



**T.C.
YALOVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GEZGİN ROBOTUN GÖRME TABANLI VE GERÇEK ZAMANLI
OLARAK ENGELDEN SAKINIMI**

**YÜKSEK LİSANS TEZİ
Adem HİÇDURMAZ
175105005**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Adem TUNCER

HAZİRAN 2020



**T.C.
YALOVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GEZGİN ROBOTUN GÖRME TABANLI VE GERÇEK ZAMANLI
OLARAK ENGELDEN SAKINIMI**

**YÜKSEK LİSANS TEZİ
Adem HİÇDURMAZ
175105005**

**Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı**

Tez Danışmanı: Dr. Öğr. Üyesi Adem TUNCER

HAZİRAN 2020

YALOVA Üniversitesi Fen Bilimleri Enstitüsü'nün 175105005 numaralı Yüksek Lisans Öğrencisi Adem HİÇDURMAZ ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **“GEZGİN ROBOTUN GÖRME TABANLI VE GERÇEK ZAMANLI OLARAK ENGELDEN SAKINIMI”** başlıklı tezini aşağıda imzaları olan jüri önünde sunmuş ve oy birliği/oy çokluğu ile başarılı bulunmuştur.

	İmza	/Kanaati (Kabul/Red)
Tez Danışmanı :		/.....
Jüri Üyeleri :		/.....
		/.....

Teslim Tarihi : 25 Haziran 2020
Savunma Tarihi : 9 Temmuz 2020

ÖNSÖZ

Yüksek lisans eğitimim boyunca benden yardımlarını esirgemeyen, akademik anlamda bana her zaman destek olan ve doğru yolu gösteren sayın danışman hocam Dr. Öğr. Üyesi Adem TUNCER'e teşekkür ederim. Her zaman yanımda oldukları ve benden desteklerini hiçbir zaman esirgemedikleri için aileme teşekkür ederim.

Haziran 2020

Adem HİÇDURMAZ

Bilgisayar Mühendisi

İÇİNDEKİLER

	Sayfa No
ÖNSÖZ..	iii
İÇİNDEKİLER	v
KISALTMALAR	ix
ÇİZELGE LİSTESİ	xi
ŞEKİL LİSTESİ	xiii
ÖZET.....	xv
ABSTRACT	xvii
1. GİRİŞ..	1
1.1 Tezin Amacı ve Katkısı	2
1.2 Literatür Araştırması	3
2. ROBOT İŞLETİM SİSTEMİ (ROS)	7
2.1 ROS Sürümleri	8
2.2 ROS Mimarisi	8
2.3 ROS Dosya Sistemi	9
2.4 ROS Kavramları	10
2.4.1 ROS Master..	10
2.4.2 ROS Düğüm	11
2.4.3 ROS Konu....	12
2.4.4 ROS Mesaj..	12
2.4.5 ROS Servis..	13
2.4.6 ROS Eylem	14
2.5 ROS Araçları	14
2.5.1 rviz.....	14
2.5.2 rqt.....	14
2.5.3 rosbag.....	15
2.5.4 roslaunch....	15
2.5.5 ROS Konsolu ve Komut Seti	16
2.6 Benzetim Ortamı	17
2.6.1 Gazebo.....	17
2.6.2 SDF dosya biçimi	18
2.7 OpenCV..	18
2.8 Python Programlama Dili	19
2.9 Turtlebot...	19
3. GÖRÜNTÜ İŞLEME	21

3.1 Sayısal Görüntü İşleme.....	22
3.2 Görüntü İşleme Yaklaşımları.....	24
3.2.1 Düzeltme.....	24
3.2.2 Dönüşüm.....	24
3.2.3 Ayırıştırma.....	24
3.3 Görüntü Bölütleme.....	24
3.3.1 Bölge Tabanlı Görüntü Bölütleme.....	25
3.3.1.1 Bölge büyütmesi.....	25
3.3.1.2 Ayırma ve birleştirme	26
3.3.2 Kenar Tabanlı Görüntü Bölütleme.....	26
3.3.3 Eşik Tabanlı Görüntü Bölütleme.....	27
3.3.4 Kümeleme/Sınıflandırma Tabanlı Görüntü Bölütleme.....	27
3.3.4.1 K-ortalama... ..	28
3.3.4.2 K-ortaylar.....	28
3.3.4.3 K-En yakın komşuluk	28
3.3.4.4 Bulanık c-ortalama.....	29
3.4 Robotlarda Bilgisayarlı Görme.....	29
4. MATERYAL VE METOT	31
4.1 Çalışma Ortamının Hazırlanması.....	33
4.2 Sistem Tasarımı	33
4.3 Eşik Değeri ile Filtreleme ve Engelden Sakınım	35
4.4 Uygulama.....	40
5. TARTIŞMA VE SONUÇ.....	43
KAYNAKLAR.....	45
ÖZGEÇMİŞ	51

KISALTMALAR

AOYD	: Aritmetik Ortalamanın Yüzdesel Değişimi
API	: Uygulama Programlama Arayüzü (Application Programming
BSD	: Berkeley Yazılım Dağıtımı (Berkeley Software Distribution)
CV	: Bilgisayarlı Görme (Computer Vision)
DDS	: Veri Dağıtım Hizmeti (Data Distribution Service)
FPS	: Saniyedeki Çerçeve Sayısı (Frames Per Second)
GPS	: Küresel Konumlama Sistemi (Global Positioning System) Interface)
KNN	: K-En Yakın Komşuluk (K-Nearest Neighbor)
OpenCV	: Açık Kaynak Bilgisayarlı Görme (Open Source Computer Vision)
OSRF	: Açık Kaynak Robotik Vakfı (Open Source Robotic Foundation)
PCA	: Temel Bileşenler Analizi (Principal Component Analysis) Protocol/Internet Protocol)
RGB	: Kırmızı Yeşil Mavi (Red Green Blue)
ROI	: İlgi Bölgesi (Region of Interest)
ROS	: Robot İşletim Sistemi (Robot Operating System)
RPC	: Uzaktan Yordam Çağırma (Remote Procedure Call)
SDF	: Standart Tanım Formatı (Standart Definition Format)
SVM	: Destek Vektör Makinesi (Support Vector Machine)
TCP/IP	: İletim Denetimi Protokolü / Internet Protokolü (Transmission Control
UDP	: Kullanıcı Veri Bloğu Protokolü (User Datagram Protocol)
URI	: Tekdüzen Kaynak Tanımlayıcısı (Uniform Resource Identifier)
XML	: Genişletilebilir İşaretleme Dili (Extensible Markup Language)

ÇİZELGE LİSTESİ

	Sayfa No
Çizelge 2.1: ROS düğüm komutları	11
Çizelge 2.2: ROS konu komutları.....	12
Çizelge 2.3: ROS mesaj komutları	13
Çizelge 2.4: ROS servis komutları	13
Çizelge 2.5: rviz komutları	14
Çizelge 2.6: rqt komutları	15
Çizelge 2.7: rosbag komutları	15
Çizelge 2.8: ROS komut seti	16
Çizelge 4.1: Varyans, aritmetik ortalama ve standart sapma için değerler.....	36
Çizelge 4.2: Benzetim Ortamı Parametreleri.....	41

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Güncel ROS mimarisi	9
Şekil 2.2: ROS dosya sistemi	9
Şekil 2.3: ROS sunucu mimarisi	10
Şekil 2.4: ROS düğüm konu düğüm ilişkisi	13
Şekil 2.5: ROS-Gazebo platformu genel mimarisi	17
Şekil 2.6: Willow Garage tarafından geliştirilen Turtlebot robot kiti	19
Şekil 3.1: Genel görüntü işleme akışı	21
Şekil 3.2: Sayısallaştırılmış bir görüntüdeki pikseller	22
Şekil 3.3: Görüntünün ikili görüntü olarak sayısal değeri	22
Şekil 3.4: İkili (sayısal), gri ton ve renkli görüntü örnekleri	23
Şekil 3.5: Görüntü bölütleme yaklaşımları	25
Şekil 3.6: Bölge tabanlı görüntü bölütleme örnekleri	25
Şekil 3.7: Kenar bulma algoritmaları ile işlenmiş örnek görüntüler	26
Şekil 3.8: Gri ton görüntüye uygulanan eşik değeri ve etkisi	27
Şekil 4.1: Robotun görüntü alması ve hareketi	31
Şekil 4.2: Robotun görüntü olarak işlenmesi ve karar süreci	32
Şekil 4.3: Uygulama üzerindeki ROS düğüm yayın ve abonelikleri	34
Şekil 4.4: Bölütlenmiş benzetim ortamı görüntüsünden bir kesit (64×48)	35
Şekil 4.5: Bölütlerde renkler	36
Şekil 4.6: Filtre matrisi	37
Şekil 4.7: Filtrelenmiş matris üzerinde engel kontrolü	38
Şekil 4.8: Benzetim görüntüsü, filtre matrisi ve yön matrisi	39
Şekil 4.9: Yön matrisi ve sağ-sol kenar kontrolü	39
Şekil 4.10: Sağ-sol hareket akış şeması	40
Şekil 4.11: Benzetim ortamı	40
Şekil 4.12: Benzetim ortamı robot kamerası görüntüleri (I)	41
Şekil 4.13: Benzetim ortamı ve robot kamerası görüntüleri (II)	42
Şekil 4.14: Benzetim ortamı ve robot kamerası görüntüleri (III)	42

GEZGİN ROBOTUN GÖRME TABANLI VE GERÇEK ZAMANLI OLARAK ENGELDEN SAKINIMI

ÖZET

Gezgin robotların hareketinde, engelden sakınım, üzerinde çalışılan en önemli araştırma konularından biridir. Otonom robotların hareket ederken engelden sakınabilmeleri için oldukça karmaşık algoritmalar ve işlemler gerekmektedir. Bu konuda robot üzerindeki sensör verileri kullanılarak çeşitli haritalama yöntemleri önerilmiş ve engelden sakınarak gezinme için yaklaşımlarda bulunulmuştur.

Tez çalışmasının amacı herhangi bir harita bilgisine ve sensör verisine ihtiyaç duymadan bilgisayarlı görme yaklaşımı ile engellere çarpmadan hareket edebilen otonom gezgin bir robot benzetimi gerçekleştirmektir. Çalışmada gezgin robotun engellerden sakınarak gezinmesi için iki aşamalı bir süreç takip edilmektedir. Birinci aşamada robot kamerasından alınan görüntüler işlenerek görüntüde zemin ayrımı yapılmaktadır. İkinci aşamada ise, zemin olarak işaretlenen alanlar kontrol edilmektedir. İşaretlenen alan robotun ilerlemesi için uygun ise ileri yönlü hareket yapılmaktadır. Harekete uygun değilse, yani herhangi bir engel tespit edilmiş ise, mevcut görüntü tekrar işlenerek sağa ya da sola doğru dönme hareketi yaptırılmaktadır. Tüm bu işlemler yinelemeli olarak devam etmekte ve bir sonraki görüntüde benzer durum varsa dönmeye devam edilmekte, yoksa ileri yönlü hareket başlatılmaktadır. İki aşamadan oluşan bu sistem gerçek zamanlı olarak çalışmaktadır. Bunun en büyük avantajı ise dinamik ortamlar için de sistemin kullanışlı olmasıdır.

Çalışmada robotlar için tasarlanan ve açık kaynak yazılımlardan oluşan ROS işletim sistemi kullanılmıştır. Benzetim ortamı olarak ise ROS ile entegre olabilen Gazebo seçilmiştir. Çalışmada sunulan yöntem, TurtleBot robotu üzerinde test edilmiştir. Bilgisayarlı görme çalışması için kullanılan kamera, TurtleBot üzerinde bulunan 640×480 piksel ve renkli görüntü alabilen web kameradır.

Gezgin robotlar için tasarlanan bu modelde, Turtlebot robotu, herhangi bir çarpışma olmadan hareket edebilmektedir. ROS-Gazebo benzetim ortamından elde edilen sonuçlar, önerilen yöntemin engellerden kaçınmak için düşük maliyetli ve esnek bir yöntem olduğunu göstermektedir.

Anahtar Kelimeler: Gezgin Robot; Zemin Tespiti; Görüntü Bölütleme; Engelden Sakınım

VISION BASED AND REAL TIME OBSTACLE AVOIDANCE OF MOBILE ROBOT

ABSTRACT

In the navigation of mobile robots, target recognition and obstacle avoidance are one of the most important research topics studied. Very complex algorithms and operations are required for autonomous robots to recognize targets and move towards a specific target. In this regard, various mapping methods, sensor data and various approaches have been adopted and designs have been made.

The aim of this study is to realize an autonomous robot simulation that can move without hitting obstacles with computerized vision without using a map and using any sensor data. In the study, a two-step process is followed for the traveling robot to navigate avoiding obstacles. In the first stage, the images taken from the robot camera are processed and ground separation is made in the image. In the second stage, the area marked as the floor is checked. If it is suitable for the robot to move forward, forward motion is made. If it is not suitable for movement, that is, if any obstacle is detected, the existing image is reprocessed, making only a rotational movement to the right or left. Since all these processes continue recursively if there is a similar situation in the next image, it continues to rotate or forward motion is started. The system, consisting of two stages, operates in real-time. The biggest advantage of this is that the system is useful in dynamic environments. The environment is evaluated for each image taken on the webcam, and the robot can decide on movements such as rotation or progression.

In this study, the ROS operating system designed for robots and consisting of open sources is used. Gazebo, which can be integrated with ROS, was chosen as the simulation medium. Algorithms created by the researcher were tested on the TurtleBot robot. The webcam used for the computer vision model is a webcam defined for TurtleBot.

In this model designed for mobile robots, the model robot can move without any collision. Results from the ROS-Gazebo simulation environment show that the proposed method is a low cost and flexible method to avoid obstacles.

Keywords: Mobile Robot; Floor Detection; Image Segmentation; Obstacle Avoidance

1. GİRİŞ

Robotlar uzun zamandır tıp [1], tarım [2,3], uzay [4], savunma [5], acil durum [6], ulaşım [7] gibi birçok alanda kullanılmakta ve halen üzerine yapılan çalışmalar devam etmektedir. Günümüzde havalimanlarında, hastanelerde hatta evlerde, insan yaşamını kolaylaştıran robotlar geliştirilmektedir. Gezgin robotlar ise tehlike arz eden işlerde, gezegen araştırmalarında, yük taşımada ve özel yaşamda kullanılmak üzere geliştirilmekte olan robotlardır. Gezgin robotların dinamik ortamlarda hareketinde eş zamanlı çalışma ile maliyet kritik unsurlar arasındadır. Bu çalışmada maliyeti düşük ve eş zamanlı çalışabilecek bir robot modeline odaklanılmıştır.

Robotlar üzerinde; kızılötesi, ultrasonik, lazer, küresel konumlama sistemi (GPS) gibi çeşitli sensörler barındırır [8]. Bu sensörlerin birbirlerine göre üstünlükleri ve zayıflıkları bulunmaktadır [9,10]. Kızılötesi ve sonar gibi sensörler nispeten daha az maliyetli olmasına karşın, çözünürlükleri yeterli değildir ve farklı ortamlardan kolayca etkilenebilir. Lazer ise iyi bir çözünürlük ve mesafe ölçümü vermesine karşılık oldukça pahalıdır. GPS belirli bir çözünürlüğe kadar konum bilgisi verse dahi, engel bilgisi veremez ve uygun sayıda uydunun görüş alanında değilse, iletişim verisi sağlayamaz. Bu sensörlerin tamamı etkili bir gezinme için yeterli değildir. Robotun bu sensörlerle birlikte bir hedefe ulaşması için ayrıca bir kamera kullanması gerekmektedir. Bilgisayarlı görmenin önemi burada ortaya çıkmaktadır. Bilgisayarlı görme ile kameradan alınan görüntüler ile dinamik bir ortam tanımlanabilir ve aynı görüntüler farklı teknolojilerle işlenerek hedef tanıma, hedef sayma gibi yan görevler icra edilebilir.

Gezgin robot, üzerindeki sensörler aracılığıyla elde ettiği bilgileri işleyebilen ve bu bilgiler doğrultusunda hareket edebilen, verilen görevleri yerine getirebilen bir sistemdir. Çalışmada, kapalı ortamlarda robotun engellerden kaçınarak hareket edebilmesi için zemin algılamaya dönük yeni bir yaklaşım sunulmuştur. Gezinmeye temel oluşturan unsur, bilgisayarlı görme modelidir. Bunun sebebi, görme temelli yaklaşımların, diğer yaklaşımlara göre daha az maliyetli olması ve sürekli geliştirilmesidir [11,12]. Gezgin robotun sahip olması gereken işlevler arasında çevreyi tanımlayabilme, hedef tanıyabilme ve hedefe engellerden kaçınarak, kendine zarar vermeden ulaşabilme kabiliyeti bulunmaktadır. Çalışmanın odak noktası, gezgin robotun daha önceden çevre hakkında bilgi edinmeden ve harita kullanmadan,

tamamen bilgisayarlı görme modeli ile hareket etmesidir. Gezgin robotlar global ve lokal olarak seyir yapabilir. Çalışmada lokal seyir türü kullanılmıştır. Robot web kamerası aracılığı ile görüntü elde edebilir ve çarpışmalardan kaçınarak hareketi gerçekleştirebilir.

Genel olarak bilgisayarlı görme üzerine yapılan çalışmalarda görüntü işleme teknikleri kullanılmaktadır. Görüntü işleme tekniklerinde, görüntü üzerinde nesnelerin algılanması, zeminin algılanmasına göre daha kolaydır. Nesnelerin özelliklerinin filtreler yardımıyla çıkarılması, zemine göre daha olasıdır. Çünkü nesnelerin farklı nesnelerle ya da fonla farkı, renk geçiş filtreleri ile anlaşılabilir. Ancak zemin için bu kolay değildir. Robotun ilerlemesi için öncelikli bilgi zemin bilgisidir. Bir alanın zemin olup olmadığını anlamak için görüntünün tamamının incelenmesi gerekir. Bu işlem için görüntü ayrıştırma tekniklerinden faydalanılır. Bu çalışmada görüntü üzerinden zemin algılanmasına yönelik eş zamanlı çalışan bir yöntem sunulmuştur.

