör	22
3-7- Yol Bulma Algoritması	24
4- Sonuç	
5- Kaynaklar	
7- Özgeçmişler	

Şekil Listesi :

- 1- Şekil-1- Gant Diyagramı
- 2- Şekil-2- Satranç Tablosu (Orijinal)
- 3- Şekil-3- Satranç Tablosu (Median Filter)
- 4- Şekil-4- Paintte Hazırlanmış Olan Resim (Orijinal)
- 5- Şekil-5- Web Kamera Görüntüsü (Küp Önyüz) Yakın Çekim
- 6- Şekil-5- Web Kamera Görüntüsü (Küp Önyüz) Uzak Çekim
- 7- Şekil-6- Yönler
- 8- Şekil-7-
- 9- Şekil-8- Engel(Küp1)
- 10- Şekil-9- Engel(Küp2)
- 11- Şekil-10 -Engel(Üçgen1)
- 12- Şekil-11 -Engel (Üçgen2)
- 13- Şekil-12 -Engel (Daire1)
- 14- Şekil-13- Engel (Daire2)

Tablo Listesi:

- 1- Tablo-1- Maliyet Analizi Tablosu
- 2- Tablo-2- Örnek Resim Pikselleri

ÖNSÖZ

Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi Bilgisayar Mühendisliği bölümü öğrencileri ve öğretim görevlileri tarafından gerçekleştirilmekte bu projede günümüzde çok büyük bir yankı yaratmakta olan robotlar işlenmiştir. Robotların hayatımızdaki yeri git gide artmakta ve bize sağladıkları faydalar da zamanla çoğalmaktadır. Bu çerçevede üniversitemiz bu gelişmelerde rol oynamaktadır.

Projeyi yürüten ve desteklerini esirgemeyen Sayın Yrd. Doç Dr. Sırma YAVUZ ve Zeynep KURT'a teşekkürlerimizi iletiriz.

ABSTRACT

In this paper we are developing an object recognition system that localises and identifies objects for an autonomous system which is developed in Yıldız Technical University, Computer Science Department. The purpose of the robot is to navigate in an unknown environment starting from an unknown location to build the map of the area while it can compute its own location at the same time.

Our aim is to develop an object recognition system for the robot which can see front and can follow the correct way for it. Infrared sensors are used on the vehicle to determine the distance of surrounding objects and to choose the path to follow. Although infrared sensors, object localisation and identification is a crucial problem for advanced mobile service robots. Because sometimes sensors can't detect an object in front of it because of its geometric shape.

1- GİRİŞ

1-1- ROBOT SİSTEMLERİ

Son günlerde hangi haber bültenini açsanız, hep onlardan bahsediliyor; yeni yapılan robot köpek, marsa gönderilen Pathfinder, futbol oynayan robotlar... Peki nedir robot dediğimiz? Kökleri nereye kadar uzanır? Robotu robot yapan özellikler nelerdir?

İnsanlığın kendine yardımcı olacak mekanizmalar düşünmesi, tahminlerinizden çok daha eskilere uzanmakta. M.Ö. 800'de, Homeros, Ilyada adlı eserinde verilen görevleri yerine getirebilen hareketli uçayaklılardan bahseder. MÖ 350'de Aristo, bir eserinde; "Eğer her araç kendi işini görebilseydi, insan eline ihtiyaç duymadan mekik kendi dokuyabilse, lir kendi çalabilseydi, yöneticilerin elemanlara ihtiyacı kalmazdı." diyerek ilk otomasyon fikrini ortaya atmıştır. 13.yy ise Eb-ül-İz-el-Cezeri adlı bir Arap otomatlar hakkında bir kitap yazdı. Kitapta 300'e yakın otomatik mekanizmanın yanı sıra, çamaşır teknesini doldurup boşaltabilen otomatik bir Arap kadını resmediliyordu.

Peki, robot ne zaman robot özelliği kazanır? Bilgisayarınızın yazıcısı veya mutfak robotunuz gerçekten birer robot mudur? Bir makineye robot diyebilmek için, en önemli şartlardan birisi algılamadır.

Bir robot sınırlı da olsa dış dünyadan bazı algılar yapabilmelidir. Bu algılamalar, kimyasal, konum, renk, ışık, şekil gibi geniş bir yelpazede yer alır. Daha sonra elde ettiği bu verileri, otonom olarak yorumlayabilmeli, algıya ne gibi tepkide bulunacağına karar vermelidir. Son olarak da verdiği bu kararını uygulamaya koyabilmelidir. Kısaca robot üç ana kısımdan oluşur;

1. Çevre hakkında gerçek-zamanlı bilgi elde etmek için kullanılan alıcılar,
2. Karar vermeyi ve kontrolü sağlayan elektronik beyin,
3. Verilen kararların uygulamasını sağlayan eyleyiciler ve hareket sistemleri

Robot teknolojisinde uygulama düzeyinde yakın gelecek içindeki beklentiler, robot teknolojisi üzerine yoğunlaşmış araştırma merkezleri ve üniversitelerdeki araştırma çalışmalarının sonuçlarına bağlı olarak gelişecektir. Endüstriyel robot teknolojisinin beklentileri daha hafif ve daha hızlı robotlar geliştirilmesi yönündedir. Üretim maliyetlerinin azaltılması üretim mühendislerinin robot teknolojisinden bekledikleri en önemli hususdur. Benzer şekilde endüstriyel nitelikli gezer robotların üretim maliyetini azaltıcı ve üretim hızını arttırıcı etkileri beklenmektedir.

1-2- GÖRÜNTÜNÜN İŞLENEREK NESNELERİN TANINMASININ ÖNEMİ

Robotların bir yerden başka bir yere belli bir yolu izleyerek gitmek için ortamda bulunan bazı engelleri gözlemleyerek tanınması ve ona göre yolunu belirlemesi gerekmektedir. Bu aşamada harita çıkarma amaçlı sensörler kullanılmaktadır, ayrıca kameraya yerleştirilen kameralar vasıtasıyla ortamdan görüntü alınarak nesnelerin tanınması yön belirlemede kullanılmaktadır. Karşısına gelen şekillerden nesnenin ne olduğunu anlaması robotik sistemlerde çok kullanılan “nesne tanıma” işlemleri de çok önemlidir.

Geliştirdiğimiz bu projede de oluşturulan robot sisteminde yön bulma amaçlı engel tanımaya yönelik bir çalışma yapmaktayız. Bu aşamada kullandığımız donanımsal ve yazılımsal araçlar sistem yapısına uyum sağlayabilecek bir yapıda seçilmiştir.

2- FİZİBİLİTE ANALİZİ

2-1- Teknik Fizibilite Çalışması

2-1-1- Sistemde kullanılacak olan Donanım Araçları

Projede Kullanılacak olan Robotun maliyeti 4348.6 YTL olarak bölümdeki hocalarımız tarafından hesaplanmıştır.

Projede ek olarak kullanılacak donanım parçası ise normal kalitede görüntü alabilecek bir kameradır.

2-1-2- Sistemde kullanılacak olan Yazılım Araçları

Proje de kullanılacak olan robotun programlaması C programlama dilinde yapılmıştır. Bizim kullanacağımız dil ise C++ olacaktır. Bunun nedeni C++ dilinin C dilini temel alması ve daha ileri seviyede nesneye yönelik bir dil olmasıdır.

2-1-3- Kullanılan İşletim Sistemi

Windows XP işletim sistemi kullanılacaktır. XP işletim sistemi bir çok program ile uyum göstermektedir. Yapılacak olan projenin altyapısında da uyumluluk göstereceğinden bu işletim sistemi seçilmiştir.