Uygulanan yöntem, benzetim ortamı oluşturularak gerçekleştirilmiştir. Bunun sebebi gerçek bir robot ile deneme yapmanın pahalı ve zaman alıcı olmasıdır. Deneysel çalışmalarda, robotun beklenen hareketleri gerçekleştirememesi gibi olası durumlarda zarar görmesi ihtimali artmaktadır. Bu sebeple çalışmada Gazebo benzetim ortamı kullanılmıştır. Gerçek ortamlarda kullanılabilen TurtleBot robotu benzetim ortamında modellenerek deneysel çalışmalar gerçekleştirilmiştir. Robot işletim sistemi (ROS) da bir işletim sistemi platformudur ve gerçek robotlar üzerinde ve endüstriyel ortamlarda kullanılabilmektedir. ROS robotlara yüklenebilir ve Gazebo ile kolayca entegre olabilir. Çalışmada ROS-Gazebo ortamı oluşturulmuş ve görüntü işleme algoritmaları, geliştirilen modele yüklenerek test edilmiştir.

Tez çalışması 5 bölümden oluşmaktadır. Birinci bölümde çalışmanın amacı, literatür araştırması ve görüntü işleme üzerine gerçekleştirilen çalışmalardan bahsedilmiştir. İkinci bölümde kullanılan ROS işletim sistemi ve benzetim ortamı olarak kullanılan Gazebo anlatılmıştır. Üçüncü bölümde bilgisayarlı görme ve görüntü işleme teknikleri anlatılmıştır. Dördüncü bölümde önerilen yöntemlerin gerçekleştirilme ve test adımlarından bahsedilmiştir. Beşinci bölümde ise sonuç ve öneriler sunulmuştur.

1.1 Tezin Amacı ve Katkısı

Tez çalışmasında kapalı ve dinamik ortamlarda, robotun sadece kameradan alınan görüntülerle zemini algılaması ve engellerden kaçınması hedeflenmiştir. Kameradan

alınan görüntülerin işlenerek zeminin algılanması ve engellerden sakınılması için eşzamanlı çalışabilecek bir yöntem geliştirilmiştir. Geliştirilen yöntem, gerçek robotlarda kullanılabilen ROS ortamında test edilmiş ve benzetim ortamında etkileri incelenmiştir. Çalışmada, önceden eğitim gerektirmeyen, görüntü bölütlemeye dayalı yöntemler kullanılarak, gerçek zamanlı uygulanabilir bir model oluşturulmuş ve robot gezinmesine daha esnek bir yaklaşım amaçlanmıştır. Robotlar konusunda yapılan çalışmalar görüldüğünden daha karmaşıktır. Bu çalışmanın bundan sonraki robotik çalışmalarda araştırmacılar için alternatif bir kaynak olabileceği düşünülmektedir.

Çalışmanın literatüre ana katkıları aşağıdaki gibi özetlenmiştir:

- Tek bir kamera dışında başka ek bir donanıma gerek duymadan zemin tespitini başarılı bir şekilde gerçekleştirebilen etkili bir yöntem önerilmiştir.
- Gezgin robot, bilinen ve bilinmeyen bir ortamdaki engellerden kaçınarak gezinme yeteneğine sahiptir.
- Engeller ayrı ayrı algılanmadığı için ek filtrelere (örn. Canny, Sobel ve Laplacian) gerek kalmadan engellerden kaçınma, hızlı ve kolay bir şekilde sağlanabilmektedir.
- Deneysel çalışmalar ROS-Gazebo ortamında test edilerek, önerilen yöntemin gerçek dünyada uygulanabilir bir model olduğu görülmüştür.

1.2 Literatür Araştırması

Engelleri tespit etmek ve bu engellerden kaçınmak, robot gezinmesinde dikkate alınması gereken en önemli ve zorlu konular arasındadır. Engellerden kaçınma problemini çözmek için literatürde çeşitli yöntemler önerilmiştir. Bazı yaklaşımlar arasında lazer sensörleri [13], ultrasonik sensörler [14] yer almaktadır. Engellerden sakınmak için sensörlere ek olarak, görme tabanlı yöntemler de kullanılmıştır. Monoküler [15,16] veya stereo [17] kameralar kullanılarak görme tabanlı yaklaşımların olduğu çalışmalar mevcuttur. Görme tabanlı sistemler, çevre hakkında geniş bir perspektifle önemli bilgiler sağlamaktadır [18]. Kameralarla alınan görüntüler, parlaklık, renk ve doku dahil olmak üzere çevre hakkında bol miktarda bilgi sağlayabilir. Bu sayede otonom bir gezgin robot kamera kullanarak, önceden bilgisinin olmadığı, bilinmeyen bir ortamı haritalayabilir ve ortam hakkında gerekli olan bilgileri elde edebilir.

Görme tabanlı sistemlerde, görüntüler genellikle gerçek zamanlı olarak işlenir. Kameradan alınan görüntüler, ışık yoğunluğu, nesnelerin parlaklığı ve kameranın açısı gibi görüntü kalitesinde bozulmaya neden olan faktörler yüzünden çeşitli gürültüler içerebilir. Görüntü işleme sırasında, bu olumsuz etkileri ortadan kaldırmak ve başarılı sonuçlar elde etmek için bölütleme ve eşikleme teknikleri uygulanmaktadır [19]. Görüntü işlemenin en temel sorunlarından biri olan görüntü bölütleme, temel olarak sayısal bir görüntüyü piksellerin özelliklerine göre çeşitli bölümlere ayırmaktır [20,21].

Literatürdeki çalışmaların bir kısmı zemin tespitine odaklanmıştır. Engelleri tespit etmek yerine, robotun serbestçe hareket edebileceği alanlar, kamera görüntüleri işlenerek elde edilir. Zemin görüntülerini, çarpışma olmadan gezinmek için işlemek, özellikle iç mekânlarda kullanılan robotlar için pratik ve etkilidir. Zemin algılama yöntemi, robot üzerinde kamera tarafından elde edilen görüntüleri işleyerek, zeminin ve engellerin düşük bir maliyetle ayrılmasını sağlar. Konum ve boyutların tespiti gibi engellerin özelliklerini belirlemek için ek işlem maliyeti gerektirmez. Bu çalışmada, monoküler görmeye dayalı, engellerden etkili bir şekilde kaçınmak için zemin tabanlı yeni bir algılama yaklaşımı önerilmiştir. Bu çalışmadaki motivasyon, alınan görüntülerdeki zemini görüntünün geri kalanından ayırt etmek ve robotun serbestçe hareket edebileceği alanları belirleyerek, eş zamanlı olarak görüntüler üzerinde basit ve hızlı filtreleme yaparak etkili bir teknik sunmaktır. Zemin tabanlı algılamanın en büyük dezavantajı, kapı, duvar gibi zemin görüntüsüne yakın nesnelerin algılanamamasıdır. Deneysel çalışmalar önerilen yaklaşımın bu problemi etkili bir şekilde ortadan kaldırdığını göstermiştir.

Kameradan alınan görüntülerde renkler birbirinden önemli ölçüde farklı olduğundan standart bölütleme yöntemleri çok iyi sonuçlar elde edebilmektedir [22]. Engeller belirli kenar özelliklerine sahip olmakla birlikte, zemin olarak tanımlanan alanlar değişkendir ve zeminin özelliklerini görüntülerden elde etmek çok zordur. Bu çalışmada zemin alanlarını belirlemek için görüntülere istatistiksel bir yöntem uygulanmış, eşikleme için ise görüntü bölütleme tekniği kullanılmıştır. Eşikleme tespit sürecinde büyük önem taşımaktadır [23] ve en yaygın ve basit yöntemlerden [43] biridir. Her görüntüdeki negatif faktörlerin (ışık, gölge vb.) ve renklerin etkileri değişeceğinden, bölütlere ayırma, uyarlanabilir eşik değeri ile gerçekleştirilmiştir. Bölütler arasındaki karşılaştırmalar, AOYD (Aritmetik Ortalamanın Yüzdesel

Değişimi)’ye göre yapılmıştır. Önerilen yöntem statik ve dinamik ortamlarda etkili bir şekilde kullanılabilir.

Engel algılama ve kaçınma, gezgin robot navigasyon çalışmalarında kilit rol oynamaktadır. Bu problemi çözmek için, özellikle görme tabanlı pek çok çalışma gerçekleştirilmiştir.

Li ve arkadaşları [24], engelden kaçınma problemi için, görüntü işlemeye dayalı önerilerde bulunmuştur. Hareket yolunu belirlemek ve engellerden kaçınmak için dinamik pencere yaklaşımı (DWA) ve yapay potansiyel alan algoritmasını birleştirmişlerdir. Robotun oyun alanına rastgele yerleştirilen sekiz adet engelden kaçınması sağlanmıştır.

Kalogeiton ve arkadaşları [25], stereo kamera kullanarak hem engelleri hem de boş alanları haritalamak için bir çalışma gerçekleştirmişlerdir. Ortamın haritalandırılmasından sonra, bir sonraki optimum hareket için bilişsel tabanlı uyarlamalı optimizasyon (CAO) yaklaşımını tercih etmişlerdir.

Benn ve Lauria [26], görüntü renk bölütleme ve kenar algılamanın bir kombinasyonunu kullanarak dinamik ortamlarda kullanılabilen iki aşamalı bir yöntem önermişlerdir. Çalışmanın amacı zemini ve engelleri birbirinden ayırmaya yöneliktir. İlk aşamada renkli görüntüler için bölütleme uygulamışlar, ikinci aşamada ise Canny filtresini kullanmışlardır. Ayrıca Hough algoritması ile zemin kenarları netleştirilmiştir. Bununla birlikte, çalışmada piksel tabanlı değerlendirme ve kenar filtreleri uygulandığından, zemine benzer yüzeylerin ayırt edilmesi zorlaşmaktadır.

Bhowmick ve arkadaşları [27], zemin tespiti için aydınlatma etkisini azaltacak bir yöntem önermişlerdir. Görüntüye ait YCbCr renk uzayındaki histogram özellikleri kullanılarak zemin tespiti yapılmıştır. Çalışma, ışık kaynaklarının konumundan dolayı farklı sonuçlar verebilmektedir.

Chun ve arkadaşları [28] çalışmalarında doğrusal olmayan difüzyon ve görüntü bölütleme teknikleri kullanarak, derinlik bilgisi tahminine yönelik bir yöntem sunmuşlardır. Üç aşamadan oluşan yöntemde, ilk aşamada doğrusal olmayan difüzyon tekniği ile zemin aday bölgelerinin belirlenmesi, ikinci aşamada bölütleme teknikleri ile zemin alanlarının tespiti ve üçüncü aşamada ise derinlik tahmini bulunmaktadır.

Li ve Birchfield [29], kalibre edilmemiş bir kamera kullanarak görüntü tabanlı bir zemin algılama algoritması önermişlerdir. Önerdikleri yöntemde, Canny filtresi kullanılarak görüntü üzerinde yatay çizgiler belirlenmekte ve daha sonra görüntü bölütlemesi yapılarak yatay çizgilerin zemin/duvar çizgisi olup olmadığına yönelik tahmin yapılmaktadır.

Ling ve arkadaşları [30], tavana sabitlenmiş bir kamera ile iç mekân robot gezinmesinde görme tabanlı zemin bölütleme yöntemi önermişlerdir. K-Ortalama ve Geliştirilmiş Temel Bileşenler Analizi (PCA) kombinasyonu birlikte kullanılmıştır. Ancak çalışmalarına bakıldığında, kamera tavanda olduğu için robotun altından geçebileceği nesnelerin de zemin olarak algılanabileceği söylenebilir.

Benabid ve arkadaşları [31], otonom gezgin robotlar için gerçek zamanlı ve görüntü işlemeye dayalı bir sistem önermişlerdir. 640×480 boyutlarında ve 25 fps (Saniyedeki Çerçeve Sayısı) akış hızında görüntüler işlenerek engelden sakınım algoritması gerçekleştirilmiştir.

Mane ve Vhanale [32], kinect kamera kullanarak beş aşamalı gerçek zamanlı engelden tespit ve sakınma algoritması önermişlerdir. Kinect kamera ile alınan görüntüler işlenerek derinlik haritası oluşturulmuş ve gerçek dünya ile eşleştirilmiştir. İç mekânda test edilen sistem 640×480 piksel boyutundaki görüntüleri ilgi bölgesi (ROI) tekniği ile işleyerek engelleri algılamaktadır.

2. ROBOT İŞLETİM SİSTEMİ (ROS)

ROS, Stanford Üniversitesi akademisyenlerinin robotik ürünler geliştirilmesinde en büyük engel olan disiplinler arası uyumsuzluk sorununa çözüm olarak, özel bir şirketin desteğini de alarak ortaya çıkardığı bir yazılımdır [33]. 2007 yılında ilk sürümünü çıkarmıştır ve açık kaynaklı bir yazılımdır. Bu çalışmada kararlı sürüm olan Kinect sürümü kullanılmıştır. ROS, süreçler arasında iletişim katmanı oluşturan bir soyutlama yazılımıdır. Bu yazılım donanım soyutlaması, aygıt sürücülerini, birden çok makine üzerindeki işlemler arasında iletişim, test ve görselleştirme araçları sunar. Her biri özel işlevsellik sağlayabilen 2000'in üzerinde paketten oluşur. Yazılımın temel özelliği, çalışma şekli ve iletişim biçimidir. Bununla birlikte belirli donanımların nasıl çalıştığını bilmeden karmaşık yazılımlar yazma imkânı tanır. Birden fazla cihazda çalıştırılabilir ve cihazlar birbiri ile haberleşebilir. Bir ağ oluşturmak mümkündür. Farklı alternatifleri olmasına rağmen (MRS, Player, YARP, MRPT, Microsoft RDS, LCM, CARMEN, MOOS, ORCA, OROCOS) daha dinamik ve geniş bir geliştirici ve kullanıcı topluluğuna sahip ve benzetim ortamı (Gazebo) ile rahat entegre olması bu çalışmada kullanılmasında etkili olmuştur. ROS'un farklı sistemlerle haberleşmesi, en çok tercih edilen sistemlerden biri haline gelmesinin nedenidir. Linux sürümleri üzerinde rahatlıkla çalışmaktadır. ROS bu özelliklerinden dolayı, akademik araştırmalar başta olmak üzere eğitim uygulamalarında, yaygın robot sorunlarının çözümünde kullanılmaktadır.

Robot yazılımlarında, basit görünen işlerin arka planında oldukça karmaşık yazılımlar mevcuttur. Hem donanım hem yazılım katmanında, tasarım aşaması da dahil olmak üzere önemli ölçüde zaman harcanır. ROS bu noktada geliştiricilere hazır donanım ve algoritmaların uygulanabileceği yazılım katmanları sunar. Böylece her geliştirici tekrar tekrar kod yazmak yerine bir ölçüde mevcut kodları kullanarak, kendi uygulamasını geliştirebilir. Adındaki "işletim sistemi" kavramı buradan gelmektedir. Temel amacı gerçek bir işletim sistemi gibi, donanım soyutlaması yapmak ve standart işlevler sunmaktadır. Bellek ve merkezi işlem birimi yönetimi, donanım sürücülerinin yönetimi, eşzamanlılık, veri birleştirme gibi özelliklere sahiptir. Google ve Yahoo gibi arama motoru teknolojilerinin geliştirilmesine dahil olan ve Scott Hassan tarafından kurulan Willow Garage şirketi tarafından desteklenmektedir [33]. Willow Garage tamamen kişisel robotlara dönük araştırma ve geliştirme yapan bir şirket olarak kendini tanımlamaktadır.

ROS eşler arası iletişim sağlar (peer to peer). Aynı zamanda çok dillidir. XML-RPC (Genişletilebilir İşaret Dili-Uzaktan Yordam Çağırma) teknolojileri ile iletişim kurduğu için, dilden bağımsız programlama yapılabilir. Asenkron ya da senkron mesaj iletimine açıktır. ROS, Gazebo, OpenCV gibi farklı kütüphaneleri kullanabilir. ROS üzerindeki yazılımlar Berkeley Yazılım Dağıtımı (BSD) lisansı ile lisanslanmaktadır.

2.1 ROS Sürümleri

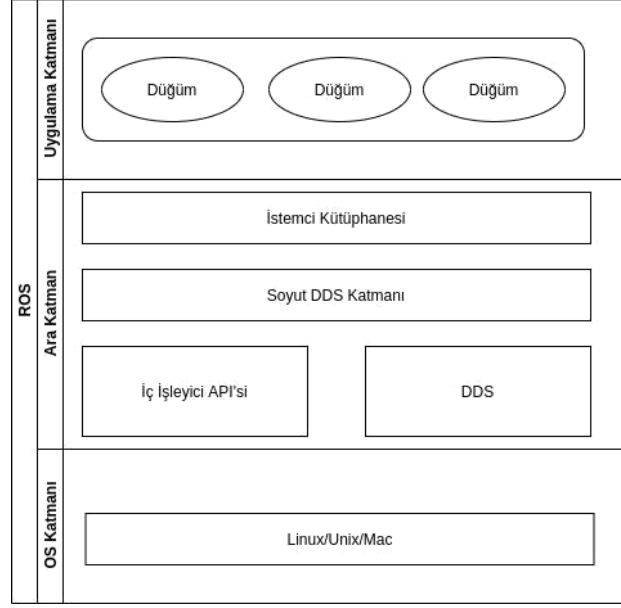
ROS dağıtımları, dağıtım özel paketlerden oluşur. Bu yönüyle Linux dağıtımlarına benzer. Dağıtımların amacı, ROS paket geliştirmelerinin sürmekte iken kullanıcıların stabil bir ortamda geliştirmelerine devam edebilmesi içindir. Ortalama her yıl bir ROS dağıtımı duyurulmaktadır. İlk dağıtım 2010 yılında ROS Box Turtle iken, tez çalışmasında kullanılan dağıtım 2016 yılında duyurulan ROS Kinect sürümüdür.