2-2- Ekonomik Fizibilite Çalışması

Sistemin bütünüyle tasarımı sırasında yapılması gereken mali harcamalar aşağıda bulunun maliyet hesaplaması tablosunda verilmektedir. Gerek donanımsal gerekse yazılımsal araç ve gereçler maliyet hesaplaması içerisine katılmıştır.

2-2-1- Maliyet Hesaplaması

Robotun Maliyeti	4.348.60 YTL
Digital Kamera	150.00 YTL
Windows XP Professional	\$299.00
Borland C++ Builder 6.0	\$500.00
TOPLAM	5.000 YTL + 800.00\$

Tablo-1- Maliyet Analizi Tablosu

2-3- Legal Fizibilite Çalışması

Bu proje T.C. Yıldız Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Bilgisayar Bilimleri Mühendisliği Bölümü tarafından hazırlanacaktır. Bütün hakları bölüme aittir. Bölümün izni olamadan programın herhangi bir parçası ya da tamamı hiçbir yolla çoğaltılamaz ve başka programlarda kullanılamaz.

2-4- Kullanılabilirlik Fizibilitesi :

Kurduğumuz bu sistem en son yazılım araçlarıyla geliştirilmekte olup, işlevsel olarak istenilen görevi yapabilmektedir. Ayrıca ana sisteme entegre olacak bu sistem beklenen performansı ile sisteme uygun olarak tasarlanmıştır.

2-5- Alternatif Fizibilite :

Sistemi teknik, ekonomik ve kullanılabilirlik açıdan değerlendirdiğimizde bir çok çözüm yolu ile karşılaşmaktadır. Bizim çözümümüz sistemi en ekonomik ve kullanılabilir hale getirmektedir.

2-6- Gant Diyagramı :

Sistem analizi aşamasında sistemin zaman planlaması oluşturulmuştur. Zaman planlaması yani Gant diyagramı şekil-1' de aşağıda ayrıntıları ile gösterilmiştir.

3- SİSTEMİN YAPISI

Proje kapsamında yukarıda da belirttiğimiz geliştirilmekte olan robot için yön bulma işleminde büyük bir öneme sahip olan robotun önüne gelen engelleri tanıyarak yolunu ona göre çizmesi işlenmektedir. Sistemin temel olarak kazandırılması istenen özellik engel tanıma olarak nitelendirilebilir.

Bu işlemi yaparken izlediğimiz metotlar ve metotlar için kullandığımız algoritmalar aşağıda gösterilmektedir.

Yaptığımız işlemler sırasıyla aşağıdaki gibidir.

3-1- Resim üzerinde bulunan parazitlerin yok edilmesi:

Bu işlem elimizde yer alan bir görüntü üzerinde var olan bazı parazitleri temizlemekten ibarettir. Bunun için kullanılan averaging filter, median filter metotlarının ikisini de deneyerek en iyi sonucu veren median filter'ı seçtik.

3-1-1- Median Filter :

Median filter lineer olmayan ve resim üzerinde yer alan parazitleri yok eden bir alçak geçiren filtre çeşididir. Bu yöntemin çalışma şekli aşağıdaki gibidir.

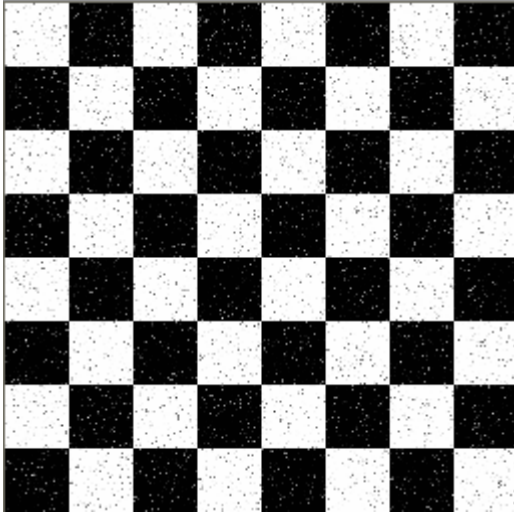
17	34	42	34	44	56	54	35
23	27	45	56	67	78	89	12
14	15	13	35	37	34	24	28
25	67	89	90	45	26	24	27
17	34	42	34	44	56	54	35
23	34	45	56	67	78	89	12
14	15	13	35	37	34	24	28
25	67	89	90	45	26	24	27

Tablo- 2 – Örnek Resim Pikselleri

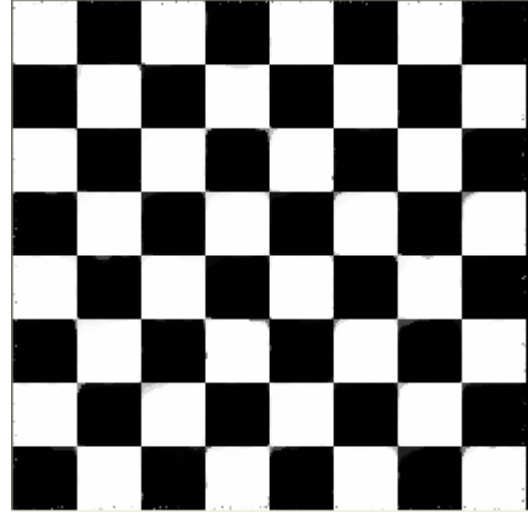
Yukarıdaki şeklin örnek bir resim olduğunu düşünelim. Resmimiz içerisinde seçtiğimiz 3*3 lük matris şeklindeki alanımızdaki piksel kodlarını alıp sıralıyoruz. Daha sonra sıraladığımız matristeki N/2 . sıradaki elemanı belirleyip resmimiz içindeki N/2 . sırada yer alan piksel yerine yazıyoruz. Bu işlemi yaparken dikkat etmemiz gereken şey sınırlarda yer alan renk kodlarını yeni bir dosya ya yeniden yazmak ve diğer median filter uyguladığımız pikselleri de gerekli yerlerine düzgün bir şekilde yazmaktır.

Sıralı Matrisimiz → 13 14 15 17 23 27 34 42 45

Aşağıdaki şekillerde yaptığımız bu uygulamaya ait denemeler bulunmaktadır. Yaptığımız çalışmanın iyi anlaşılabilmesi için resmin üzerine çeşitli parazitler yapılmış örnekler kullandık.



Şekil-2- Satranç Tablosu (Orijinal)



Şekil-3- Satranç Tablosu (Median Filter)

3-2- Resim üzerinde tanınacak olan nesnenin sınırlarının belirlenmesi :

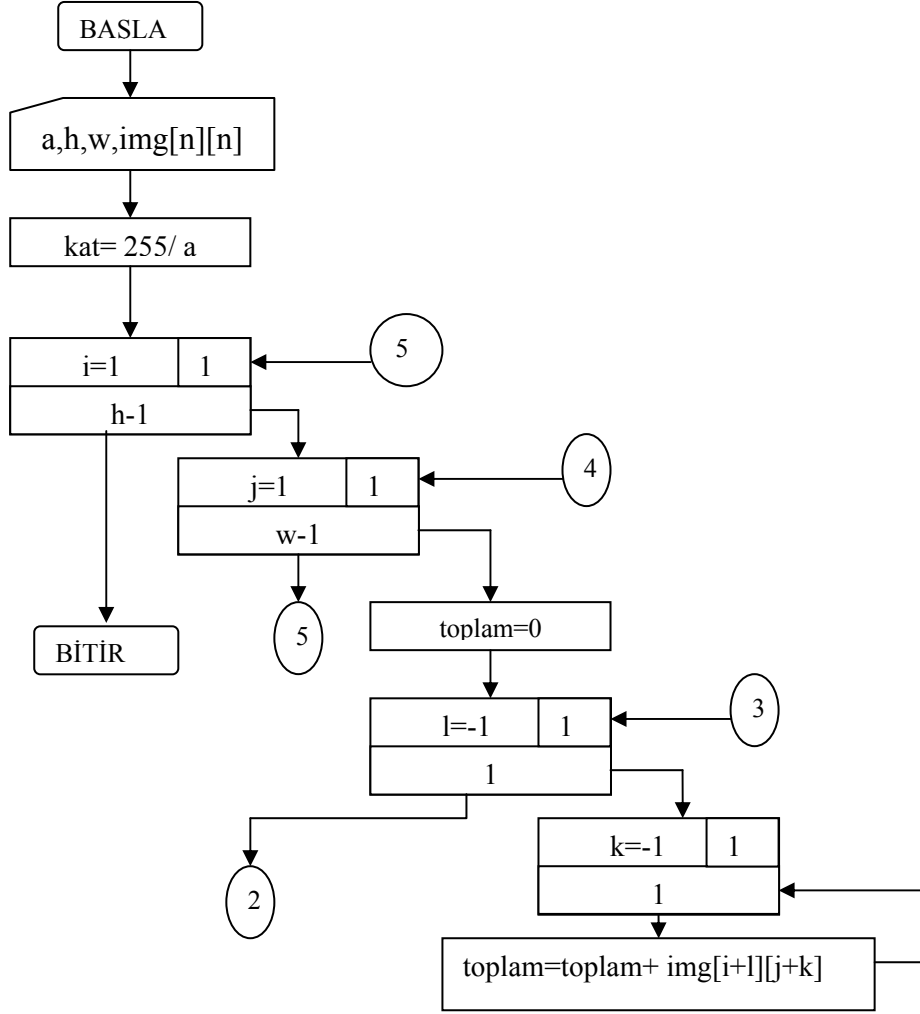
Bu işlem başlıktan da anlaşılacağı gibi robot üzerinde yer alacak olan kameradan elde edilen görüntülerde yer alan bazı temel şekillerin tanınması aşamasında sınırlarının ortaya çıkarılması işlemidir.

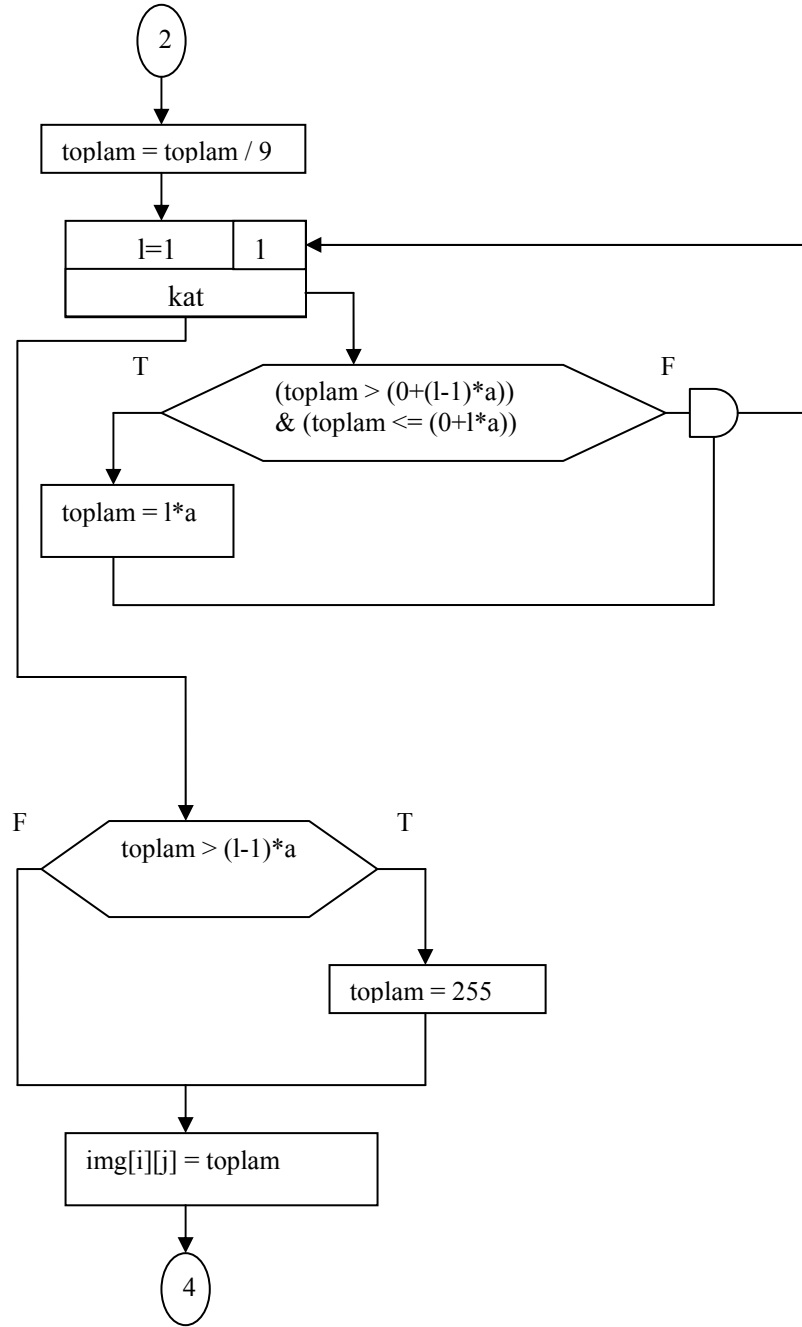
Bu işlem için geliştirilmiş bir çok yöntem kullanılmaktadır. Bizim yöntemimizi, bu yöntemler içerisinde en iyi sonucu veren Canny Edge Detector yöntemi ve kendimizin oluşturduğu farklı bir yöntem kullanarak belirledik. Fakat bu ikisi arasından da kendi yöntemimizi seçerek bu işlemi gerçekleştirmiş olduk.

3-2-1- Yöntemin Açıklaması:

Nesne tanımda, yapılması gereken en önemli çalışmalardan biri Edge Triggıng (Sınır belirleme) problemidir. Bunun için yapılan bir çok çalışma vardır. Projemizde sınır tanımayaya yönelik olarak kendi yöntemimizi yarattık ve başarılar sonuçlar elde ettik. Yöntemimizin çıkış amacı nesneler arasında ki renk farklılığında yola çıkmaktadır. Robotun çalışacağı ortam ve önüne çıkacak olan engeller birbirinden renk olarak ayrılabilir. Örneğin beyaz bir zeminde siyah bir kutunun robotun karşısına çıkma durumunu göze alırsak, robotun cismi renginden bu ortamdan ayırması gerekmektedir. Çalışmalarımızda pgm dosyaları kullandığımızdan resimler siyah beyaz ve renk kodları 0 – 255 arasında olacaktır. Girilen hassasiyet derecesine bağlı olarak, 0 - 255 arasını bölümlendiriyoruz. Örneğin hassasiyet derecesi 50 girilirse aralıklar 0 – 50 , 50 -100, 100... şeklinde olmaktadır. Daha sonra resim piksel piksel taranarak, o piksel ve etrafında 9 değeri ortalamaşı alınıyor. Çıkan sonuç hangi aralıkta ise o aralığın üst sınırı, tekrar alınan piksel değeri olarak atanıyor. Bu işlemler bütün pikseller için yapılıyor. İşlemlerden sonra resimde aynı renk koduna sahip şekiller oluşmuş oluyor. Böylelikle resmimizi renklere göre şekillere ayırmayı başarmış oluyoruz. Sonrasına ise sınırları çizme işlemi geliyor. Bu arada resmin başından sonuna taranarak, elde edilen her değeri için etrafında kendi renk kodundan başka renk var mı diye kontrol ediliyor var ise o piksel başka bir resim matrisinde 0 (siyah) değeri alıyor. En sonda dosyaya siyah beyaz resim yazılıyor.