ROS 2 kavramı ise genel ROS mimarisindeki değişiklikten sonra eski mimari ile farklılığı ortaya koymak için kullanılır. 2018 yılı sonrası sürümler ROS 2 olarak geçer. ROS ve ROS 2 arasında mimari açıdan farklar vardır, gelişen ihtiyaçlara göre değişiklikler ve eklemeler yapılmıştır. Düğümlerin birbiri ile haberleşmesi için DDS katmanı üzerine soyutlama katmanı eklenmiştir ve en belirgin farkta budur.

2.2 ROS Mimarisi

ROS Linux/Unix/MAC türevi işletim sistemlerine rahatlıkla kurulabilir. Raspberry Pi ve benzeri donanımlar için uygulanabilir. Kurulduğu ortama bağlı donanımlarla yazılan uygulamaların haberleşmesi mümkündür. Şekil 2.1’de görüldüğü gibi, işletim sistemi üzerinde iki katman daha vardır. Ara katman olarak belirtilen katman uygulama ile işletim sistemi arasındaki trafiği yönetir. Birden fazla düğüm olabilir.

ROS istemcileri arasındaki iletişimi sağlamak için DDS katmanı kullanılır. DDS dağıtık ortamlarda gerçek zamanlı çalışmalar için tasarlanmış bir veri iletim biçimidir. Kullanıcı Veri Bloğu Protokolü (UDP) tabanlı bir protokol olduğundan gecikmesi mikro saniye mertebesinde. ROS verisi DDS üzerinden yayınlanır. Fakat verinin kimlere iletileceğini belirlemez. Kimler veriye erişmek istiyorsa abone olabilir. Bu uygulama ile uygulama istemcileri birbirinden bağımsız hale getirilir. Böyle bir istemcinin susması durumunda diğer istemcileri etkileyecek bir durum oluşmaz.

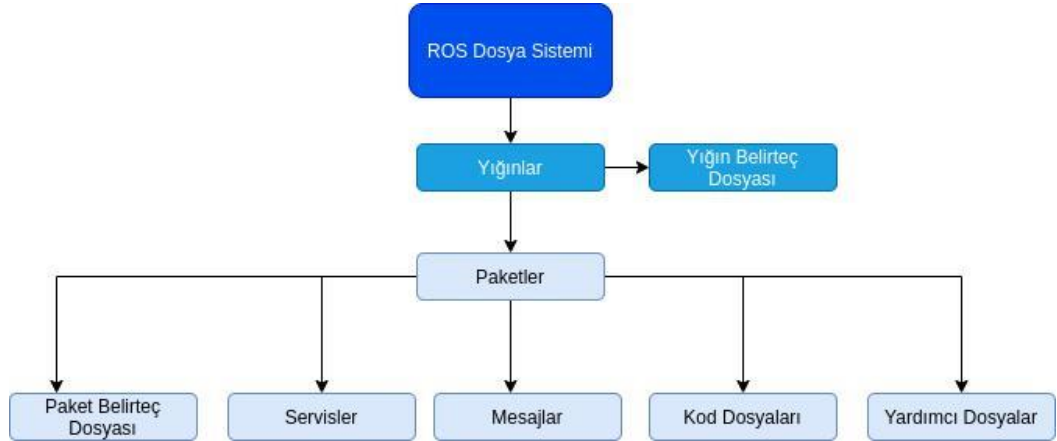


Şekil 2.1: Güncel ROS mimarisi

Farklı işletim sistemleri ile entegre olabilen güncel ROS mimarisi şekil 2.1’de sunulmuştur.

2.3 ROS Dosya Sistemi

ROS kaynakları disk üzerinde hiyerarşik bir yapıda düzenlenmiştir. Bu hiyerarşiyle ilgili iki önemli kavram vardır, bunlar paketler ve yığınlardır. Paketler ROS sistemindeki en temel birimdir. Bir paket, paket gereksinimlerini belirten manifesto dosyası, harici kütüphaneler, veriler, yapılandırma dosyaları ve kod dosyalarını içerir. Bir paketin içinde paket olamaz. Paketler birleşerek yığınları oluşturur. Yığın kök klasöründe yığına ait manifesto dosyası bulunur. Bu şekilde ortaya çıkan ROS dosya sistemi, versiyon ismi verilerek dağıtım halini alır.

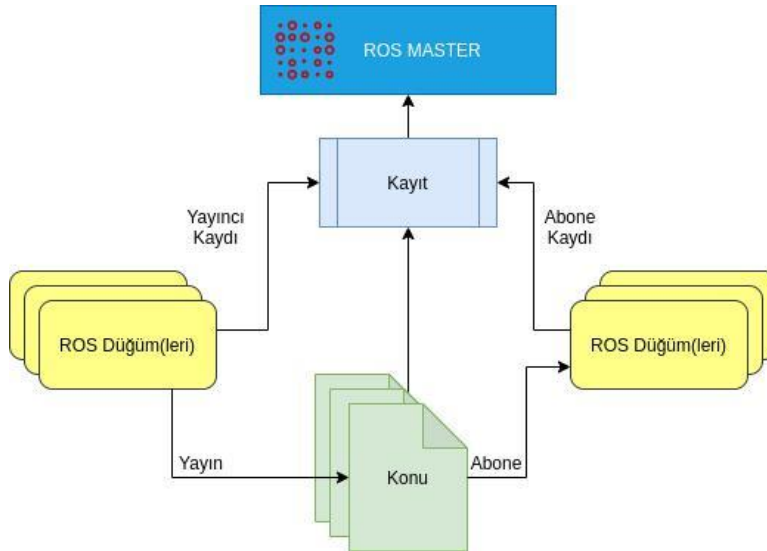


Şekil 2.2: ROS dosya sistemi

ROS sistemindeki temel amaç, veri alışverişi yapması gereken çok sayıda dosyayı senkron ya da asenkron olarak haberleştirebilmektir. Bir robot; basınç, mesafe, sıcaklık, ısı gibi sensörler ile kamera GPS, mikrofon, jiroskop, ivmeölçer gibi çok sayıda donanımdan oluşur. Tüm bu donanımlardan üretilen verinin işlenmesi, ilgili ünitelere aktarılması sürekli ve eşzamanlı olur. ROS tüm bu iletişimi ve kaynakları şekil 2.2'deki dosya mimarisi ile yönetir ve düzenler.

2.4 ROS Kavramları

ROS, robotlar ve robotların sahip olduğu donanımlar için bir istemci/sunucu sistemi sunar. ROS sistemi; düğümleri, konuları, hizmetleri ve parametreleri içeren paketlerden oluşur. Düğümler yürütülebilir programlardır. Genel ROS iletişim konsepti şekil 2.3'deki gibidir.



1. Şekil 2.3: ROS sunucu mimarisi

2.4.1 ROS Master

ROS tek bir bilgisayarda ya da bilgisayarlar arasında çalışarak hizmet verebilir. ROS modülleri düğümlerden oluşur ve birbiri ile iletişim kurar. Düğümler birer isim ile çağrılır ve arkasında yürütülebilir programlar bulunur. Her bir düğüm sistemin bir yönünü kontrol eder. Düğümler farklı donanımlar (bilgisayarlar, robotlar) üzerinde bulunabilir. ROS Master bu dağıtık yapının merkezinde bulunur ve tüm ortamı dinleyerek kontrolünü sağlar.

ROS Master bu hizmeti yani düğümlerin birbiri ile iletişim kurması ve veri alışverişinde bulunması için bildirim ve kayıt hizmeti sunar. Kayıt hizmeti XML ve

RPC protokolleri ile çalışır. Düğümlerin, konuların, servislerin birbiri ile eşleşmesini sağlar. ROS iletim katmanı, TCPROS olarak adlandırılmıştır ve iletişim protokolü olarak TCP/IP (Transmission Control Protocol/Internet Protocol) kullanır.

ROS Master “Parameter Server” adında bir bileşen daha içerir. Parameter Server, merkezileştirilmiş bir veri tabanı gibi düşünülebilir. Parameter Server üzerinde düğümler veri saklayabilir ya da sistem parametrelerine ulaşabilir. “roscore” komutu ile ROS sunucusu başlatılır. ROS sunucusu her başlatıldığında, bir ip adresi ve port seçilerek sunucu bağlantı noktası olarak ayarlanır. Ön tanımlı olarak yerel ip adresi (localhost) ve 11311 numaralı port URI adresi olarak kullanılır.

2.4.2 ROS Düğüm

ROS’un aslında en temel kavramı düğümdür. Tüm mimari düğüm üzerine bina edilmiştir. Birbiri ile haberleşen unsurlar için kullanılan kavramdır. Veri üreten, içeren ya da işleyen her unsur düğüm olabilir. Sensörlerden üretilen veriler için bir düğüm oluşturulabileceği gibi, sensör verileri üzerinden işlem yapan bir algoritmaya sahip dosya ile de düğüm oluşturulabilir. Oluşturulan her bir düğüm birbirinden bağımsız olarak ROS Master sunucusu üzerine kaydedilir. Birbirinden bağımsız çalışan düğümlerinin birinde sorun oluştuğunda diğer düğümler çalışmaya devam eder. Düğümler eşsiz isimlere sahiptir. Aynı isimde iki düğüm bulunursa ve bunlardan biri çağrılırsa, ROS ilk bulduğu düğümü çalıştırır. Bu durum sistemin hatalara karşı daha fazla toleranslı olmasını sağlar.

Çizelge 2.1: ROS düğüm komutları

Komut	Açıklama
roscore -h	Yardım dosyasını görüntüler
roscore list	Tüm düğümleri listeler
roscore info dugum_adi	İlgili düğüme ilişkin bilgileri gösterir
roscore ping dugum_adi	Düğüme bağlantı testi gerçekleştirir
roscore kill dugum_adi	İlgili düğümü sonlandırır
roscore cleanup	Erişilemeyen düğüm bilgilerini temizler

2.4.3 ROS Konu

Veri taşıma için kullanılan sistemdir. Bir veri akışı yayınlayabilir ya da bir veri akışına üye olunabilir. Veri akışı için bir başlık belirlenir, böylece farklı unsurlar bu veri akışına üye olabilir. Birden fazla düğüm tek bir konu için üyelik talebinde bulunabilir, ya da konu için veri gönderebilir. Bir düğümün birden fazla aynı isimde konuya abone olması ve mesaj alabilmesi, yedekli sistemlerin oluşturulmasına yarar. Robot için otonom yapılan görevlere dışarıdan müdahale etme olanağı verir. ROS konu komutları Çizelge 2.2’de verilmiştir. ROS konuları aşağıdaki özelliklere sahiptir.

- Çoktan çoğa bağlantılar için uygundur.
- Asenkron iletişim için kullanılır.
- Geribildirim fonksiyonlarına sahiptir.
- Çok kanallı çalışabilir.
- İstenilen mesaj türü ile yayın yapabilir.

Çizelge 2.2: ROS konu komutları

Komut	Açıklama
rostopic	ROS konuları ile ilgili bilgi verir, list, echo, info, pub parametreleri ile farklı seviyelerde bilgi görüntülenebilir.
rostopic list	Konuları listeler
rostopic info konu_adi	Parametre olarak verilen konuya ait bilgileri görüntüler
rostopic echo konu_adi	İlgili konunun yayınladığı mesajları görüntüler
rostopic pub topic_name m_t topic_adi	Yayınlamak üzere ilgili konuya mesaj gönderir. m_t mesaj tipi anlamında kullanılmıştır.

2.4.4 ROS Mesaj

Mesajlar ilkel tiplerden (string, integer, float, boolean) ya da daha önce oluşturulan farklı tiplerden oluşabilir. Bir robot motorunu temsil eden düğüm, durumunu bir mesajla yayınlayabilir.



Şekil 2.4: ROS düğüm konu düğüm ilişkisi

Şekil 2.4’te belirtildiği gibi bir düğüm bir konuya veri aktarabilir, ya da bir konuya abone olabilir. Çizelge 2.3’de ROS mesaj komutları verilmektedir.

Çizelge 2.3: ROS mesaj komutları

Komut	Açıklama
rosmg -help	Yardım dosyasını görüntüler
rosmg list	Tüm mesajları listeler
rosmg show konu_adi	İlgili konunun yayınladığı mesaj bilgisini gösterir
rosmg show std_msgs/String	İlgili mesaj içeriğini görüntüler
rosmg package nav_msgs	İlgili paketteki tüm mesajları listeler
rosmg packages	Mesaj içeren tüm paketleri listeler

2.4.5 ROS Servis

Konular çoklu iletişim için eş zamansız bir yöntemdir. Servisler ise iki düğüm arasında senkron bir şekilde iletişim sağlar. Çizelge 2.4’de ROS servis komutları verilmektedir.

- Senkron çalışır
- Bire bir iletişim için uygundur
- İstenilen mesaj türü oluşturulabilir.

Çizelge 2.4: ROS servis komutları

Komut	Açıklama
rossrv -h	Yardım dosyasını görüntüler
rossrv list	Tüm servisleri listeler
rossrv show servis_adi	İlgili servis bilgilerini görüntüler
rosservice call servis_adi	İlgili servisi çağırır

2.4.6 ROS Eylem

ROS'ta üç iletişim türü vardır. Konular, servisler ve eylemler. Konu yayını daha önce belirtildiği gibi birden fazla aboneye yayın yapabilir, ancak iletişim tek yönlüdür, yanıt beklenmez. Servislerde ise istek gönderilir ve yanıt alınır, ancak ilerleme durumu hakkında geri bildirim alınmaz. Eylemler ise servisler gibi çalışmakla beraber, farkı olarak işlemin ilerlemesi hakkında geri bildirim alır. Eylemler iptal edilebilir. Bu durum istenen eylem gerçekleştirilirken geri bildirimler takip edilir ve istenmeyen bir durum olduğunda işlem iptal edilebilir. Özellikle sonlandırılması zaman alan, kritik işlerle ilgili olarak kullanılır.

2.5 ROS Araçları

ROS, komut seti ve konsolu dışında, aynı zamanda çeşitli araçlar içerir. OpenCV, rviz, Gazebo gibi yapılarla entegre çalışabilir ve farklı dillerde geliştirme yapılabilir.

2.5.1 rviz

rviz, ROS için üç boyutlu bir görselleştirme ortamıdır. rviz, robotun anlık durumu hakkında bilgi edinimi sağlar. Sensörlerden gelen bilgilerin görselleştirilmesi ve kaydedilmesi görevini gerçekleştirir. Geliştirme ve hata ayıklama amacıyla sıklıkla kullanılır. Anlık olarak gelen veriler gözlemlenebilir. Çizelge 2.5'de rviz komutları verilmektedir.

Çizelge 2.5: rviz komutları

Komut	Açıklama
roslaunch rviz	rviz yardım dosyasını görüntüler
roslaunch rviz	rviz aracını çalıştırır

2.5.2 rqt

rqt, düğümler ve konular arasındaki hiyerarşiyi bir görsel bir diyagram olarak göstermek için Qt yazılım çerçevesi kullanılarak geliştirilmiş bir programdır. İki boyutlu çıktı verir. Çalışmada rqt ile alınan çıktı ile düğümlerin ve konuların mevcut durumu hakkında bilgi verilmiştir. Görsel diyagram, komut satırından “rqt” komutu verilerek başlatılır. Çizelge 2.6'da rqt komutları verilmektedir.

Çizelge 2.6: rqt komutları

Komut	Açıklama
rqt	rqt aracını başlatır
rqt --list-plugins	rqt eklentilerini listeler

2.5.3 rosbag

ROS çalışırken yayınlanan mesajlara daha sonra ulaşabilmek için kullanılan güçlü bir araçtır. Mesajlar, .bag uzantılı dosyalara kaydedilir. Kullanılabilir tüm konulara ait mesajlar kaydedilebilir. Komut satırı aracıdır. rosbag c++ ve python programlama dillerinden erişilebilen kod uygulama programlama arayüzü (API)'ne sahiptir. Yaygın olarak kullanılan Matlab gibi programlarla, .bag uzantılı dosyalara erişilebilir ve mesajlar okunabilir. “rosbag record -a” komutu ile başlatılmış tüm konulara ait mesajlar kaydedilmeye başlanır. Çizelge 2.7’de rosbag komutları verilmektedir.

Çizelge 2.7: rosbag komutları

Komut	Açıklama
rosbag --help	Yardım dosyasını görüntüler
rosbag info bag_file.bag	.bag uzantılı dosyaya ait bilgileri görüntüler
rosbag record -a	Çalışan konulara ait mesajları kaydetmeye başlar.
rosbag play bag_file.bag	İlgili .bag uzantılı dosyayı oynatır.
rosbag record -O bag_file /paket /konu	Belirtilen pakette çalışan konuya ait mesajları bag_file.bag adıyla kaydetmeye başlar.

2.5.4 roslaunch

ROS üzerinde çalışması istenen düğümler tek tek yürütülebilir. Fakat düğüm sayısı arttığında ve bunlar belirli sıra ile çalıştırılmak istendiğinde karmaşıklık artar. Ayrıca üzerinde çalışılan düğümlerin haricinde birçok düğüm rutin olarak başlatılabilir. Tüm bu işlemlerin hem yapılması hem de hatırlanması kullanıcı açısından zaman alıcı ve zor olabilir. Bu işlemleri daha kolay hale getirmek için launch uzantılı roslaunch dosyalarından faydalanılır. Bu dosyalar başlatılacak düğümlerin listesini XML formatında tutar. Tanımlanan düğümlere parametre verilebilir. Tüm sistemi başlatmak için tek bir roslaunch dosyası yeterlidir. Fakat organizasyon açısından birden fazla roslaunch dosyası kullanılabilir. Birden fazla roslaunch dosyası tek bir ana dosya

içinden çağrılabilir. Oluşturulan roslaunch dosyaları “roslaunch” komutundan sonra boşluk bırakılarak dosya ismi ile başlatılır. roslaunch komutu ROS sunucusunu başlattıktan sonra, çağrılan launch dosyasındaki düğümleri sırası ile başlatır.

2.5.5 ROS Konsolu ve Komut Seti

ROS sunucusu ile iletişim kurmak, çeşitli işlemleri başlatmak, bitirmek, programları derlemek, düğümleri, konuları, servisleri ve diğer çalışan uygulamaların durumu hakkında bilgi almak, ROS parametrelerini görüntülemek vb. birtakım işlemler için terminal ekranından ROS komutları gönderilebilir. Çizelge 2.8’de ROS komut seti verilmiştir.