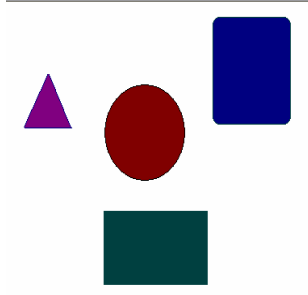
3-2-2- Yöntemin Akış Diyagramı:



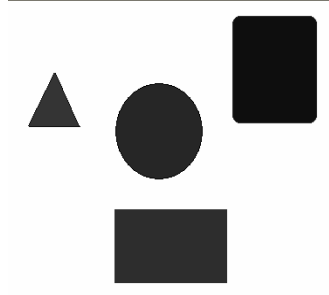


3-2-3- Yöntemin Sonuçları :

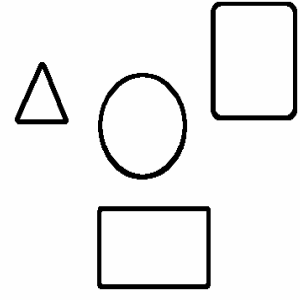
Sınırları belirleme işlemini yaparken ilk yaptığımız işlem median filter olmuştur. Daha sonra da kendi oluşturduğumuz algoritma ile sınır belirleme işlemini yaptık. Bu işlemleri deneme aşamasında web kamerası görüntülerinden yararlandık. Kendi oluşturduğumuz temel şekillerin web kamerası ile birkaç fotoğrafını çektik. Ayrıca bu işlemi deneme aşamasında hazır resimlerden de yararlandık. Bu yöneme ait sonuçlar aşağıdaki şekillerde verilmektedir.



Şekil-4 – Paintte
Hazırlanmış Resim
(Orjinal)



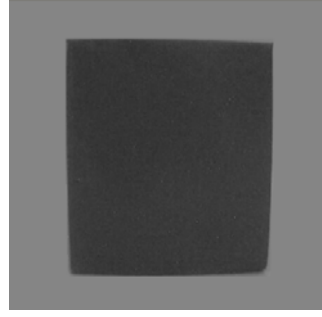
PGM (Formatında)



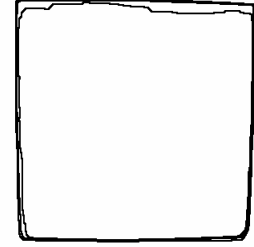
Sınır (Edge) Belirlenmiş
Resim



Şekil – 5 – Web Kamera
Görüntüsü (Küp ÖnYüz)
Yakın Çekim



PGM (Formatında)



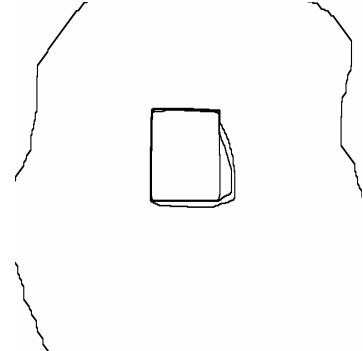
Sınır (Edge) Belirlenmiş
Resim



Şekil – 6 – Web Kamera
Görüntüsü (Küp ÖnYüz)
Uzak Çekim



PGM (Formatında)



Sınır (Edge) Belirlenmiş
Resim

3-3 - Resim üzerinde tanınacak olan nesnenin sınırlarının belirlenmesi (Yöntem-2):

Sınır belirleme şeklin arka plandan çıkarılması ve resimdeki gürültülerden kurtarılması için önemli bir yöntemdir. Burada yapılan çalışma bir sınıflandırma çalışmasıdır. Programdan girilen bir hassasiyetlik(değeri a olsun) değerinde göre bu sınıflandırma yapılmaktadır.

Yapılan çalışma şu şekildedir. Resimde karşılaşılan ilk pikselin değeri bir diziye atılır (değeri x olsun). Daha sonra gelen piksellerin değerleri $(x-a)$, $(x+a)$ aralığında ise o pikselin değeri x yapılır. Eğer bu aralıktan başka bir değerde ise bu pikselin değeri de diziye atılır (değeri y olsun). Gelen bir diğer piksel $(x-a)$, $(x+a)$ veya $(y-a)$, $(y+a)$ değerlerinde ise o pikselin değeri x veya y yapılır. Eğer değeri yine bu aralıklarda değilse pikselin değeri yine diziye atılır. Bütün bu işlemler aynı sıra ile resmin sonunda kadar yapılır. Bu işlemde resimde aynı renk kodlarından oluşan şekiller mevcut olacaktır. Ve her şeklin renk değerinin tutanda bir dizi elde etmiş olacağız ki bu dizi sonra resmin segmentasyonunda kullanılacaktır.

3-4- ŞEKİL TANIMA

3-4-1-Şekil Tanıma Yöntemi (1)

Robotun önüne çıkan cisme bağlı olarak hareketini değiştirebilmesi için cismin hangi şekil olduğunu, görüntünün neresinde başladığı ve uzunluk – genişlik bilgilerini bilmesi gerekir. Şekil tanınmadan diğer aşamalara geçilemeyeceğinden ilk aşama olarak şekil tanınmalıdır. Şekil tanıma yöntemi olarak 2 yöntem belirledik.

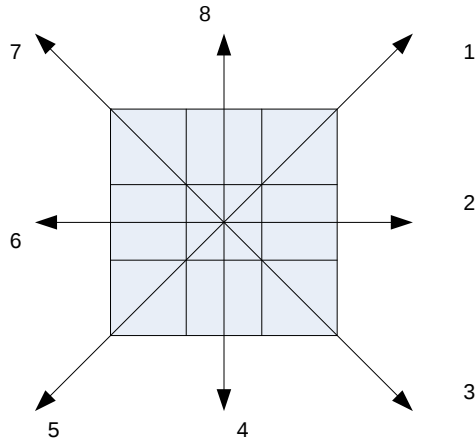
Çalışma Prensibi

Kare, dikdörtgen, yuvarlak ve üçgen özellikleri göze alındığında hepsi kendi içinde tutarlı ve değişmez özellikler sağlamaktadır. Örneğin Kare: 4 kenarı birbirine eşit vb..

Eğer bu özellikler bir şekilde ölçülebilir veya belirlenebilir ise şekil tanınabilir. Bunun için şeklin kenarlarını belirlemek ve bir şekilde ölçebilmek gerekir.

Şeklin kenarlarını ölçebilmek için geliştirilen yöntem, şeklin sol üst köşesinden başlanarak kenarın aldığı yolun hangi eksen doğrultusunda gittiği bilgisi tutmaktır.

Bunun için bir pixelin etrafındaki bitler eksen çizgilerine göre numaralandırılmıştır.



Şeklin sol üst köşesinden başladıktan sonra o an üzerinde bulunduğumuz pixel ortada olmak şartıyla gidilebilecek yönler aşağıdaki gibi belirlenmiştir. Şeklin etrafını dolaşırken izlenecek yön saatin dönme yönünde olacağından burada da rakamlar öncelikleri belirtmek şartıyla saat yönünde bir sıra belirlenmiştir. Taramaya matrisin

sol üst köşesinden başlandığından bulunan ilk pixelin gidebileceği en yukarıdaki yön 1 olacağından başlama noktası burası alınmıştır. Yönlerin belirlenmesinden sonra, izlenilen yol şu şekildedir; o an üzerinde bulunduğumuz pixel ortadaki

Yönler

ixeldir. Etrafındaki pixeller kontrol

Şekil-6-

edilerek rengi o anda üzerinde çalıştığımız renkle aynı olan ve öncelik sırası en yüksek olan pixele atlanır. Atladığımız pixel artık ortadaki pixeldir ve aynı işlemler bu pixel içinde tekrarlanır. Tek bir renk için hangi yönde kaç kere hareket ettiği kaydedilir. Bu

yönler neden kaydedilir? Bir kareyi ele alırsak elde edilen sonuçlar incelendiğinde 2, 4, 6 ve 8 yönlerinde hareket sayılarının eşit olduğu çok rahatlıkla gözlemlenebilir. Eğer her cisim için gidilen yönlerin sayıları tutulabilirse, bulmak istenilen şekillerin özellikleri bulunan sonuçlarla karşılaştırılabilir. Diğer bir örnek olarak r yarıçaplı bir daireyi ele alalım, bir dairenin etrafında dolaşmak demek karşılıklı kenarlarda eşit sayıda yol almak demektir. Bunun sebebi dairenin kenarındaki her nokta için tanjantların toplamı 0 olmasıdır. Daire için sonuçlar incelendiğinde (0 hatasız bir daire için) 1, 3, 5 ve 7 yönlerindeki hareket sayıları birbirine eşit ve 2, 4, 6 ve 8 yönlerindeki hareket sayıları birbirine eşit olmalıdır.