Çizelge 2.8: ROS komut seti

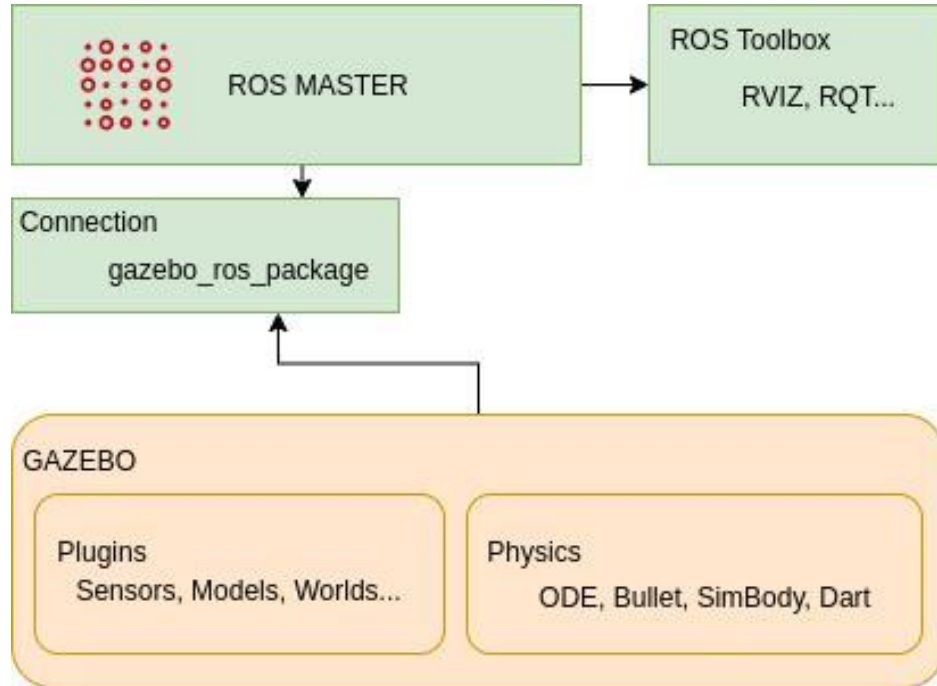
Komut	Açıklama
ROS_MASTER_URI	Robot için URI adresi set eder. export ROS_MASTER_URI=http://<robotip>:11311
ROS_HOSTNAME	ROS sunucusu için hostname set eder. export ROS_HOSTNAME=<robot-ip>”
roscd	Belirtilen klasöre geçilir. “roscd paket_adi”
rossrv	Belirli bir servise ait bilgileri görüntüler. rossrv show servis_adi
rosmg	Belirtilen mesaj tanımıyla ilgili bilgileri görüntüler. rosmg show mesaj_tipi
roslaunch	Belirtilen ros programını çalıştırır. Çalışacak dosyanın paketi, dosyanın adı ve varsa parametreleri verilir. “roslaunch program_paketi program_dosyasi arg1 arg 2 ...”
roscore	roscore isimli çekirdek düğümünü başlatır. “roscore”
roslaunch	roslaunch, belirli konular ve parametrelerle bir grup düğümü başlatmak için kullanılır. Önce ROS Master olarak da bilinen bir “roscore” olup olmadığını ve çalışıp çalışmadığını kontrol eder. roslaunch bulunmazsa roscore başlatılır. Ardından, komut başlatma dosyası ile başlatma dosyasındaki tüm düğümleri çalıştırır.

2.6 Benzetim Ortamı

Robot çalışmalarında benzetim ortamı temel gereksinimlerden biridir. Tamamen gerçek ortamda çalışmak hem zaman alıcı hem de pahalıdır. Gerçek ortamlarda robotların zarar görme durumu olduğundan tüm testlerin bu ortamlarda gerçekleştirilmesi mümkün değildir. Bundan dolayı robotlar için benzetim ortamları geliştirilmiştir. Algoritmaların ve testlerin başarılı bir şekilde gerçekleşmesi için, benzetim ortamının robot özelliklerini gerçeğe en yakın şekilde yansıtması gerekmektedir. Çalışmada benzetim ortamı olarak Gazebo tercih edilmiştir.

2.6.1 Gazebo

Güney Kaliforniya Üniversitesi tarafından 2002 yılından itibaren geliştirilmeye başlanmıştır. Sadece iç ortamlar değil, dış ortamlar için de kullanışlı olacak şekilde bir yapı üzerine bina edilmiştir. 2011 yılında ROS mühendisleri tarafından, ROS-Gazebo entegrasyonu gerçekleştirilmiştir. 2012 yılında Açık Kaynak Robotik Vakfı (OSRF) tarafından sorumluluk alınıp desteklenmeye devam edilmiştir [34].



Şekil 2.5: ROS-Gazebo platformu genel mimarisi

Gazebo, Linux dağıtımları üzerinde oldukça iyi çalışır. Gazebo ile 3D ortamlar oluşturulup robot ve modeller tasarlanabilir. Gazebo üzerinde dünya ile tabir edilen iç ve dış ortamlar ve yine model olarak tarif edilen dünya ortamındaki varlıklar Standart Tanım Formatı (SDF) ile tanımlanır. Şekil 2.5'te Gazebo ve ROS entegrasyonu

sunulmuştur. Gazebo karmaşık iç ve dış ortamlarda robotları ve ortamdaki varlıkları doğru ve verimli bir şekilde benzetim ortamına dahil eder. Oyun motorlarında olduğu gibi fizik benzetimi yapabilir. Bununla birlikte aşağıdaki yeteneklere sahiptir;

- İç ve dış ortamlar tasarlanabilir.
- Robot tasarlayabilir.
- Gerçekçi senaryolarla regresyon testleri yapabilir.
- Çoklu fizik motoruna sahiptir.
- Robot ortamları ve robot kütüphanelerine sahiptir.
- Çeşitli sensörlere sahiptir.

2.6.2 SDF dosya biçimi

SDF dosya formatı robot benzetimleri görselleştirme ve kontrol nesneleri için tanımlama yapılabilen bir XML biçimidir. SDF ile nesnelerin statik-dinamik durumları, bulunduğu ortamın aydınlatılması ve diğer fiziksel özellikleri tanımlanabilir, istendiğinde genişletilebilir.

SDF’de özelde robotlar için, robot türü fark etmeksizin (tekerlekli, insansız vb.) bir robota ait kinematik, dinamik özellikler, robot içi sensörler, yüzey özellikleri, dokular, eklem sürtünmeleri vb. özellikler tanımlanabilir. Daha önceden tanımlanmış SDF dosyaları Gazebo ortamına dahil edilebilir.

2.7 OpenCV

Gerçek zamanlı bilgisayarlı görme uygulamalarında sık kullanılan görüntü işleme amaçlı bir kütüphanedir. Açık kaynak olup BSD lisansı ile lisanslanmıştır. Birçok popüler bilgisayarlı görme algoritmalarına ait kütüphaneler barındırmaktadır. Tez çalışmasının görüntü işleme aşamalarında, görüntülerin ön işlemlerinde kullanılmıştır.

OpenCV, görüntü işleme ve makine öğrenmesi üzerine çok sayıda kütüphane barındırmaktadır. Yüz tanıma, nesne tanıma gibi algoritmaların yanında nesne sınıflandırma, optik karakter tanıma, insan davranışlarını ayırt etme gibi ileri seviye algoritmalara da sahiptir.

2.8 Python Programlama Dili

1991 yılında Guido Russom tarafından geliştirilmiş nesne yönelimli bir dildir. Yüksek seviyeli bir dil olduğundan, karmaşık uygulamaların hızlı bir biçimde gerçekleştirilmesi için çok uygundur. Python yorumlayıcı tarafından okunur. Bundan dolayı, kod dosyaları üzerinde değişikliklerin yapılması tümünün derlenmesini gerektirmediği için kolaydır. Yapay zekâ, yapay sinir ağları, matematiksel işlemler için çok geniş kütüphanelere sahiptir. Hemen hemen tüm işletim sistemlerinde çalışır, ideal geliştirme ortamı Linux işletim sistemidir. Tez çalışmasında ROS için yazılan algoritmalar Python dilinde kodlanmıştır.

ROS ile Python arasında “rospy” isimli bir arayüz bulunmaktadır. Bu arayüz istemci kütüphanesi görevi görür. Böylelikle Python programcıları ROS konularına, servislerine ve ROS Master parametrelerine kendi programları ile ulaşabilirler. “rospy” python ile ROS Master sunucusuna erişime ve fonksiyonlarını kullanma kütüphanesidir. “rostopic” ve “rosservice” gibi araçlar “rospy” arayüzünü kullanılarak geliştirilmiştir.

2.9 Turtlebot

Willow Garage tarafından tasarlanan, açık kaynak yazılım içeren ve düşük maliyetli kişisel bir robot kiti olan Turtlebot şekil 2.6’da gösterilmektedir. Geliştirilmesi başta Open Robotics ve Intel olmak üzere birçok ortak tarafından yapılmaktadır [35]. ROS için de kütüphanesi mevcuttur. 2D/3D mesafe sensörü, tek kartlı bilgisayar ve montaj kitinden oluşur. Gerçek özellikleri ile ROS-Gazebo ortamında benzetim yapılabilir. Bu çalışmada algoritmaların uygulandığı model robot olarak kullanılmıştır.



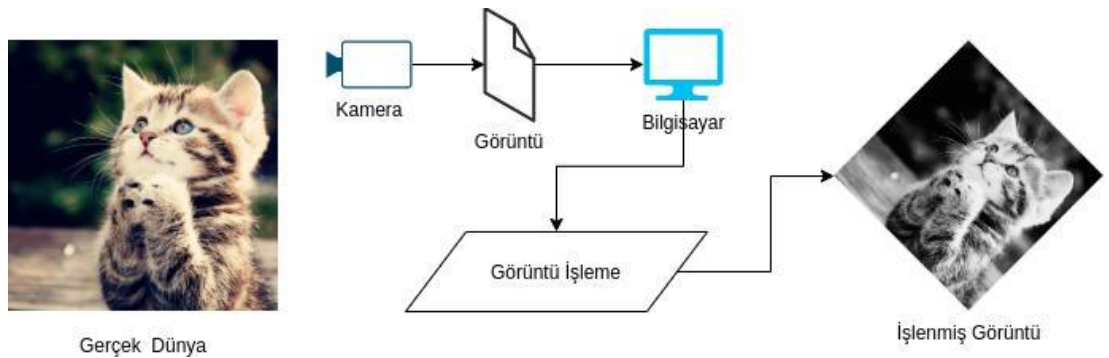
Şekil 2.6: Willow Garage tarafından geliştirilen Turtlebot robot kiti

3. GÖRÜNTÜ İŞLEME

Görüntü işleme, bir sensör aracılığı ile elde edilen görüntünün bilgisayara aktarılarak, çeşitli yöntemlerle işlenmesidir. Görüntüde işleme, görüntünün düzenlenmesi, analiz edilmesi ya da görüntüden anlamlı bilgilerin çıkarılması amacıyla yapılır. Görüntü işleme yaklaşık son 70 yılda, uzay araştırmaları, tıp, genetik, biyoloji, endüstri, tekstil, jeoloji, madencilik, inşaat, şehircilik, gıda, çevre gibi alanlarda uygulanmakta ve uygulama alanları sürekli genişlemektedir. Görüntü işleme, mobil ve robotik teknolojilerindeki gelişmelerle yeni amaçlar için farklı yaklaşımların geliştirildiği uygulamalarda, temel bileşenlerden biri haline gelmiştir [36,37]. Günümüzde, uzaktan algılamadan otonom araçlara, fotoğraf uygulamalarından tarıma kadar geniş bir uygulama alanına sahiptir. Modern bilgi toplumunun temel taşlarından biri olan çoklu ortam (multimedya) sistemleri büyük ölçüde sayısal görüntü işlemeye dayanmaktadır [38].

Sinyal işleme, analog ve sayısal sinyallerin işlenmesi ile ilgilenen bir disiplindir. Analog sinyaller iki boyutludur ve elektrik sinyallerinin manipüle edilmesine dayanır. Televizyon görüntüsü buna örnektir. Sayısal sinyal işleme ise analog sinyal işlemeye göre çok daha geniş bir alanda kullanılır. Sayısal işleme ise bilgisayarlar aracılığı ile yapılır, bundan dolayı üzerinde birçok farklı algoritma uygulanabilir. Tez çalışmasının içeriği tamamen sayısal görüntü işlemeye yöneliktir.

Görüntü işleme, girdinin bir görüntü olduğu ve çıktının görüntü veya görüntü ile ilgili karakteristik özellikler olabileceği, bilgisayarlar tarafından yürütülen algoritmalarla gerçekleştirilen bir sinyal işleme/örüntü tanıma türüdür.



Şekil 3.1: Genel görüntü işleme akışı

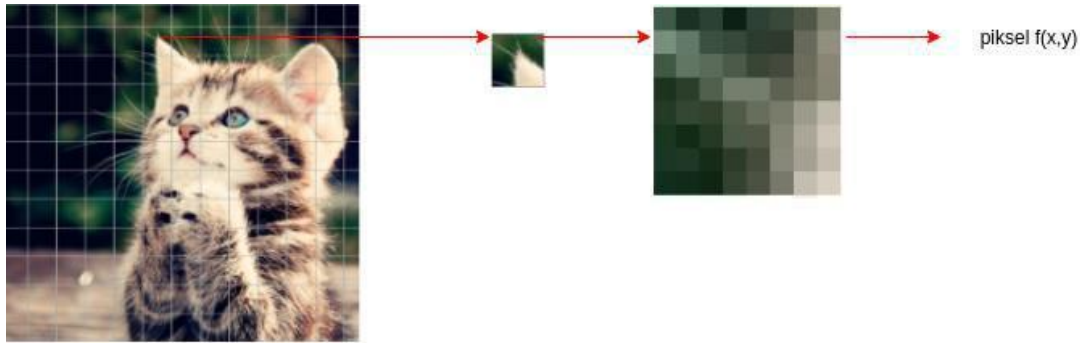
Şekil 3.1’de girdiden çıktıya kadar görüntü işleme adımları sunulmuştur. Burada gerçek dünyadan alınan ve “Görüntü” olarak ifade edilen bir görüntünün işlenmesi

içim sayısal olarak bilgisayarda kaydedilmesi gerekir. Görüntü dosyası üç adet iki boyutlu matris yapısında bir dosyadır. Görüntü dosya haline getirildikten sonra amaca yönelik olarak çok farklı algoritmalarla işlenerek bir çıktı elde edilir.

Görüntü üzerindeki her bir matris sırası ile kırmızı, yeşil ve mavi renklerini temsil eder. Matrislerin üzerinde her bir pikselin, $f(x,y)$ ile ifade edilen bir karşılığı bulunmaktadır. Rengin tonuna ve doygunluğuna göre bir pikselin değeri 0–255 arasında bir değerdir. Böylece görüntü üzerindeki belirli bir noktadaki renk, gerçeğe en yakın şekilde ifade edilmiş olur.

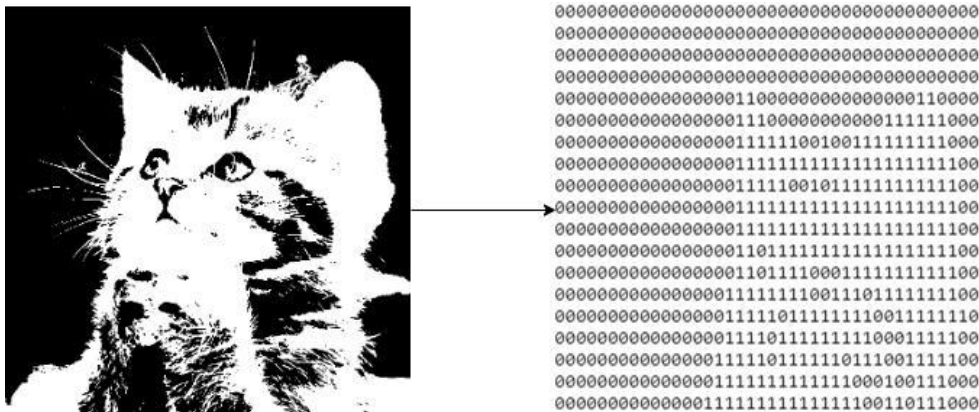
3.1 Sayısal Görüntü İşleme

Sayısal görüntü işleme disiplini, görüntülere özgü teknikleri de içeren geniş bir disiplindir. Bir görüntü iki sürekli değişken x ve y 'nin bir fonksiyonu olarak $f(x,y)$ ifade edilir.



Şekil 3.2: Sayısallaştırılmış bir görüntüdeki pikseller

Şekil 3.2’de sayısal bir görüntü için bir örnek sunulmuştur. Görüntü n satır ve m sütundan oluşur. Satır ve sütun kesişimleri piksel olarak adlandırılır. Görüntü renklendirme açısından üç gruba ayrılır. Bunlar ikili, gri ve renkli görüntülerdir.



Şekil 3.3: Görüntünün ikili görüntü olarak sayısal değeri

İkili görüntü, 2B görüntünün piksellerinin '0' ya da '1' değerlerinden birini alması ile oluşur. '0' siyah, '1' ise beyaz alanları gösterir. İkili görüntüler hızlı işlendiği için oldukça sık kullanılır. Çalışmada robot için sağlanan görüntüler ikili görüntüye çevrilmektedir. Eşikleme bölümünde bu konu detaylı olarak anlatılmıştır.

Şekil 3.3'te ikili görüntü örneği verilmektedir. Siyah alanların '0', beyaz alanların ise '1' değeri aldığı görülmektedir. Gri ton görüntülerde her bir piksel 0–255 arasında bir değer alır. Siyahtan beyaza doğru 256 farklı renk tonu elde edilmiş olur. Renkli görüntülerde ise bir piksel için üç farklı renk değeri vardır. Bunlar kırmızı, yeşil ve mavidir ve her biri yine 0–255 arasında değerler alır. Üç rengin kesişimi o noktadaki rengi ve tonu ifade eder. Şekil 3.4'te ikili, gri ve renkli görüntü birlikte verilmiştir.