Programda düzgün bir kare için alınmış sonuçlar aşağıdaki gibidir.

Hareket Yönleri	Miktar
1	1
2	101
3	1
4	107
5	1
6	101
7	1
8	107



Sonuçlar incelendiğinde karenin aslında gözle görüldüğü kadar düzgün olmadığı aslında bir dikdörtgen olduğu anlaşılmaktadır. 2 ve 6, 4 ve 8 birbirlerine eşittir.

Programda düzgün bir daire için sonuçlar aşağıdaki gibidir.

Hareket Yönleri	Miktar
1	40
2	59
3	40
4	59
5	40
6	59
7	40
8	59



Sonuçlar incelendiğinde Resimdeki şeklin düzgün bir daire olduğu anlaşılmaktadır.

Yukarda anlatıldığı şekilde hareket yönleri sayıldığında cisimler için tutarlı ve karakteristik sonuçlar elde edilebilmektedir. Buradaki önemli nokta sonuçları ne şekilde yorumlayıp ne şekilde bir sonuç üreteceğimizdir. Bunun için hata oranı belirlenmelidir. Bu hata oranı şekli tanımada ayrıntıya ne kadar bakılacağına karar veren değişkendir. Bu değer programcı tarafından belirlenebilir.

Kare veya Dikdörtgen için sonuçların yorumlanması:

Yukarıda anlatıldığı gibi bir hata oranı belirlenmelidir. Biz bu örneğimizde 10 belirleyelim. Yukarıdaki kare örneğine bakıldığında 1,3,5,7 değerlerinin ve $|2-6|$, $|4-8|$ farklarının belirlenen hata değerinden düşük olduğu görülmektedir. Bu şartlar altında bunun bir dikdörtgen olduğu söylenebilir. Kare özelliği için $2 = 4 = 6 = 8$ olmalıdır. Eğer bu özellikte sağlanırsa bu şeklin bir kare olduğu söylenebilir. Burada belirlenen hata oranı dikdörtgen şeklin ne kadar oranda yamuğa benzeyebileceğinin belirler. Çünkü bir dikdörtgen her ne kadar yamuğa başlamaya başlarsa 1,3,5,7 değerleri ona göre değişiklik göstermeye başlar. Bu durumda yamuğun şekline göre de 1,3,5,7 deki değişim miktarına yakında değerde $|2-6|$ ve $|4-8|$ farkları da o kadar atmaya başlar aralarında doğrusal bir bağlantı mevcuttur. Eğer hata oranı 0 verilirse 0 hatalı bir tam kare aranıyor demektir.

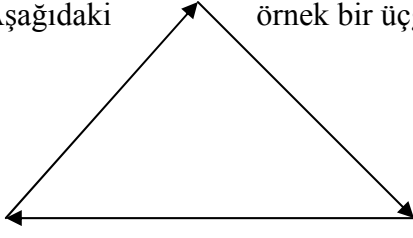
Yuvarlak ve Elips için sonuçların incelenmesi

Bir önceki örnekte olduğu gibi burada da bir hata oranı belirlenmelidir. Bu hata oranı şeklin tam bir yuvarlak veya elips veya bozulmaya uğramış bir elips olup olmadığının kararını verecektir. Yuvarlak için yukarıdaki örnek incelenirse 1,3,5,7 değerlerinin eşitliği ve 2,4,6,8 değerlerinin eşitliği durumunda şeklin tam bir yuvarlak olduğu söylenebilir. Elips için 1,3,5,7 değerleri yine birbirine eşit ancak $2=6$ ve $4=8$ şartları sağlanıyorsa şeklin bir elips olduğu rahatlıkla söylenebilir. Burada hata oranı yuvarlak veya elips şekillerinin deforme olduğu durumları tolare edecektir. Burada 1,3,5,7 birbirine eşit değil, 2,4,6,8 birbirine eşit değil ancak aralarında ki fark hata oranından küçük ise şekil yine bir yuvarlak veya bir elips olarak kabul edilebilir.

Üçgen için sonuçların incelenmesi

Üçgen için ayır edici hareket yönleri 1,3 ve 6 dır. Bunun sebebi üçgen üzerinde hareket ederken algoritmanın aşağıdaki yolu izlemesidir. Bu noktada eş kenar bir üçgen için bu hareket yönleri eşit olmalıdır. Üçgende taban genişliğini belirleyen değer 6. hareket yönünde tutulmaktadır. Herhangi bir özelliği bulunmayan bir üçgende 1,3 değerleri birbirine yakın, 6 değeri bunlardan bağımsızdır. Ve 2,4,7,8, değerleri 0 yakın değerlerdir. Burada deformasyon üçgenin belli bir açıyla dönmesi durumunda oluşur. Hata oranı göz önünde bulundurulursa 2,4,7,8 değerlerinin 0'a olan uzaklığı ile 1,3 değerleri arasında doğru bir bağlantı olacaktır. Bu özellik ve hata payı kullanılarak üçgenler saptanabilir.

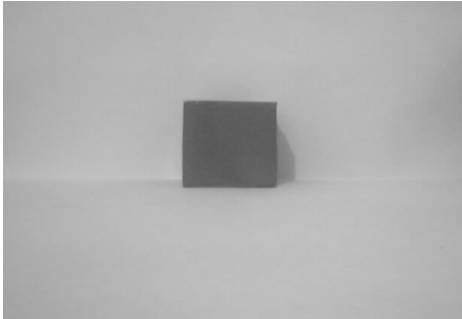
Aşağıdaki örnek bir üçgen için yapılmıştır.



Hareket Yönleri	Miktar
1	108
2	6
3	109
4	0
5	1
6	221
7	1
8	1

şeklindedir.

Yukarda çalışılan resimler paintte çizilmiş düzgün resimlerdir ancak robottan alınacak resimler gürültülü olacağından yöntemin kamera görüntüsünden alınmış resim ile de denenmesi gerekir.



Bu şekil için çıkan sonuçlar aşağıdaki gibidir.

Hareket Yönleri	Miktar
1	2
2	119
3	13
4	94
5	16
6	113
7	5
8	116

Şeklindedir. Burada hata oranı 20 gibi bir değer alınırsa bunun bir dikdörtgen olduğu rahatlıkla söylenebilir. Bu yöntemin bir avantajı aynı resimde bir den çok şekil olması

durumunda her şekli tanıyabilmesidir. Çıktı olarak bir matris üreten bu yöntem matriste bulunduğu bütün şekillerin hareket değerlerini ve başlama yerlerini tutmaktadır.

3-4-2-Şekillerin Uzunluklarının Belirlenmesi

Bu yöntemin bir diğer avantajı hareket yönlerini tutarken bir yandan da cisimlerin kenar uzunluklarını tutmakta ya da hesaplanmasını kolaylaştırmaktadır. Her bir şekil için ayrı ayrı incelenmelidir.

Kare için veya dikdörtgen için 2, 6 değerleri genişlik ve 4,8 değerleri yüksekliği pixel boyutunda tutmaktadır. Deformasyon durumunda bu değerlerin ortalaması alınarak dahi genişlik ve uzunluk değerleri rahatlıkla hesaplanabilmektedir.

Yuvarlak veya elips için 2 ve 6 değerleri yuvarlak için yarıçap, elips için büyük genişliğin uzunluğunu tutmaktadır. Elips için küçük uzunluk 8 ve 4 değerlerinden elde edilebilir.

Üçgen için 6 değeri daha öncede anlatıldığı gibi, taban uzunluğu tutmaktadır. Burada 1 veya 3 değerleri ile taban uzunluğu pisagor denklemine konulursa yükseklikte elde edilmiş olur.

3-4-3-Şekil Tanıma Yöntem (2)

Robotun önüne çıkan engellerin belirlenerek, robota yön verme işlemi için gerekli olan işlemlerimizden bir tanesi bu şeklin ne olduğunun anlaşılması, ayrıca şekle ait koordinatların belirlenmesi ve resim içerisinde başka şekiller var ise bunlara olan uzaklıklarının belirlenmesidir. Bu amaçla resim üzerindeki belli başlı şekillerin tanınması için bir yöntem belirledik.

Sırasıyla Şekil Tanıma Aşamasında Gerçekleştirilen İşlemler

3-4-3-1- Şeklin Resimden Ayrılması İşlemi (Segmentation)

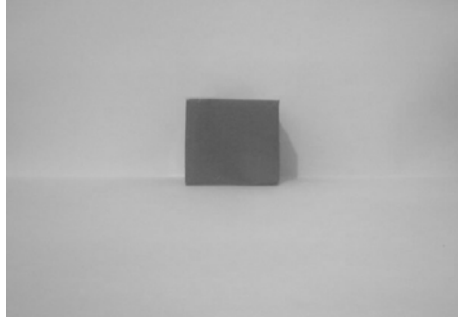
Bu yöntemde şeklin tanınması aşamasında ilk olarak yapılan işlem resim üzerinde yer alan cismimizi arka plandan ayırmaktır. Bu işleme görüntü işleme alanında segmentation denilmektedir.

Bizim segmentation için uyguladığımız yöntem gerçek şeklin arka planla olan renk farkından yararlanılarak düşünülmüştür. 2. yöntemle sınır belirlerken resimde

bulduğumuz farklı renk tonlarının değerlerinden yararlanmaktayız. Ayrıca 2. yöntemde belirlediğimiz hassasiyet derecesi ile resimde yer alan renk tonlarını istediğimiz değerlerde azaltabilir yada çoğaltabiliriz. Bu bize resmin daha net bir şekilde ayrılmasında yardımcı olan bir faktördür.

Şimdi bir örnekle yöntemi açıklamaya çalışacağım.

Orijinal Resim



a- İlk olarak resimde bulunan farklı renk tonlarını 25'lik bir hassasiyet derecesiyle elde ettik. Bu değerler sırasıyla :

152 178 126 99

olmak üzere 4 renk tonu olarak bulunur.

b- Daha sonra bu değerleri toplayıp adet sayısına bölerek aritmetik ortalamasını elde etmiş olduk.

$$\mathbf{A.Ort} = (152 + 178 + 126 + 99) / 4$$

$$\mathbf{A.Ort} = 138$$

n=4 bulunan farklı renk tonu

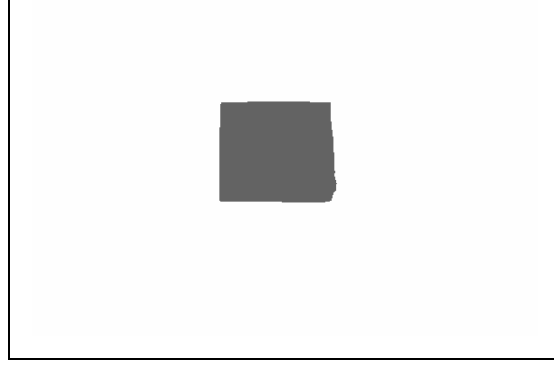
c- Bulunan bu ortalama renk tonunu %20'si alınarak ortalama değerden çıkarılır.

$$(138 * 20) / 100 = 28$$

$$\mathbf{sonuç} = 138 - 28 = 111$$

d- Daha sonra bulduğumuz bu değeri eşik değeri olarak belirleyip bu değerden küçük olan renk kodları şeklimize ait olup renk tonları korunur, bu değerden büyük olanlar ise şekle ait olmadığından dolayı beyaz renk tonu yani 255'e boyanır. Böylece şekil ve arka plan birbirinden ayrılmış olur.

Ayrılmış Resim (Segmentation)



3-4-3-2- Şeklin Tanınması

Bu aşamada elimizde bulunan segmente edilmiş resim üzerinden şekil tanınmıştır. Bu işlem için yapılan işlemler ise aşağıda sırasıyla verilmektedir.

- a- Resim üzerinde şeklin ilk başladığı noktadan başlayarak bütün satırlarda yer alan piksel sayıları bir dizi içine kayıt edilir.

Mesela dizimiz sat[n] olsun ;

sat [0]= 150 piksel , sat[1]=154 piksel sat [n]= x piksel gibi

- b- Resim üzerinde şeklin ilk başladığı noktadan başlayarak bütün sütunlarda yer alan piksel sayıları bir dizi içine kayıt edilir.

Mesela dizimiz sut[n] olsun ;

sut [0]= 180 piksel , sut[1]=198 piksel sut [n]= x piksel gibi

- c- Daha sonra bu bulduğumuz dizilerden satır ve sütun ortalamaları bulunur.

- d- Satır ve sütun ortalamalarının bulunması ile her iki yönde şekil tanımak için bakılacak piksel sayısının belirlenmesi için bu oranların %25'i alınır ve bu değerler belli edilir.

Mesela; Ortalaması 100 olan bir satır için $(100*25)/100=25$

Ortalaması 120 olan bir sütun için ise $(120*25)/100 = 30$ değerleri elde edilir.

- e- Yukarıda elde ettiğimiz değerler satır için, şeklimizin başından 25 satırlık bir alan kontrolü, ortasından 25 satırlık bir alan kontrolü ve sonundan 25 satırlık bir alan kontrolü ile gerçekleşir. Bu alan kontrolü başından,sonundan ve ortasından elde ettiğimiz 25 satırlık değerlerin ortalaması alınarak bulunur. Aynı işlem sütun için de gerçekleştirilir.

Örneğin yukarıdaki resme ait elde edilen ortalamalar şu şekildedir.

Satır için :

Tepe kısmın ortalaması → 120

Orta kısmın ortalaması → 126

Alt kısmın ortalaması → 124

Sütun için:

Sol kısmın ortalaması → 117

Orta kısmın ortalaması → 120

Sağ kısmın ortalaması → 102

- f- Buraya kadar şekli tanıyabilmemiz için gerekli tüm bilgileri elde etmiş olduk, sıra bu cismin nasıl bir geometrik şekle geldiğini öğrenmeye geldi. Bu aşamada düşünülen yöntem şu şekildedir.

→ Eğer bu şekil bir kare ise veya kareye çok yakın bir resim ise :

Yukarıda bulunan satır,sütun için başlangıç,orta ve son ortalama değerleri %20 oranında esneklik ile birbirine benzer değerler ise bu şekil bir karedir diyebiliyoruz.