Şekil 3.4: İkili (sayısal), gri ton ve renkli görüntü örnekleri

Sensörlerin görüntü elde etmesi ile ilgili standartlar bulunmaktadır. Bunlar satır ve sütun sayıları (640×480, 800×600, 1920×1080 vb.) ile akan görüntüler için zamana bağlı olarak değişen görüntü sayısı (25 fps, 30 fps) gibi parametreler temel alınarak oluşturulan standartlardır. Çalışmada iki boyutlu, tek renkli ve statik görüntü örneği ele alınmıştır.

Görüntüler bilgisayarda sayısal olarak işlenmek için bir sayı matrisine dönüştürülebilir. Sayısal görüntü işleme bu sonlu sayıların manipülasyonuna dayanır. Sayısal görüntülerin işlenmesi kategorik olarak görüntü geliştirme, düzenleme, analiz gibi işlemlerden oluşur. Görüntü düzenleme ise bozulmanın istatistiksel ya da matematiksel bir tanımının bulunduğu bozuk görüntüleri işlemeyi amaçlar. Görüntü analizi, bir görüntünün üzerinden değişiklik yapmaktan çok, görüntünün işlenerek ondan anlamlı bilgilerin çıkarılmasıdır. Görüntü analizi örnekleri, görüntü bölütleme, kenar çıkarma, doku ve hareket belirlemedir.

3.2 Görüntü İşleme Yaklaşımları

Uygulama amacına göre farklı görüntü işleme teknikleri ve algoritmaları geliştirilmiştir. Bunlar düzeltme, dönüşüm ve ayrıştırma olmak üzere üç ana başlıkta değerlendirilebilir.

3.2.1 Düzeltme

Düzeltilme işlemleri, görüntünün iyileştirilmesi ve yapılandırılması ile ilgili işlemlerdir. Bunlar; restorasyon, yeniden yapılandırma ve mozaikleme olarak sınıflandırılır. Düzeltme işlemleri, görüntü üzerinde renk dönüşümleri ile ilgili olmayıp, görüntünün olması gereken haline kavuşturulması için uygulanır. Örnek olarak bozuk sensörlerden kaynaklı pikseller varsa, bunlar çevre piksel değerleri temel alınarak düzeltilebilir. Görüntünün kalan kısmı ile ilgili bir değişiklik yapılmaz ve bu bir düzeltme işlemi olmuş olur.

3.2.2 Dönüşüm

Dönüşüm işlemleri, görüntünün amaca uygun olması ve resmin tamamını içeren işlemlerdir. Dönüşüm işlemlerinin bir kısmı piksel değerlerinin değiştirilmesiyle ilgiliyken, bir kısmı da birbirlerine göre konumu ile ilgilidir. Dosya seviyesindeki işlemler de dönüşüme dahildir. Temel olarak, kontrast germe, histogram eşitleme, veri sıkıştırma, döndürme gibi işlemler, dönüşüm içinde yer alır.

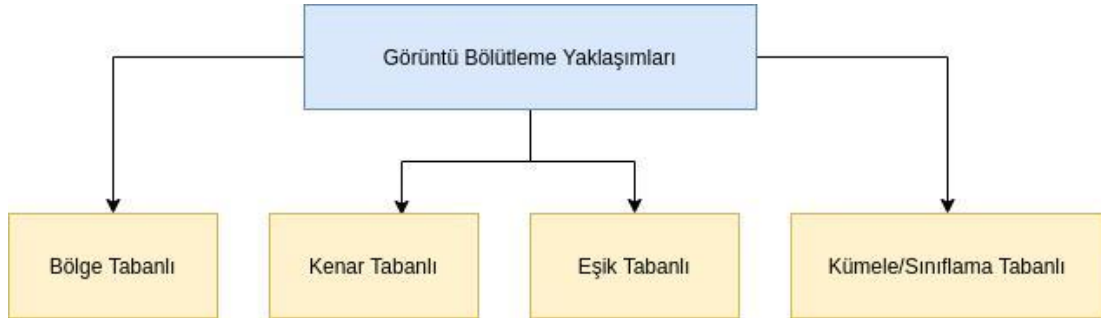
3.2.3 Ayrıştırma

Ayrıştırma, görüntünün üzerinden farklı anlamlar çıkarmak amacıyla, piksel değerlerinin birbiriyle olan ilişkisine (uzaklığına, benzerliğine) bağlı işlemlerdir. Bölütleme ve sınıflama gibi iki temel yaklaşım içerir. Bunun yanı sıra görüntü üzerinde nesnelerin ayırt edilmesi, kenarların bulunması, alt görüntüler çıkarılması, benzerliklerin bulunması gibi işlemleri de içine alır. Ayrıştırma işlemlerinde mevcut veriler kullanılabilir, bundan dolayı bir kısmı gözetimli ayrıştırma olarak kabul edilir.

3.3 Görüntü Bölütleme

Genel görüntü işleme yaklaşımlarından biri de görüntü bölütlemedir. Alınan görüntüde bir nesnenin aranması, benzerliklerin bulunması gibi birbirinden tamamen farklı amaçlar için yine birbirinden tamamen farklı algoritma ve teknikler uygulanmaktadır.

Görüntünün özelliklerine göre farklı parçalara ayrıştırılmasıyla özetlenir. Tüm görüntülere uygulanabilecek genel bir yaklaşım yoktur, görüntüden çıkarılmak istenen anlama bağlı olarak uygun bir yöntem seçilir. Aynı yöntemin uygulanması durumunda bile başarı, görüntüden görüntüye değişir. Görüntü bölütleme, amaca ve yönteme göre gruplanabilir.



Şekil 3.5: Görüntü bölütleme yaklaşımları

Şekil 3.5’te kullanılan yöntemlere göre bir kategorik değerlendirme yapılmıştır [46,47]. Her bir kategoride birçok algoritma kullanılmakla beraber, en yaygın kullanılan algoritmalar ele alınmıştır.

3.3.1 Bölge Tabanlı Görüntü Bölütleme

Bir görüntünün R (region) ile gösterilen bölgesi; gri seviye, renk, doku gibi kriterlerle ilgili olarak görüntünün homojen bir alt kümesi olarak açıklanmaktadır. Bir görüntüdeki R bölgesi, benzer özelliklere sahip bir grup pikselden oluşmuştur.



Şekil 3.6: Bölge tabanlı görüntü bölütleme örnekleri

Şekil 3.6’da bölge tabanlı bölütleme örneği sunulmuştur. Görüntünün bölgelere ayrılmasında bölge büyütmesi ve ayırma/birleştirme teknikleri kullanılır.

3.3.1.1 Bölge büyütmesi

Bu teknikte, görüntü içinden rastgele pikseller seçilir. Anlatımda diğer piksellerle karışmaması için bu piksellere tohum denilmiştir. Tohum seçimi görüntü üzerinden

manuel ya da otomatik olarak yapılabilir. Tohum ile komşuları arasında benzerlik ölçüleri tanımlanır. Tohum olarak tabir edilen piksel ile benzer özellikler gösteren (gri ton, renk, parlaklık gibi) pikseller gruplanır. Görüntüdeki nesnelerden birinin iç yüzeyinden seçilen tohuma komşu pikseller belirlenirken, nesnenin kenarlarına gelince bölge tamamlanmış olur. Bölge tabanlı bölütlemeye, benzerliğe karar verilmesi için uygun bir eşik değeri yöntemi kullanılır. Eşik değeri ile gri değerlerin farkı, varyansı ve mekânsal uzaklıkları tespit edilir.

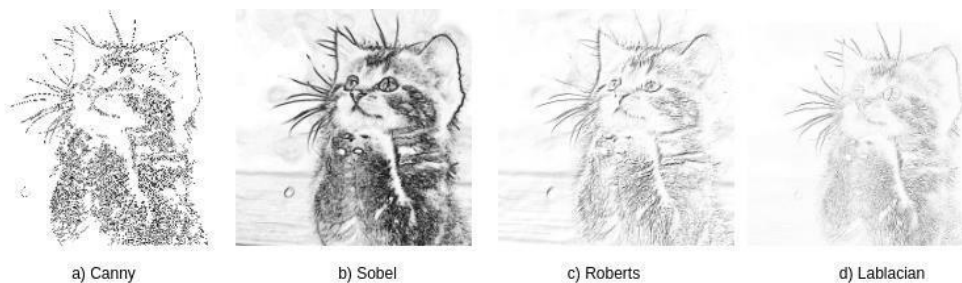
3.3.1.2 Ayırma ve birleştirme

Bu teknikte görüntünün tamamı üzerinden (tek bir bölgeden) görüntünün alt bölgelerine ulaşılır. Görüntünün tamamında varyans çok büyükse, görüntü bölgelere ayrılır. Ayrılan bölgeler içinde varyansı büyük olanlar tekrar bölgelere ayrılırken, birbirine benzer olan bölgeler birleştirilir. Bölünme ve birleşme kriterini geçen bir değer olmayana dek işlem devam eder.

3.3.2 Kenar Tabanlı Görüntü Bölütleme

Görüntülerdeki nesneler ışık ve gölge gibi etkilerden dolayı kenarlara sahiptir. Bu durum görüntü üzerine, nesnelerin kenar piksellerinde ani gri seviye değişimleri olarak yansır. Bu sert piksel geçişlerinin tespit edilerek nesnelerin kenarlarının bulunması yöntemi, kenar tabanlı bölütleme ile ifade edilir. Bunlar, türev almaya dayalı yöntemlerdir.

Kenar tabanlı bölütleme yönteminde filtreler kullanılır. Genel olarak 3×3 'lük filtreler tercih edilir. Canny, Sobel, Roberts ve Laplacian en çok kullanılan kenar bulma filtreleridir [39-42]. Şekil 3.7'de Canny, Sobel, Roberts ve Laplacian filtrelerinin aynı görüntü için ürettiği sonuçlar gösterilmiştir.



Şekil 3.7: Kenar bulma algoritmaları ile işlenmiş örnek görüntüler

Kenar bulma algoritmalarında filtre boyutları değiştirilerek farklı sonuçlar elde edilebilir.

3.3.3 Eşik Tabanlı Görüntü Bölütleme

Eşikleme yöntemi, görüntü işlemede uzun zamandır kullanılan bir yaklaşımdır. Bu yaklaşım gri tonlamalı görüntülere uygulanır [43]. Basit tanımı ile görüntü üzerinde hedef seçilen alan için bir değer belirlenir. Görüntü pikselleri, belirlenen bir eşik (T) değerinin altında ya da üstünde olma durumuna göre değerlendirilir. Genel olarak eşiklemenin başarısı; ışık, gölge gibi etkenlere bağlıdır. Görüntü ikilileştirme olarak da bilinen, görüntü üzerinde belirlenen eşik değerine göre piksellerin tümü iki değer alabilir. Uygulanan eşik değeri farklı yöntemlerle seçilebilir [44]. Görüntünün tümü için tek bir eşik değeri seçilebileceği gibi, lokal olarak da eşikleme değeri seçilebilir.



Şekil 3.8: Gri ton görüntüye uygulanan eşik değeri ve etkisi

Şekil 3.8’de gri ton görüntüye T değeri olarak sırası ile 40 ve 200 verilmiştir. Şekilde T değerinin görüntüye olan etkisi görülmektedir. Bu etki için uygulanan eşik değeri formülü Denklem 3.1’de verilmiştir.

$$q(x,y) \begin{cases} 1, & g(x,y) > T \\ 0, & g(x,y) \leq T \end{cases} \quad (3.1)$$

3.3.4 Kümeleme/Sınıflandırma Tabanlı Görüntü Bölütleme

Kümeleme, bir görüntüyü oluşturan piksellerin belli normlara göre gruplanmasıdır. Amaç, görüntüde kendi içinde homojenlik taşıyan alanların belirlenmesidir. Kümeleme tabanlı yaklaşımlarda; K-Ortalama, Bulanık C-Ortalamalar gibi algoritmalar kullanılmaktadır. Bu tür algoritmalarda birbirine göre avantaj ve dezavantajlar bulunduğundan, problemin büyüklüğüne ve karmaşıklığına göre uygun algoritma seçilmelidir. Kümeleme algoritmaları zaman alıcı olduğundan bölütleme verisinin azaltılması performans için temel unsurdur.

3.3.4.1 K-ortalama

K-Ortalama, denetimsiz bir kümeleme algoritmasıdır [48]. Sınıflandırma algoritmalarına benzer, ancak sınıflar süreç sonunda ortaya çıkar. Veri kümesindeki elemanlardan en çok birbirine benzeyenlerin gruplar oluşturulur. Bir verinin küme merkezine uzaklığına bakılarak kümeye atama yapılır. Manhattan, Euclid, Minkowski ve Pearson gibi uzaklık metrikleri kullanılır.

Algoritmanın temel amacı N adet veriyi K adet kümeye ayırmaktır. Küme içi benzerliklerin maksimum, farklılıkların ise minimum olması amaçlanır. Algoritmanın çıktısı önceden tanımlanacak küme sayısına bağlı olduğu için, kötü bir K değeri seçildiğinde, algoritma çıktısı farklı kümeleme sonuçlarına yaklaşacağı için istenen sonuç elde edilemeyebilir. Renkleri önceden bilinmeyen 100 çift ayakkabının renklerine ve numaralarına göre kümelenmesi örnek olabilir. Bu örnek için K değeri 2 alınırsa ve 2'den fazla renk varsa küme sayısı problemin çözümü için yeterli olmayacaktır. Bu durum K değerini uygun olarak belirlemenin önemini göstermektedir.

Farklı yöntemler kullanılarak K değeri belirlenebilir. Bunlardan biri de 1'den 10' kadar farklı K değerleri ile algoritmayı işletmek ve her yinelemede küme için uzaklıkların değerine bakarak seçim yapmaktır. Küme için uzaklıkların en düşük olduğu değer, K için en doğru küme sayısı demektir.

3.3.4.2 K-ortaylar

Veri kümesinde, kümeyi en iyi temsil eden K tane eleman belirlenir, bunlara küme ortayları denir. Ortay örnek, diğer örneklerle olan ortalama uzaklığı minimum yapan kümenin en merkezi örneğidir. Yöntem her bir veri ve o verinin referans noktası arasındaki benzerliği maksimum yapmayı amaçlamaktadır [49].

3.3.4.3 K-En yakın komşuluk

K-En Yakın Komşuluk (KNN) algoritması, birçok alanda kullanılan başarılı bir sınıflandırma algoritmasıdır. Önceden etiketlenmiş bir sınıf seti verilir. Görüntüler üzerinde uygulanması, verilen eğitim setine uygun olarak pikselleri sınıflandırmasıdır. Sınıflandırma işlemini uzaklık hesaplarına göre yapar, yaygın olarak Öklid mesafesi kullanılır. K değeri en yakın komşu sayısını temsil eder [50].

3.3.4.4 Bulanık c-ortalama

Yaygın kullanılan kümeleme yöntemlerinden biridir. Dunn tarafından önerilmiş (1973) ve Bezdek (1981) tarafından geliştirilmiştir [51]. Bulanık c-ortalamalar yönteminin kümeleme işlemi, amaç fonksiyonunun minimize edilmesine dayanır. Her bir veri iki ya da daha fazla kümeye ayrılabilir. Veri setindeki her bir veri 0 ile 1 arasında değişen bir üyelik değerine sahiptir. Verinin tüm kümelere ait üyelik değerleri toplamı 1'dir. Küme üyelik değerlerinden en büyük olanı, ait olduğu kümeyi belirtir. Amaç fonksiyonu istenen değere yakınsadığında, kümeleme tamamlanır.

Bunlar dışında, Destek Vektör Makinaları, Temel Bileşenler Analizi ve Bayes gibi farklı sınıflandırıcılar da görüntü işlemede kullanılmaktadır.

3.4 Robotlarda Bilgisayarlı Görme

Bilgisayarlı görme, sayısal görüntülerin amaca uygun olarak anlamlandırılması için çok çeşitli yaklaşım ve yöntemlere sahip disiplinler arası bir alandır. Sayısal görüntülerin elde edilmesi, işlenmesi, analiz edilmesi ve çeşitli anlamlar çıkarılabilmesi amacıyla, gerçek dünyadan yüksek boyutlu verilerin bilgisayar ile işlenmesi için yöntemler sunar [45]. Alt alanları arasında sanal gerçeklik, olay algılama, video izleme, nesne tanıma, 3B poz tahmini, öğrenme, indeksleme, hareket tahmini ve görüntü restorasyonu gibi alanlar bulunur.

Bilgisayarlı görme ile görüntü işleme birbirinden farklıdır. Görüntü işleme, görüntü üzerinde basitleştirme, iyileştirme, dönüşüm gibi fotometrik işlemlerle ilgilenir. Bir çeşit sayısal sinyal işleme türüdür. Amaç görüntüden yeni bir görüntü oluşturmaktır ve görüntünün içeriğinin anlaşılması ile ilgilenilmez. Bilgisayarlı görme ise görüntü işleme tekniklerinden faydalanmakla birlikte, görüntüden anlam çıkarmayla ilgilenir. Bilgisayarlı görme için görüntü işleme teknikleri önemlidir ancak bunlar tek başına yeterli değildir. Görmenin karmaşıklığından kaynaklanan sorunları çözmek esas amaçtır.

Görüntüden anlam çıkarmak insan için çok basit bir işlem olmasına karşın bilgisayarlar için son derece zordur. Genel amaçlı bir görme makinesi inşa etmek için gerekli disiplinler henüz oluşturulamamıştır. Son yıllarda gerek donanım gerekse de yazılım alanındaki ilerlemeler çok fazla veri seti oluşturmuş olsa da bunlar çözülecek probleme göre çok zayıf kalmaktadır.

Robotlar için görme ise yapay görme, elektronik, makine, fizik, bilgisayar bilimleri gibi farklı alanlara ait teknikleri kullanan çok disiplinli bir konudur. Robotlar için görme, çoğu kez dinamik ortamlar için düşünülür. Robot görmesi otonom araçlardan insansız hava araçlarına, servis robotlarından tehlikeli işler için üretilen robotlara kadar çok çeşitli alanlarda ve çok çeşitli amaçlar için geliştirilmektedir.

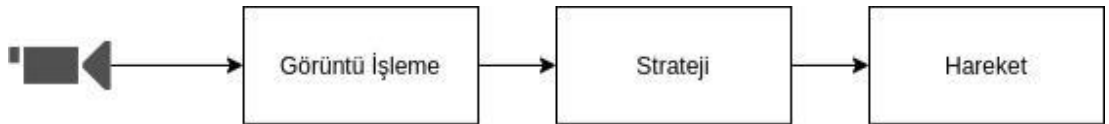
Robot görüşü, sadece görüşten anlam çıkarma değil, bunun bir üst adımı olan, robotun işlenen ve anlam çıkarılan sürecin sonunda ne yapması gerektiğine karar vermekle de ilgilidir. Hatta geri bildirimde bulunmak, diğer robotlarla haberleşmek gibi konularla bu görevler çoğaltılabilir. Bilgisayarlı görmeyi kapsayan bir üst modeldir. Robotun güvenli bir şekilde engellerden kaçınması, hedef tanınması, yol planlaması gibi birçok karmaşık görevi eş zamanlı olarak yerine getirmek için bilgisayarlı görme, faydalanılan alanlardan sadece biridir.

Tez çalışmasında dinamik ortamlarda hareket edebilen gezgin robotlar için bilgisayarlı görme ile engelden kaçınma üzerine benzetim çalışmaları yapılmış ve sonuçları gözlemlenmiştir.

4. MATERYAL VE METOT

Robot ile görmede hız unsuru çok önemlidir. Görüntü işleme sürecinde yapılan işlemlerin sisteme aşırı yük bindirmeden en kısa sürede tamamlanması gerekir. Robot hareket ederken alınan görüntülerin işlenmesinde gecikme yaşanması durumunda, çarpışma ya da yanlış yöne ilerleme gibi sonuçlar ortaya çıkacaktır. Robotun engellerden kaçınarak ilerlemesi için zemin alanını doğru tespit etmesi gerekir. Robotun verimli hareketi için görüntüdeki nesneleri tanımak yerine (duvar, kapı, masa vb.) zemini tanımlayabilmek ve tanımladığı zemin üzerinde hareket etmek görüntü işleme açısından daha verimlidir. Nesneleri tanımak hem maliyet hem de zaman açısından daha zor bir işlemdir. Ayrıca tüm nesneleri tanımlamak da mümkün değildir. Zemin tanımlamak ise görüntü bölütleme ile doğrudan ilgilidir. Görüntü bölütleme için en iyi algoritma maalesef yoktur. Uygulama amacına göre seçilmesi ve denenmesi gerekir. Robotun hareketli, ortamında dinamik olduğu bir durumda zaman/maliyet açısından da algoritmaların değerlendirilmesi ve gerekli iyileştirmelerin yapılması gerekir. Çalışmada önerilen yaklaşım robot ile görmede gereksinimleri karşılayacak şekildedir. Kamera ile alınan görüntülerin eş zamanlı ve hızlı şekilde işlenmesi için iki kademeli bir yöntem önerilmiştir.

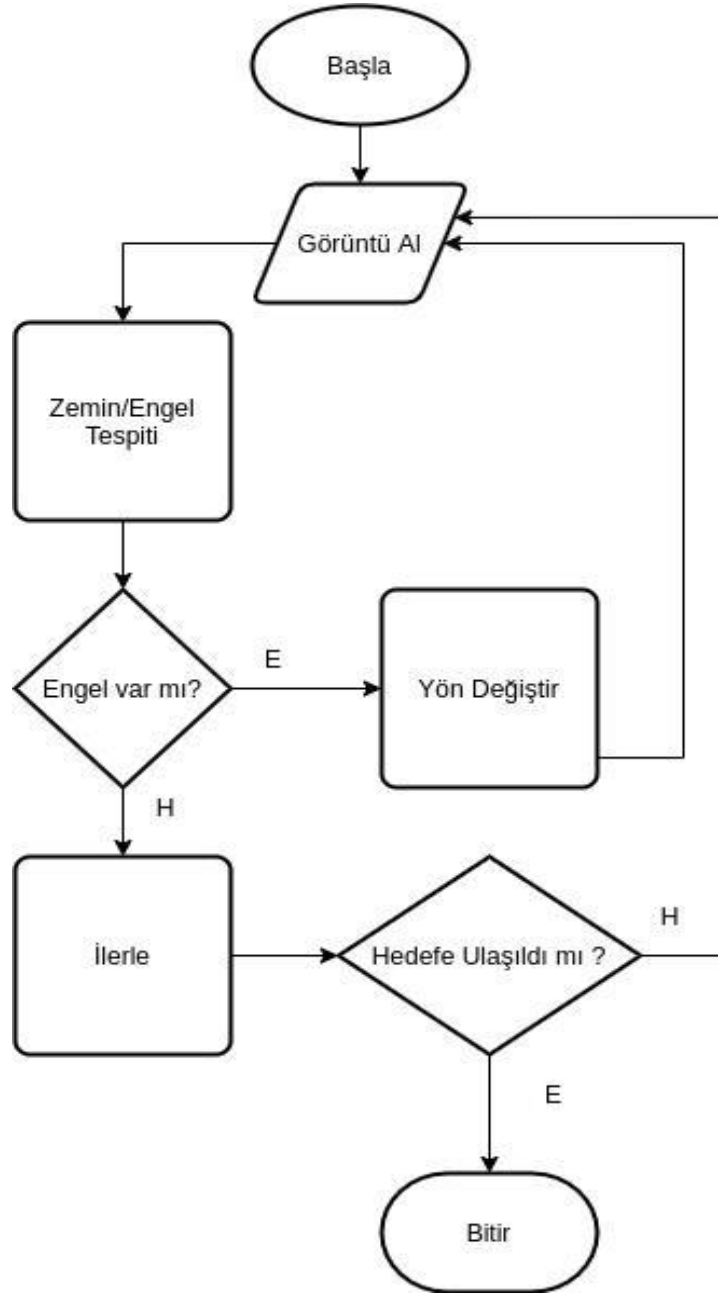
Algoritmanın odak noktası, görüntüde bütünsel bir değerlendirme yapmadan önce, örnek bir zemin alanı seçerek algoritmayı bu zemin alanını baz alarak işletmektir. Uygulamada bu konu açıklanmıştır.



Şekil 4.1: Robotun görüntü alması ve hareketi

Şekil 4.1’de belirtildiği gibi, robot öncelikle görüntü alır. İkinci adımda alınan görüntü işlenir. Strateji kutusuyla belirtilen üçüncü adımda ise görüntüden çıkarılan anlama göre, hareketsiz kalabilir, sağa ya da sola dönebilir veya hareket edebilir. Strateji adımında farklı sensörlerden gelen bilgiler de değerlendirilebilir. Ancak çalışma konusu itibari ile sadece görüntüden çıkarılan anlama göre, bir sonraki duruma geçilmesi planlanmıştır.

Şekil 4.2’de belirtildiği gibi görüntü alındıktan sonra her bir görüntü çerçevesi için zemin engel tespiti yapılır. Zemin engel tespit işleminde sürecin hızlı ilerlemesi için, robotun mevcut durumda ilerlediği yol üzerinde engel olup olmadığı kontrol edilir. Engel yoksa farklı bir işlem yapılmadan ilerlemeye devam edilir. Engel varsa görüntü işlemeye devam edilir ve görüntünün sağ ve sol yanlarından hangisinin ilerlemek için daha uygun olduğuna bakılır. Hangi tarafta zemin daha fazla görüntülenebiliyorsa o tarafa dönülür.



Şekil 4.2: Robotun görüntü alarak işlemesi ve karar süreci

4.1 Çalışma Ortamının Hazırlanması

Bu çalışmada Ubuntu işletim sistemi (16.04 LTS) üzerinde ROS Kintetic sürümü tercih edilmiştir. ROS Kintetic ile uyumlu olan Gazebo 7.16 versiyonu, benzetim ortamı için kullanılmıştır. Görüntü işleme ile zemin/engel tespiti yaparak serbest gezinme amacıyla Turtlebot robotundan faydalanılmıştır.

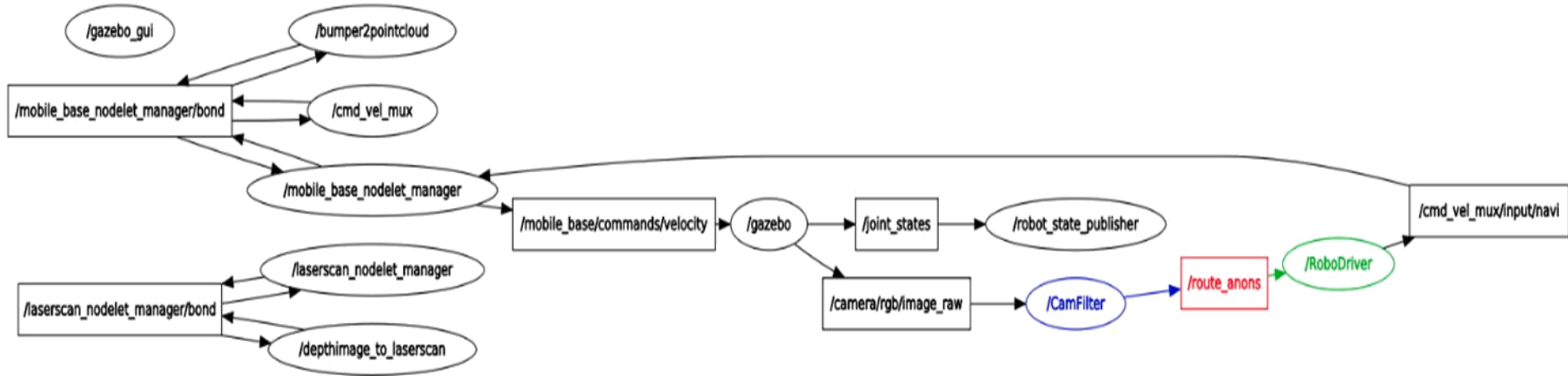
Gazebo benzetim ortamı için, indoor.world dosyası oluşturulmuştur. ROS başlatıldığında genel parametreler alabilmektedir. Turtlebot robotuna ortam olarak bu dosyanın yolu verilmiştir.

ROS genel parametrelerle çalıştırıldıktan sonra, görüntü işleme algoritmasını değiştirmek için ilgili script durdurulup diğer algoritma scriptinin değiştirilmesi yeterlidir.

4.2 Sistem Tasarımı

Şekil 4.3'te sunulduğu gibi robot kamerası üzerinden görüntü almak ve işlemek için “/camera/rgb/image_raw” yayınına “/CamFilter” düğümüyle abone olunmaktadır. İşlenen görüntü robot gezinmesini sağlayan “roboDriver” düğümüne hız ve yön bilgisi yayınlamaktadır. Bu bilgi Bölüm 2’de detayları açıklanan ROS mesajlarıyla sağlanmaktadır. “roboDriver” düğümüne her görüntünün işlenmesinden sonra veri aktarımı için “routeParams” yayını oluşturulmaktadır. Bu yayın da yine “mobile_base_commands/velocity” düğümüne ulaşmakta ve robotun hareketi sağlanmış olmaktadır. Bu esnada görüntü değişmekte ve her yeni görüntü için aynı süreç işletilmektedir. RouteParams.msg dosyası iki değişkene sahiptir. Birincisi float64 tipinde “linear” hız değişkenidir. Diğeri ise yine aynı tipte “angular” yani dönme açısı değişkenidir.

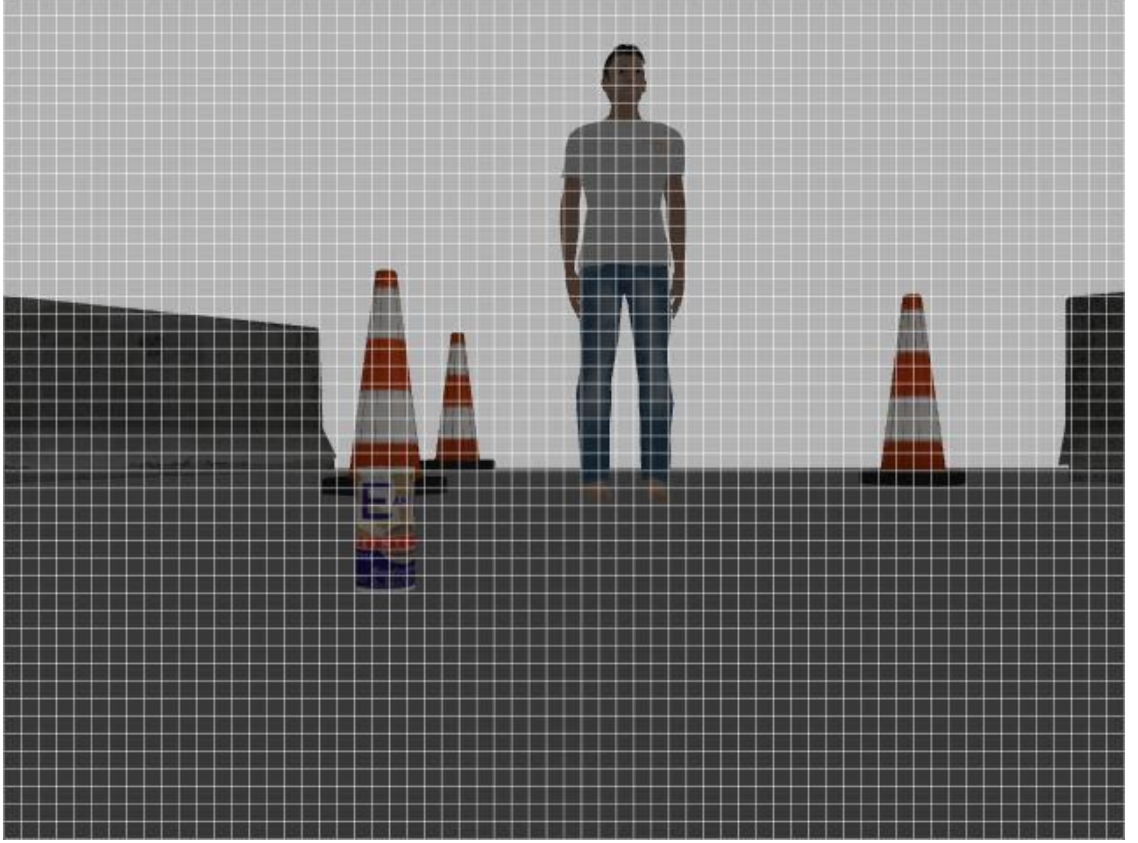
Kamera üzerinden görüntü almak için “/camera/rgb/image_raw” yayınına abone olunmaktadır. Aynı zamanda “RouteParams” yayını oluşturulmakta ve bu yayın ile “RouteParams” mesajı yayınlanmaktadır. “/camera/rgb/image_raw” yayını ile gelen her görüntü anlık olarak işlenmekte ve hareket için karar verilmektedir.



Şekil 4.3: Uygulama üzerindeki ROS düğüm yayın ve abonelikleri

4.3 Eşik Değeri ile Filtreleme ve Engelden Sakınım

Görüntü bölütleme, görüntü işlemede ilk ve en önemli adımlardan biridir. Bu işlemde, bir görüntü birden çok parçaya bölünür. Işık, gölge ve desenler, zemin algılamada piksel tabanlı yaklaşım üzerinde önemli bir etkiye sahip faktörler olduğundan, görüntü parçalara bölünür ve daha sonra işlenir. Şekil 4.4'te gösterildiği gibi görüntülere ızgara tabanlı bölütleme yapılmıştır (64×48). Görüntünün alt orta kısmından alınan parça örnek zemin alanı kabul edilmiştir.



Şekil 4.4: Bölütlenmiş benzetim ortamı görüntüsünden bir kesit (64×48).

Her bir alandaki aritmetik ortalama o alanın değeri olarak yazılır. Aşağıda eşik değeri uygulayarak filtre matrisinin oluşturulması işlemine ait denklemler sunulmuştur. Sırası ile Denklem 4.1'de n piksel sayısını, hesaplanan A değeri ise, görüntü üzerindeki her bir kesitin piksellerinin toplamının ortalamasıdır. Denklem 4.2'de hesaplanan p değeri ise, zemin kabul edilen alanın ortalama değeri ile mevcut Denklem 4.1'de hesaplanan A değeri ile yüzdesel farkını ifade eder. Denklem 4.3'te, FM filtrelenmiş matrisi, i ve j matrisin boyutlarını, T değeri ise uygulanabilir eşik değerini ifade eder. Denklemden daha önceden belirlenen yüzdesel değişim için eşik değeri olan T değeri ile p değeri

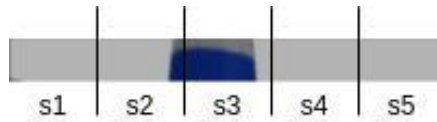
karşılaştırılmakta ve filtre matrisinin değerleri bu denklemlere göre '0' ya da '1' olarak verilmektedir.

$$= \frac{1}{\sum} \quad (4.1)$$

$$= \frac{| - () |}{() \times 100} \quad (4.2)$$

$$= \begin{cases} 0, & \text{g} \geq \\ 1, & \text{g} < \end{cases} \quad (4.3)$$

Bölütlerin genel ortalamasının, zemin olarak belirlenen bölütün ortalamasına yüzdesel olarak uzaklığı, belirlenen yüzdesel eşik değerinden eşit ya da yüksekse sıfır, düşük ise bir değeri verilir ve filtrelenmiş matris elde edilir. Filtrelenmiş matris varyans ya da standart sapma kullanılarak da elde edilebilir. Ancak farklı renk yoğunluklarında birbirine çok yakın pikseller için bu değerler, aritmetik ortalamaları farklı olmasına rağmen sıfıra yaklaşmaktadır. Şekil 4.5'te bir görüntüden alınan kesit 5 parçalık piksel kümelerine bölünmüştür.