→ Eğer bu şekil bir dikdörtgen ise veya dikdörtgene çok yakın bir resim ise :

Satır için başlangıç,orta ve alt ortalama değerleri %20 esneklik oranında birbirleriyle benzer, sütun için başlangıç,orta ve alt ortalama değerleri %20 esneklik oranında birbirleriyle benzer ise bu şekil dikdörtgendir diyebiliyoruz.

→ Eğer bu şekil bir yuvarlak ise veya yuvarlağa çok yakın bir resim ise :

Bir yuvarlak olduğunu anlayabilmemiz için satır ve sütun için baş ve son ortalama değerleri birbirine %20 hassasiyette yakın ayrıca orta ortalama değerleri ise satır ve sütun için %10 oranında birbirine yakın ve köşe oranlarından büyük ise bu bir yuvarlaktır diyebiliyoruz.

→ Eğer bu şekil bir üçgen ise veya üçgene çok yakın bir resim ise :

Yukarıda ki temel şekillerimizden sonra bir şeklin üçgen olduğunu anlayabilmek için satır için başlangıç ve son ortalamanın toplamı orta ortalamadan küçük veya sütun için baş ve son ortalamanın toplamı yine orta ortalamadan küçük ise bu bir üçgendir diyebiliyoruz.

Şekillerde ortalamaların %20 olarak belirlenmesi denemler sonucunda elde edilen sonuçlarla belirlenmiştir. Bu değerler bizi büyük oranda başarıya götüren değerlerdir.

Resmin şeklinin belirlenmesi boyutlarının belirlenmesi ve robotun hareketinin hangi yöne olacağının belirlenmesinde önemli bir noktadır. Bu yüzden bu yöntem bize resim içerisinde şeklin koordinatlarını ve ne tip bir şekil olduğunu göstermektedir.

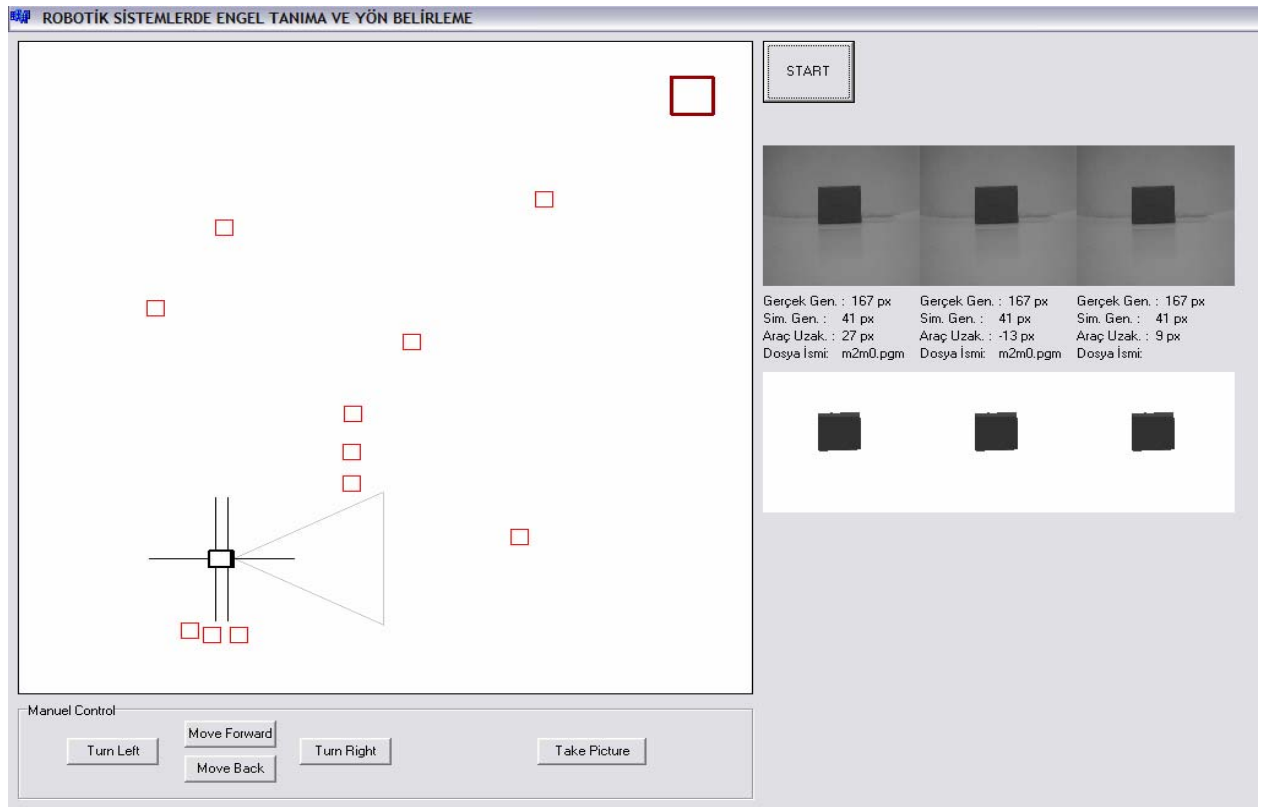
3-5- GENEL ANLAMDA ENGEL TANIMA İŞLEMİ

Engel tanıma işlemi robotun yönünü belirleme işlemi için kullanılmaktadır. Bu aşamada gerçekleştirilen iki yöntemden birincisi resim içerisinde bulunan bütün engellere ait genişlik,yükseklik gibi bilgileri çıkarmaktadır. İkinci yöntem de üzerinde tek engel bulunan resimlere ait yükseklik,genişlik bilgileri çıkarabilmektedir.

Projenin geri kalan kısmında bu yöntemlerden ilki kullanılmak üzere iyileştirilmiştir. Bu yöntem bundan sonra robotun ilerlemesini, yön bulmasını bilgisayar ortamında gerçekleştirmek için bir sümilasyon ile kullanılacaktır.

3-6- SÜMİLATÖR

Robot üzerinde yer alan kamera ile ve çevresinde bulunan sensörleri yardımıyla yolunu belirlemesi amaçlanmaktadır. Bu aşamada bu işlemin bilgisayar ortamında gerçekleştirilebilmesi için c++ gui'si kullanılarak görsel olarak robotun kamerası ve sensörlerinden bilgilerin alınması ile yolunu izlemesi tasarlanmıştır. Aşağıdaki şekilde bu işlem genel hatlarıyla gözlemlenmektedir.

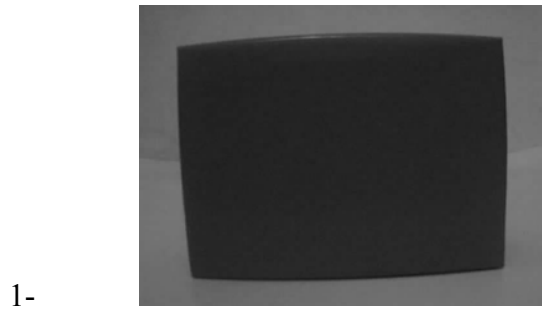


Bu sistem üzerinde robotu temsil eden dikdörtgen şeklinde bir cisim bulunmaktadır. Bu cisim üzerinde de ön kısmını belirtir bir işaret bulunmaktadır. Ayrıca sensörleri ifade eden sağ ve sol tarafta ikişer adet ve arka ve ön tarafta da birer adet olmak üzere 6 adet sensör bulunmaktadır. Robot üzerinde bulunduğu alan üzerinde 90 derecelik yönler dönebilmektedir. İleriye ve geriye doğru hareket kabiliyeti bulunmaktadır.