Şekil 4.5: Bölütlerde renkler

Çizelge 4.1'de aritmetik ortalamanın etkisi gösterilmiştir. Çizelgede S2, S4, S5 bölütlerinin varyans, standart sapma ve aritmetik ortalamaları aynıdır. Fakat S bölütü, standart sapma ve varyans değerlerinde S4, S5 bölütlerine yaklaşık bir değerkken, aritmetik ortalaması çok farklıdır. Bunun sebebi, homojen alanlarda standart sapma ve varyansın sıfır olmasıdır. Fakat aritmetik ortalama sıfırdan farklı bir değer olmaktadır.

Çizelge 4.1: Varyans, aritmetik ortalama ve standart sapma için değerler

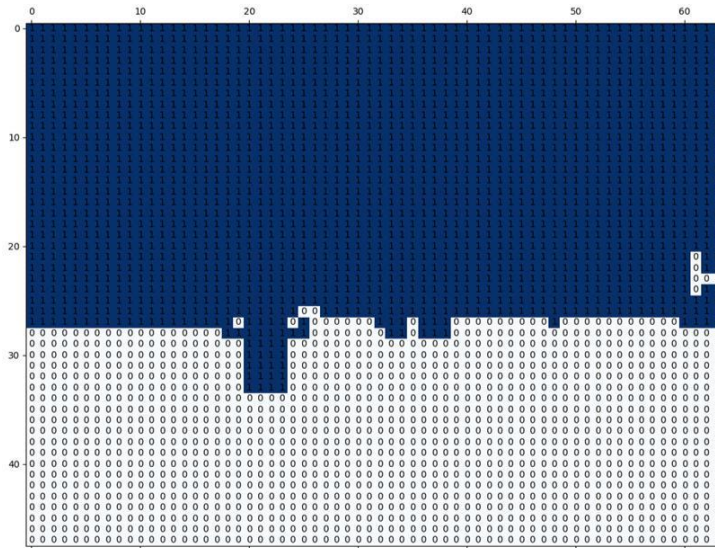
Metot	S	S1	S2	S3	S4	S5
Varyans	0.0	289	805	1407	0	0
Aritmetik Ort.	56	176	168	115	179.0	179.0
Standart Sapma	0.0	17	28	37	0	0

Dolayısı ile görüntü işlemede kullanılan varyans ve standart sapma değerleri, zemin ve diğer nesneler arasında ayırım yapmak için yetersiz olacaktır. Bu nedenle çalışmada AOYD önerilmiştir.

Aşağıda filtreleme algoritması verilmiştir.

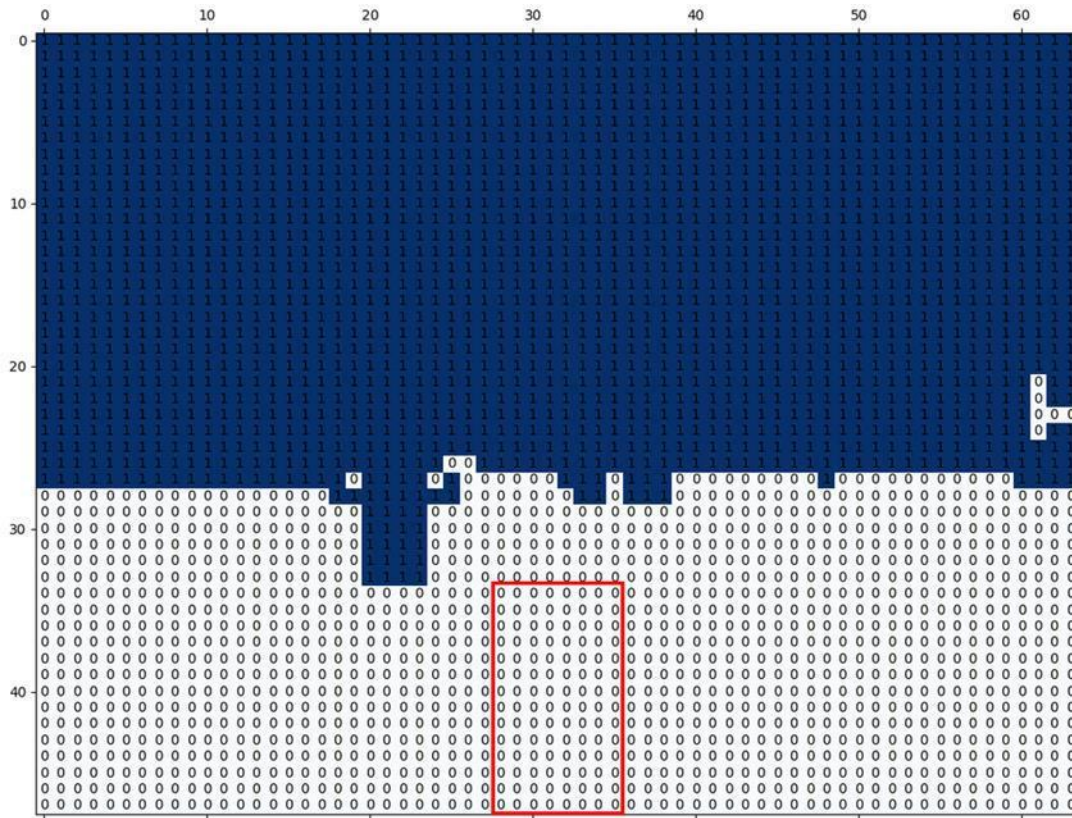
```
Girdi: Matrix M (Kameradan Alınan Görüntü)
Çıktı: FiltreMatrisi FM
s <-- ornekZeminAlaniniAl(M)
t <-- Değişim yüzdesi eşiği
v <-- Referans varyans değeri satır
<-- matrisSatirSayisiniAl(M) sutun
<-- matrisSutunSayisiniAl(M) for
each i=0 to satir do
  for each j=0 to sutun do
    m=mevcutParcaninOrtalaması(M,i,j)
    p=degisimYuzdesiniHesapla(s,m)
    eğer t => p
      FM[i][j] =
    0 değilse
      FM[i][j] = 1
    end
  end
end
eğer FM.varyans <= v
  esikDegeriniGuncelle(t=t+t*0.1)
return FM
```

Yukarıda verilen algoritma uygulandığında Şekil 4.6'daki gibi, 64×48 boyutlarında filtre matrisi elde edilmektedir.



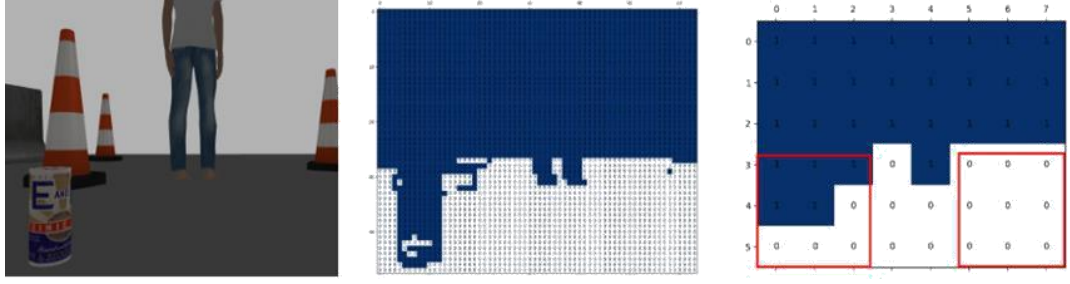
Şekil 4.6: Filtre matrisi

64×48’lik filtrelenmiş görüntüde alt merkezin ortasından her iki yana doğru 4 birimlik ve matrisin son satırından geriye doğru 14 birimlik dikdörtgen alan seçilir. Seçilen alanın büyüklüğü robot büyüklüğü ile ilintilidir. Görüntüye robotun eşdeğeri bir robot eklenerek ne kadar alan kapladığı gözlenmiş ve buna göre boyutlar belirlenmiştir. Seçilen alanın toplamının sıfıra (zemin kabul edilen bölgelere) eşit olup olmadığına bakılır. Eğer toplam sıfıra eşitse, robot hareket etmeye devam eder ve bir sonraki alınan görüntü için aynı işlemi uygular. İşlemler yinelemeli olarak devam edeceği için engel yoksa robot ilerlemeye devam edecektir.



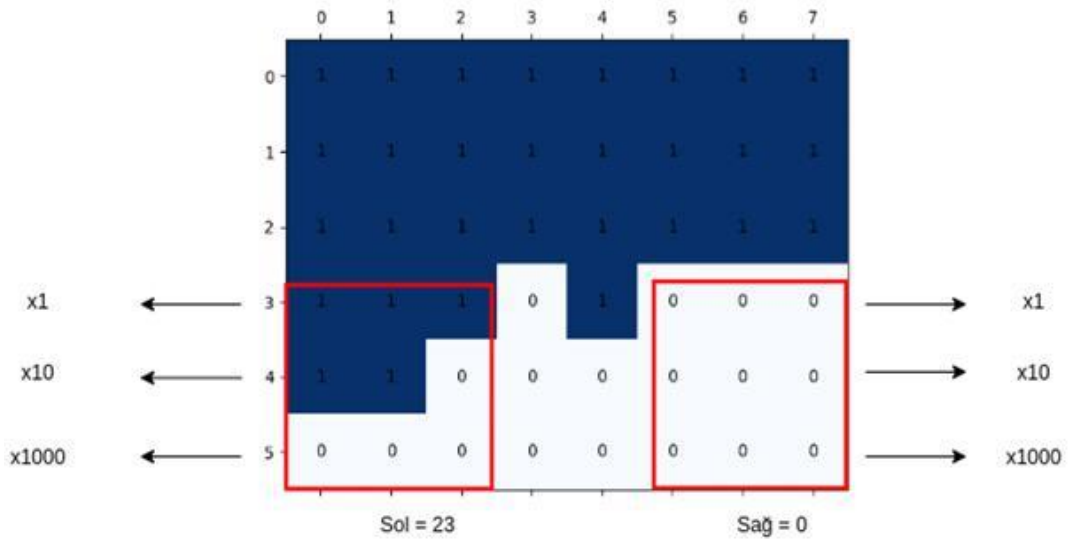
Şekil 4.7: Filtrelenmiş matris üzerinde engel kontrolü

Eğer toplam sıfırdan farklı ise, bu bir engel olduğu anlamına gelir. Şekil 4.7’de bu duruma bir örnek verilmiştir. Böyle bir durumda 64×48’lik matris aynı filtreleme mantığı ile tekrar filtrelenir. 8×6 boyutlarında yeni bir hareket karar matrisi elde edilir. Bu matrisin sağ ve sol kenar toplamaları ağırlıklı yönteme göre toplanır. Hangi toplam değeri düşükse o tarafta daha az engel olduğu varsayılır ve robota dönme komutu verilir. İşlemler bu şekilde yinelemeli olarak devam eder.



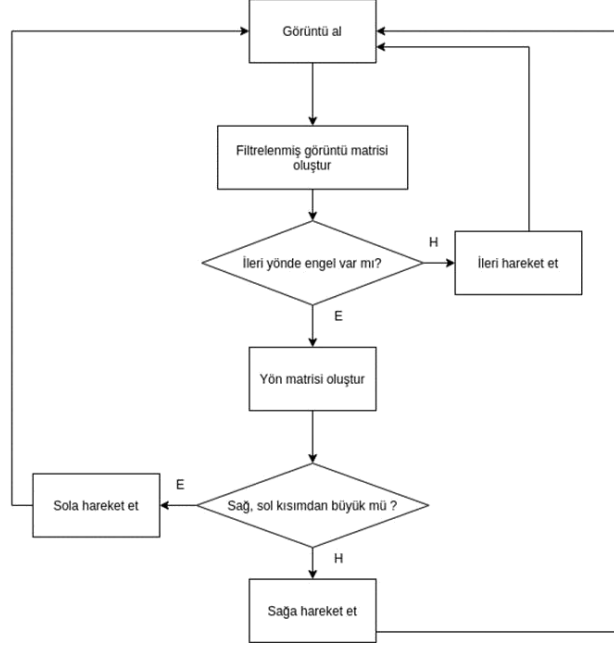
Şekil 4.8: Benzetim görüntüsü, filtre matrisi ve yön matrisi

Şekil 4.8’de sol tarafta daha az zemin alanı fark edilmektedir. Orijinal görüntüdeki kutunun filtre matrisinde kapladığı alan görülmektedir. Ön kısımda engel algılandığı için hareket karar matrisinde sağ ve sol kenar toplamaları hesaplanmıştır. Sağ kenar toplamı soldan küçük olduğu için robot belirli bir açıda sağa dönüş hareketi yapacaktır.



Şekil 4.9: Yön matrisi ve sağ-sol kenar kontrolü

Şekil 4.9’da sağ kırmızı çerçeve içinde 3 satır bulunmaktadır. En üst satır, önemi en az olan satırdır. Bu satırdaki rakamların toplamı 1 ile çarpılır. Bir altındaki satırdaki rakamların toplamı 10 ile ve en alt satırdaki rakamların toplamı da 100 ile çarpılır ve tamamı toplanır. Aynı işlem sol kırmızı çerçeveye de uygulanır. Sağ ve sol alanların toplamı hesaplanırken 1, 10, 100 gibi katsayıların kullanılmasının sebebi, yakın olan engelin daha yüksek puana sahip olması gerektiridir. Hareket için bir tercih yapılırken yakından uzağa doğru mümkün olan en fazla zemin yüzeyine sahip yön seçilir. Sağ ve sol toplamlar karşılaştırılarak düşük olan yöne hareket sağlanır. Sağa ve sola hareketlerde robotun ileri yönlü hareketi de devam eder. Eşzamanlı olarak hem görüntü işlemeye devam edilir hem de engellere çarpmadan ilerlemeye devam edilir.



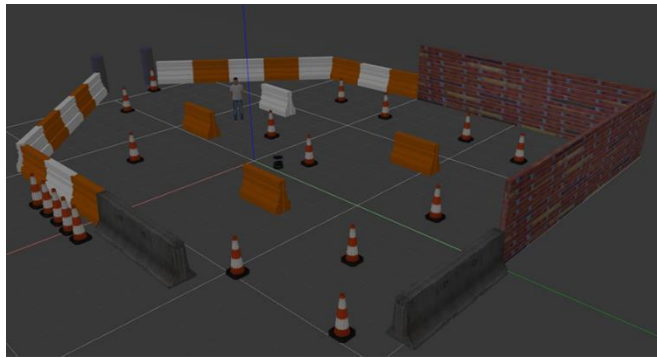
Şekil 4.10: Sağ-sol hareket akış şeması

Robotun görüntü almasından hareket aşamasına kadar olan akış diyagramı Şekil 4.10’da gösterilmiştir.

4.4 Uygulama

Deney ortamı için i7/1.8 GHz işlemcili bir bilgisayar kullanılmıştır. ROS ve Gazebo, kurulumu yapılmış ve Turtlebot model robotu bu ortama entegre edilmiştir.

Benzetim ortamı olarak kullanılan Gazebo platformuna, düz ve açılı devam eden kenar duvarlarla birlikte, değişik ebatlarda engeller eklenmiştir. Şekil 4.11’de benzetim ortamından örnek bir kesit verilmektedir.



Şekil 4.11: Benzetim ortamı

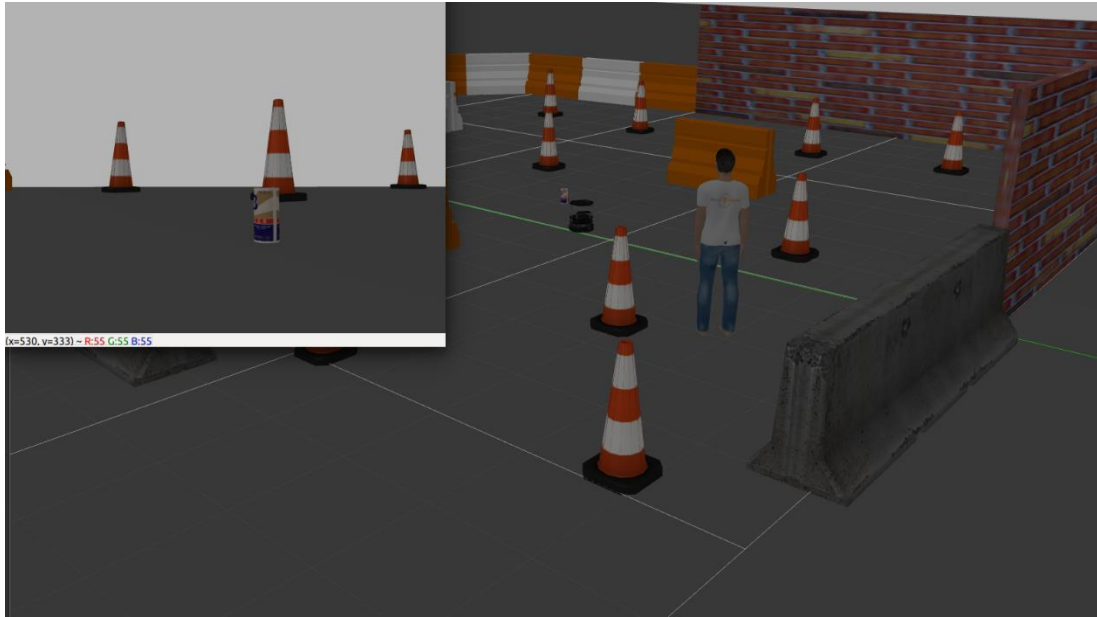
Benzetim ortamında kullanılan robota ait kamera parametreleri ve ROS parametreleri Çizelge 4.2’de verilmiştir.