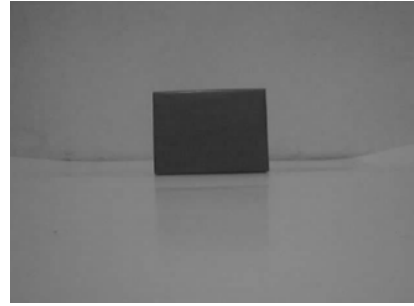
Form üzerinde engellerin belirlenmesi mouse tıklaması ile işaretlenerek koyulmaktadır. Bu engeller bizim şekil tanıma yöntemi ile bilgilerini elde ettiğimiz resimlerden oluşmaktadır. Bu engeller için kendimiz bir veri tabanı oluşturduk ve içerisine temel şekilleri ifade eden engellere ait resimleri koyduk. Bu veritabanına ait bazı örnek resimler aşağıda görülmektedir. Şekil tanıma işlemini gri seviyesinde yaptığımız için bu resimler pgm formatındadır.

Örnek Veritabanı Resimleri:

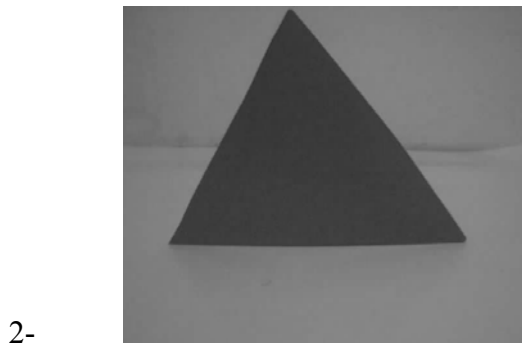
İçerisinde tek engel bulunan resimler:



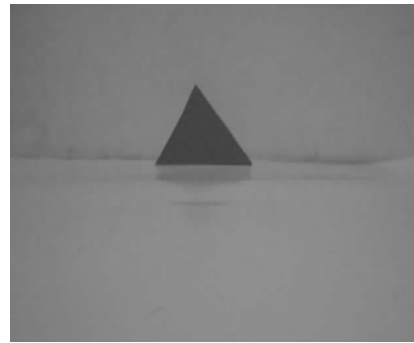
Şekil-8-Engel(Küp1)



Şekil-9-Engel(Küp2)

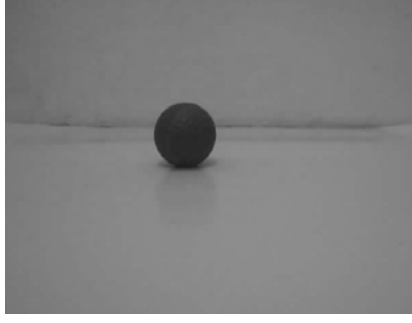


Şekil-10-Engel(Üçgen1)



Şekil-11-Engel(Üçgen2)

3-



Şekil-12-Engel(Daire1)



Şekil-13-Engel(Daire2)

Engel veritabanında kullanılmak üzere üçgen, yuvarlak, daire şekillerinden farklı üç boyutta şekiller örnek olarak kullanılmıştır. Ayrıca bu resimlerin ikiyeşerli resimleri de farklı kombinasyonlarla elde edilip kullanılmak üzere saklanmaktadır.

Çalışma şekli:

Yukarıda bulunan programa ait resim ve engellere ait resimlerden de anlaşılacağı üzere robotu temsil eden cisim ve engelleri temsil eden şekiller form üzerinde cismin hareket yönlerinin belirlenmesi için kullanılacaktır. Robot başlangıç noktasından bitiş noktasına varabilmek için karşısına çıkan engellere göre yönünü bularak hareket etmektedir. Engeller robotun görüş alanına girdiğinde veritabanından random olarak seçilen şekiller engel olarak şekil tanıma yöntemiyle tanınıp genişliğine göre, başlangıç ve bitiş koordinatlarına göre yol belirleme işlemi yapılmaktadır.

Veri tabanında saklanan engeller robot engelle karşılaştığında bir picture box ile form kenarında gösterilmektedir. Böylece robotun karşısına çıkan engelleri de görebilme fırsatı yakalamış oluyoruz.

Ayrıca robot kenarlara belirli mesafede hareket etmektedir. Bunu sensörleri yardımıyla kontrol etmektedir. Bu şekilde robotun kenarlara çarpması da engellenmektedir.

3-7-YOL BULMA ALGORİTMASI

Pseudo Code

START

Do

// Araç 4 yönündeyken

```
if arac.direction==4
    takepicture();
    if bilgicount==-1
        movecar();//normal yada engelboyutuartınormal boy kadar
    else if (bilgicount>0)
        movecar(); //daha az ilerle
```

// Araç 1 yönündeyken

```
if arac.direction==1
    takepicture();
    if bilgicount==-1
        movecar();//normal yada engelboyutuartınormal boy kadar
    else if (bilgicount>0)
        movecar(); //daha az ilerle
```

// Araç 3 yönündeyken

```
if arac.direction==3
    takepicture();
    if bilgicount==-1
        movecar();//normal yada engelboyutuartınormal boy kadar
    else if (bilgicount>0)
        movecar(); //daha az ilerle
```

// Araç 2 yönündeyken

```
if arac.direction==3
    takepicture();
    if bilgicount==-1
        movecar();//normal yada engel boyutu + normal boy kadar
    else if (bilgicount>0)
        movecar(); //daha az ilerle
```

// 4 Yön için Eğer Engel var ise duruma göre yön belirle

```
if araba.direction==4 && bilgicount>0
    check sensors and determine direction
if araba.direction==3 && bilgicount>0
    check sensors and determine direction
if araba.direction==2 && bilgicount>0
    check sensors and determine direction
if araba.direction==1 && bilgicount>0
    check sensors and determine direction
```

// 4 yön için araçtaki sensörlerin durumuna göre yön belirle

check sensors direction 1

check sensors direction 2

check sensors direction 3

check sensors direction 4

While arac.x != lastpointx && arac.y != lastpointy;

END

SONUÇ:

Bu çalışmada hedeflenen işlem geliştirilmekte olan robotun kamera desteğiyle önüne çıkan engellere göre uyum sağlayarak yönünü bu engellere göre belirleyebilmesidir. İlk aşamada robotun belirli aralıklarla kamera görüntüsü alarak bu görüntüyü siyah beyaz bir resim formatında saklaması ve bu saklanan resimlerin kullandığımız şekil tanıma yöntemiyle bizim için gerekli olan özelliklerin bulunması sağlanmıştır. Daha sonrada bu özellikleri kullanarak robotun hangi yöne ne kadar uzunlukta gideceği belirlenmiştir.

KAYNAKLAR

- 1- R.Gonzales,R. Woods, "**Digital Image Processing**", Prentice Hall,© 2002
- 2- Nicolas Mailllot,Monique Tonnat,Celine Hudelot,"**Ontology Based Object Learning and Recognition : Application to Image Retrieval**", INRIA Sophia Antipolis - Orion Team, 2004
- 3- Rebecca Fay, Ulrich Kaufmann, Friedhelm Schwenker and Günther Palm ,"**Learning Object Recognition in a NeuroBotic System**", University of Ulm Department of Neural Information Processing, 2004
- 4- Alexander C. Berg Tamara L. Berg Jitendra Malik, "**Shape Matching and Object Recognition using Low Distortion Correspondences**", Department of Electrical Engineering and Computer Science U.C. Berkeley, 2004

ÖZGEÇMİŞ(1)

Ad Soyad	Necmi ÖZTÜRK
Doğum Tarihi	01.07.1983
Doğum yeri	Düzce
Lise	1997 - 2001 Düzce Anadolu Meslek Lisesi
Staj Yaptığı Yerler	Nortel Networks Telekomünikasyon A.Ş. - İstanbul - (5 hafta) I-3S Yazılım Şirketi, Konya (6 hafta)

ÖZGEÇMİŞ(1)

Ad Soyad	Aytunç BEKEN
Doğum Tarihi	
Doğum yeri	
Lise	
Staj Yaptığı Yerler	