Çizelge 4.2: Benzetim Ortamı Parametreleri

Parametre	Açıklama
Kamera Çözünürlüğü	640×480
Resim Sıklığı	20 fps
Robot Hızı	0.25 m/s
Dönüş Açısı	0 m/s
Engel Durumunda Robot Hızı	0.15 m/s
Engel Durumunda Dönüş Açısı	0.40 rad/s
Bölüt Boyutu	10×10

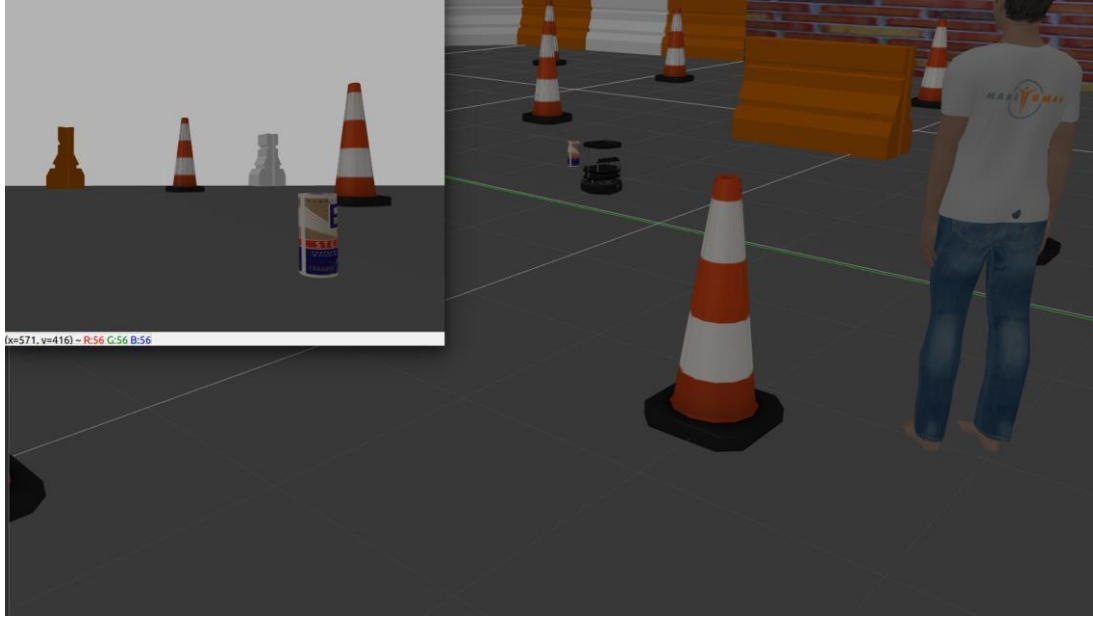
Robotun bulunduğu kapalı ortamın özelliklerine göre, bölüt boyutları değiştirilebilir. Zeminin desenli olduğu görüntülerde bölüt boyutları arttırılabilir. Düşük olduğu durumlarda ise, daha fazla zemin alanı keşfetmek için bölüt boyutları düşürülebilir.

Şekil 4.12’de robot ortamda düz bir şekilde ilerlemektedir. Genel görüntüdeki robotun kamera görüntüsü şeklin sol üst köşesine atılanmıştır. Robot bir süre ilerledikten sonra önündeki engeli algılayacak ve engelin solundan devam edecektir.



Şekil 4.12: Benzetim ortamı robot kamerası görüntüleri (I)

Robotun ilerlemesinde ikinci aşamada engele iyice yaklaşmış bulunmaktadır. Şekil 4.13'te bu durum görülmektedir.



Şekil 4.13: Benzetim ortamı ve robot kamerası görüntüleri (II)

Şekil 4.14'te engelin solunda geçilerek robot ilerlemeye devam etmiştir.



Şekil 4.14: Benzetim ortamı ve robot kamerası görüntüleri (III)

Şekil 12,13 ve 14'te görüldüğü gibi robot zemini algılamakta ve önüne çıkan engelin görüntüdeki pozisyonuna bağlı manevra yapmakta ve engele çarpmadan sağından veya solundan geçmektedir. Birden fazla engelin bulunduğu ortamda robot engellere ve duvarlara çarpmadan hareket edebilmekte ve seyrini sürdürmektedir.

5. TARTIŞMA VE SONUÇ

Bilgisayarların daha hızlı işlem yapması, boyutlarının küçülmesi ve donanım kapasitelerinin artmasıyla birlikte, görüntü işleme ve bir alt kümesi olan bilgisayarlı görme alanlarında yapılan çalışmalar da artmıştır. Bilgisayarlı görme robotların belirli görevleri yerine getirmesi için kullanılan temel teknolojilerden biridir. Kara, deniz ve hava ortamlarında çeşitli görevleri yerine getirmesi için tasarlanan robotlarda oldukça sık kullanılmaktadır. Kara araçlarında tasarımlar dış ve iç ortam özelliklerine göre değişmektedir. Çalışmamızda kapalı ortamlar için geliştirilen robotların zemin algılaması ve engelden sakınımı için bir yöntem önerilmiştir.

Kapalı ortamlar için geliştirilen robotların önündeki en önemli sorunlardan biri ortamın değişmesidir. Ortamın bir haritasını tutan yöntemler, ortam değiştiğinde kullandıkları haritaları güncellemek durumundadır. Haritaların güncellenmesinde, oluşturulduğu yöntemlere göre çeşitli hatalar oluşmakta ve güncelleme işlemi zaman almaktadır. Bilgisayarlı görme ise dinamik ortamlar için elverişlidir. Gerçek zamanlı görüntü işleme, nesne tanıma ve öğrenme algoritmaları ile desteklenerek, kapalı ortamlarda verimli şekilde hareket eden robotlar geliştirilebilir.

Robot hareketi için geliştirilen algoritmaların bir kısmı tamamen statik ortamlar içindir. Bu durum güncel kullanım için sınırlamalar getirmektedir. Dinamik ortamlar için tasarlanan sistemlerde ise; dinamik ortam robotun hareketi için kullanılabilir. Öğrenme tabanlı algoritmaların kararsızlığını arttırmaktadır. Robot belli bir öğrenmeyi tamamladığında eğer bulunduğu ortamda değişiklikler olduysa kazanmış olduğu yeteneği devam ettirememektedir.

Tez çalışmasında gezgin robotların dinamik ortamlarda engelden sakınarak gezinmesi için, kamera dışında herhangi bir sensör kullanmadan, gerçek zamanlı görüntü işlemeye dayalı iki aşamadan oluşan yeni bir yöntem sunulmuştur. Sunulan yöntem aynı zamanda benzetim ortamında test edilmiştir. Yöntemin birinci aşamasında robotun bulunduğu ortamdan örnek bir zemin görüntüsü alınmakta ve görüntünün diğer alanları ile karşılaştırılarak zemin alanı ortaya çıkarılmaktadır. Bu işlemde yüzdesel bir değer eşik değeri olarak kullanılmıştır. Birinci aşama sonunda zemin alanına göre filtrelenmiş matris elde edilmektedir. İkinci aşamada ise birinci aşamada elde edilen görüntü tekrar işlenmekte ve daha düşük boyutlu yön matrisi elde

edilmektedir. Robotun gezinmesi, elde edilen matrislerin deęerlendirmesine baęlı olarak gerekleřtirilmektedir.

alıřmanın odak noktası, zemin tespiti yapmak ve grntde zemin dıřında farklı bir nesne algılanması durumunda ki bu nesneler engel olarak iřaretlenmektedir, engellerden sakınarak gezinmektir. alıřmada gerekleřtirilen algoritmalar ROS zerinde test edilmiř ve benzetim ortamı olarak kullanılan Gazebo platformunda robotun hareketleri gzlenmiřtir. Bylece ne srlen yntemin gerek dnyada da uygulanabileceęi gsterilmiřtir.

Kamera ile alınan grntlerin gerek zamanlı iřlenmesi zerine, AOYD ve eřikleme yntemi uygulanmıřtır. Alınan grntler bltlenerek bir eřik deęeri ile filtreleme iřlemi yapılmıřtır. Filtreleme iřleminin odak noktası; ortamdaki zemin bltlerinin dięer bltlerden ayrılması zerinedir. Benzetim ortamında elde edilen sonular, robotun engellerden bařarılı bir řekilde sakındıęını ve sistemin kapalı ortamlarda kullanılabileceęini gstermektedir.

Robotun gvenli bir řekilde engellerden kaınması, hedef tanınması, yol planlaması gibi birok karmařık grevi eř zamanlı olarak yerine getirmesi iin olduka karmařık algoritmalar gerekmektedir. ok fazla sayıda grevin birbiri ile akıřmadan yrtlebilmesi iin, hızlı sonular reten yntemler tercih edilmektedir. alıřmamızda sadece bilgisayarlı grmeden anlam ıkarma deęil, aynı zamanda robotun hareketine karar verebilecek bir yntem nerilmiřtir. Robot hareketi iin farklı algoritmalarla birlikte kullanılabilecek bir etkili bir yntemdir.

KAYNAKLAR

- [1] D. B. Camarillo, T. M. Kurummel, Robotic technology in surgery: Past, present, and future, *The American Journal of Surgery*. 188 (2004) 2-15.
- [2] Duckett, T.; Pearson, S.; Blackmore, S.; Grieve, B.; Chen, W.H.; Cielniak, G.; Cleaversmith, J.; Dai, J.; Davis, S.; Fox, C.; et al, *Agricultural robotics: The future of robotic agriculture, Uk-Ras Network Robotics and Autonomous System*. (2018) 1-36.
- [3] N. Noguchi, Oscar C.Barawid, *Robot Farming System Using Multiple Robot Tractors in Japan Agriculture, IFAC Proceedings Volumes*. 44 (2011) 633-637.
- [4] A. Flores, O. Ma, K. Pham, S. Ulrich, *A review of space robotics technologies for on-orbit servicing, Progress in Aerospace Sciences*. 68 (2014) 1-26.
- [5] P. S. Sapaty, “Military Robotics: Latest Trends and Spatial Grasp Solutions, *International Journal of Advanced Research in Artificial Intelligence*. 4 (2015) 9-18.
- [6] Tang, B.; Jiang, C.; He, H.; Guo, Y. *Human mobility modeling for robot-assisted evacuation in complex indoor environments. IEEE Trans. Hum. Mach. Syst.* 46 (2016) 694–707.
- [7] *How Robots are Changing Logistics*, distrelec.de/current/en/robotics/how-robots-are-changing-logistics/, 2018.
- [8] *Types of Robot Sensors*, robotplatform.com/knowledge/sensors/types_of_robot_sensors.html, 2020.
- [9] H. Everett, *Sensors for Mobile Robots Theory and Application*, San Diego, California, 1995.
- [10] J. Borenstein H. R. Everett L. Feng D. Wehe, *Mobile robot positioning: Sensors and techniques, Special Issue on Mobile Robots*. 14 (1997) 231-249.
- [11] Y. Zhao, L. Gong, Y. Huang, C. Liu, *A review of key techniques of vision-based control for harvesting robot, Computers and Electronics in Agriculture*. 127 (2016), 311-323.

- [12] N. Herakovic, Robot Vision in Industrial Assembly and Quality Control Processes, in Robot Vision, 2010, pp. 501-534.
- [13] H. Surmann, A. Nüchter, J. Hertzberg, An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, Robotics and Autonomous Systems, 45 (2003) 181–198.
- [14] S. Kim, H. Kim, Optimally overlapped ultrasonic sensor ring design for minimal positional uncertainty in obstacle detection, Int. J. Control, Automation and Systems, 8 (2010) 1280–1287.
- [15] A. Al-Kaff, F. García, D. Martín, A. Gomez, A. De La Escalera, J. Armingol, Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs, Sensors, 17 (2017) 1061.
- [16] H. Alvarez, L. M. Paz, J. Sturm, D. Cremers, Collision avoidance for quadrotors with a monocular camera, In Experimental Robotics, Springer, Cham., 2016, pp. 195–209,
- [17] K. McGuire, G. De Croon, C. De Wagter, K. Tuyls, H. Kappen, Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone, IEEE Robotics and Automation Letters, 2 (2017) 1070–1076.
- [18] J. J. Lee, M. Shinozuka, A vision-based system for remote sensing of bridge displacement, NDT & E International. 39 (2006) 425-431.
- [19] N. M. Zaitoun, M. J. Aqel, Survey on image segmentation techniques, Procedia Computer Science. 65 (2015) 797–806.
- [20] N. Ikonomakis, K. N. Plataniotis, A. N. Venetsanopoulos, Color image segmentation for multimedia applications, Journal of Intelligent and Robotic Systems. 28 (2000) 5–20.
- [21] H. Zhang, X. Wang, Y. Chen, G. Jiang, S. Lin, Research on Vision-Based Navigation for Plant Protection UAV under the Near Color Background, Symmetry. 11 (2019) 533.
- [22] S. J. Dumble, P. W. Gibbens, Horizon profile detection for attitude determination, Journal of Intelligent & Robotic Systems. 68 (2012) 339–357.

- [23] W. Benn, S. Lauria, Robot navigation control based on monocular images: an image processing algorithm for obstacle avoidance decisions, *Mathematical Problems in Engineering*. (2012) 1-14.
- [24] S. A. Li, L. H. Chou, T. H. Chang, C. H. Yang, Y. C. Chang, Obstacle Avoidance of Mobile Robot Based on HyperOmni Vision, *Sensors and Materials*. 31 (2019) 1021–1036.
- [25] V. S. Kalogeiton, K. Ioannidis, G. C. Sirakoulis, E. B. Kosmatopoulos, Real-Time Active SLAM and Obstacle Avoidance for an Autonomous Robot Based on Stereo Vision, *Cybernetics and Systems*. 50 (2019) 239–260.
- [26] W. Benn, S. Lauria, Robot navigation control based on monocular images: an image processing algorithm for obstacle avoidance decisions, *Mathematical Problems in Engineering*, 14 pages, 2012.
- [27] S. Bhowmick, A. Pant, J. Mukherjee, A. K. Deb, A novel floor segmentation algorithm for mobile robot navigation, In 2015 Fifth National Conference on Computer, pp. 1-4, 2015.
- [28] C. Chun, D. Park, W. Kim, C. Kim, Floor detection based depth estimation from a single indoor scene, In 2013 IEEE International Conference on Image Processing, pp. 3358–3362, 2013.
- [29] Y. Li, S. T. Birchfield, Image-based segmentation of indoor corridor floors for a mobile robot, In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 837–843, 2010.
- [30] M. Ling, W. Jianming, Z. Bo, W. Shengbei, Automatic floor segmentation for indoor robot navigation, In IEEE International International Conference on Signal Processing Systems, vol. 1, pp. 684–689, 2010.
- [31] S. Benabid, FPGA-Based Real-Time Embedded Vision System for Autonomous Mobile Robots, *European Union*. (2019) 1093-1096.
- [32] S. Mane, S. Vhanale, Real Time Obstacle Detection For Mobile Robot Navigation Using Stereo Vision, 2016 International Conference on Computing, Analytics and Security Trends (CAST) College of Engineering Pune, India. Dec 19-21, pp. 637-642, 2016.

- [33] About ROS, <https://www.ros.org/about-ros/>, erişim tarihi Şubat 2020.
- [34] Gazebo, <http://gazebo-sim.org/>, erişim tarihi Şubat 2020.
- [35] Turtlebot, <https://www.turtlebot.com/partners/>, erişim tarihi Şubat 2020.
- [36] Radu Adrian, C. M. Simion, Industrial Applications of Image Processing, *Acta Universitatis Cibiniensis. Technical Series.* 64 (2014), 17-21.
- [37] S. Kumar, Importance & Applications of Digital Image Processing, *International Journal of Computer Science & Engineering Technology (IJCSET).* 7 (2016) 316-320.
- [38] B. H. Kang, A Review on Image & Video Processing,, *International Journal of Multimedia and Ubiquitous Engineering.* 2 (2007) 49-64.
- [39] M. Ali, Clausi, D, Using The Canny Edge Detector for Feature Extraction and Enhancement of Remote Sensing Images, *IEEE.* (2001) 2298-2300.
- [40] W. Gao, L. Yang, X. Zhang, H. Liu, An improved Sobel edge detection, *IEEE.* (2010) 667-71.
- [41] S. S. Al-amri, N. V. Kalyankar, S. D. Khamitkar, Image Segmentation By Using Edge Dedection, *International Journal on Computer Science and Engineering.* 2 (2010) 804-807.
- [42] S. Paris, S. W. Hasinoff, J. Kautz, Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid, *Communications of the ACM.* 58 (2011) 1-11.
- [43] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man, And Cybernetics,* 9 (1979) 62-66..
- [44] P. K. Sahoo, S. Soltani, A. K. C. Wong, “A Survey of Thresholding Techniques, *Computer Vision, Graphics, and Image Processing.* 41 (1988) 233-260.
- [45] M. Hassaballah, K. M. Hosny, Recent Advances in Computer Vision: Theories and Applications, Springer International Publishing, Springer Nature Switzerland AG, 2019.
- [46] S. Yuheng, Y. Ho, “Image Segmentation Algorithms Overview”, *ArXiv, abs/1707.02051,* 2017

[47] Image Segmentation

www.cse.iitm.ac.in/~vplab/courses/CV_DIP/PDF/lect-Segmen.pdf, Nisan 2020

[48] D. T. Pham, S. S. Dimov, C. D. Nguyen, “Selection of K in K-means clustering”, Proceedings of The Institution of Mechanical Engineers Part C- journal of Mechanical Engineering Science - PROC INST MECH ENG C-J MECH E. 219. 103-119.

[49] M. F. Esen, E. Bilgiç, “Topsis Ve K-Medoids Yöntemleriyle Karar Verme: R Programlama Dili İle Müşteri Geri Kazanma Uygulaması”, Yönetim ve Ekonomi Araştırmaları Dergisi, Yıl: 2018 Sayı: 16 p.221-237

[50] W. Cherif, “Optimization of K-NN algorithm by clustering and reliability coefficients: application to breast-cancer diagnosis”, Procedia Computer Science, Volume 127, 2018, Pages 293-299

[51] M. Işık A. Yılmaz Çamurcu, “K-Means, K-Meoids ve Bulanık C-Means Algoritmalarının Uygulamalı Olarak Performanslarının Tespiti”, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl: 6 Sayı:11Bahar 2007/1 s. 31-45

ÖZGEÇMİŞ



Adı Soyadı: Adem HİÇDURMAZ

Doğum Yeri ve Tarihi: Erzurum / 03.05.1978

E-posta: ademhicdurmaz@hotmail.com

Lisans: Ahmet Yesevi Üniversitesi, Bilgisayar Mühendisliği

TEZDEN TÜRETİLEN YAYINLAR

- [1] A. Hiçdurmaz, A. Tuncer, “Real-Time Obstacle Avoidance Based on Floor Detection for Mobile Robots”, Sakarya University Journal of Science, (Kabul edildi.)