



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



SLAM ve vSLAM ALGORİTMALARININ
İNCELENMESİ, YENİ BİR ÇOKLU ROBOT
HARİTA BİRLEŞTİRME YÖNTEMİNİN
GELİŞTİRİLMESİ VE UYGULANMASI

Mehmet KORKMAZ

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Haziran-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Mehmet KORKMAZ tarafından hazırlanan “SLAM ve vSLAM Algoritmalarının İncelenmesi, Yeni Bir Çoklu Robot Harita Birleştirme Yönteminin Geliştirilmesi ve Uygulanması” adlı tez çalışması 24/06/2019 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Cabir VURAL

Danışman

Dr. Öğr. Üye. Akif DURDU

Üye

Doç. Dr. S. Sinan GÜLTEKİN

Üye

Doç. Dr. Ömer AYDOĞDU

Üye

Dr. Öğr. Üye. Burak YILMAZ

İmza

Cabir Vural

Akif Durdu

S. Sinan Gültekin

Ömer Aydoğdu

Burak Yılmaz

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü

Bu tez çalışması TÜBİTAK tarafından 2214-A ve 2211-C programları kapsamında desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Mehmet KORKMAZ

Tarih: 24 Haziran 2019

ÖZET

DOKTORA TEZİ

SLAM ve vSLAM ALGORİTMALARININ İNCELENMESİ, YENİ BİR ÇOKLU ROBOT HARİTA BİRLEŞTİRME YÖNTEMİNİN GELİŞTİRİLMESİ VE UYGULANMASI

Mehmet KORKMAZ

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

Danışman: Dr. Öğr. Üyesi Akif DURDU

2019, 130 Sayfa

Jüri

**Dr. Öğr. Üyesi Akif DURDU
Prof. Dr. Cabir VURAL
Doç. Dr. S. Sinan GÜLTEKİN
Doç. Dr. Ömer AYDOĞDU
Dr. Öğr. Üyesi Burak YILMAZ**

Eş zamanlı konumlama ve haritalama (SLAM) probleminin çözümü günümüz teknolojisinde de kullanılan birçok ileri araştırmanın temellerini oluşturmuştur. Bu tez çalışmasında konunun önemine binaen klasik ve modern dönem SLAM algoritmalarının her ikisine dair incelemeler yapılmıştır. Klasik dönem SLAM algoritmaları için mevcut algoritmalar ve sensörlerin karşılaştırıldığı farklı uygulamalar yapılmıştır. Modern dönem SLAM yaklaşımları için mevcut yöntemlerin irdelendiği çalışmalara ek olarak ORB-SLAM yönteminin çıktılarının yoğunluk tabanlı haritalara dönüştürülmesi üzerine bir geliştirme yapılmış ve bir uygulama ile sonuçlar gösterilmiştir.

Bunlara ek olarak SLAM problemi komşuluğunda incelenen aktif SLAM ve harita birleştirme konuları üzerine de çalışmalar yapılmıştır. Aktif SLAM algoritmaları navigasyon kısmı için farklı iki yeni yaklaşım önerilmiştir. Çalışmanın birisinde derin öğrenme tabanlı bir metot geliştirilmiş diğerinde ise sola yönelimli olarak adlandırılan bir algoritma taslağı üzerinde durulmuştur. Geliştirilen algoritmalar bilgisayar benzetimi ve gerçek zamanlı robotlar üzerinde denenerek sonuç başarımları gösterilmiştir. Çoklu robot uygulamaları yerel haritaların birleştirilmesi problemi için mevcut yöntemlerin simetrik ortam özelliklerinin bulunduğu durumlarda başarımlarının iyi olmadığı görülmüştür. Bu eksiklik baz alınarak tez kapsamında güvenilir özellik tanımlama ve buna bağlı anlamlı harita birleştirme algoritması fikri önerilmiştir. Bu yeni yöntem, bilgisayar benzetimi ve gerçek zamanlı uygulamalarla denenmiş ve mevcut özellik tabanlı birleştirme yöntemlerine göre üstünlükleri gösterilmiştir.

Bunlara ilave olarak robot dışında insanlar üzerinde konumlama/haritalama uygulamalarının nasıl olacağı irdelenmiştir. İnsan ayağına monte edilen bir içsel ölçüm birimi (IMU) ile gerçek zamanlı denemeler yapılmış ve başlangıç noktası bilinmeyen durumlarda konumlama ve başlangıç noktasının nasıl bulunacağına dair durumlar incelenmiştir.

Anahtar Kelimeler: Aktif SLAM, Bayes tabanlı filtreler Haritalama, CNN, Harita eşleme, Harita Birleştirme, Konumlama, ROS, SLAM, vSLAM

ABSTRACT

PhD THESIS

INVESTIGATION OF SLAM AND vSLAM ALGORITHMS, DEVELOPMENT AND APPLICATION OF A NEW MAP MERGING TECHNIQUE

Mehmet KORKMAZ

**Konya Technical University
Institute of Graduate Studies
Department of Electrical-Electronics Engineering**

Advisor: Asst. Prof. Dr. Akif DURDU

2019, 130 Pages

Jury

Asst. Prof. Dr. Akif DURDU

Prof. Dr. Cabir VURAL

Assoc. Prof. Dr. S.Sinan GÜLTEKİN

Assoc. Prof. Dr. Ömer AYDOĞDU

Asst. Prof. Dr. Burak YILMAZ

The solution of the Simultaneous Localization and Mapping (SLAM) problem is the basis of many advanced pieces of research used in today's technology. Considering the importance of the topic, both classical and modern period SLAM algorithms have been investigated in this dissertation. Various applications which compare the existing techniques and different sensors have realized for classical period SLAM algorithms. As well as the investigation of the existing modern period SLAM methods, an improvement which basically transforms the ORB-SLAM technique feature maps into the density-based maps is provided and the results are demonstrated with a real-time application.

In addition to these, studies on active SLAM and map merging which are addressed in the neighborhood of SLAM have been realized. Two new approaches have been proposed for the navigation part of active SLAM algorithms. A deep learning-based method is developed for the first study and an algorithm framework called left-oriented is enhanced for the second study. The developed algorithms and their performance are examined on simulation and real-time environments. The failure of the current methods on environments which have symmetrical features has been witnessed for the map merging problem of the multi-robot. Based on this failure, a trusted feature and related semantic map merging algorithm have been proposed within the scope of this dissertation. This developed method is analyzed by simulation and real-time experiments and advantages of the method over the existing methods have been observed.

Besides all these, localization/mapping applications on humans other than robots have been examined. Real-time tests are performed with an inertial measurement unit (IMU) mounted on the pedestrian foot. The problems with unknown initial points are handled. Discussions and development on how to find the initial position and coordinate matching between IMU and the global map have been carried out by various real-time experiments.

Keywords: Active SLAM, Bayes-based Filters, CNN, Localization, Mapping, Map Matching, Map Merging, ROS, SLAM, vSLAM

ÖNSÖZ ve TEŞEKKÜR

Bu tez çalışmama katkıda bulunan danışmanım Sayın Dr. Öğretim Üyesi Akif DURDU'ya, The Ohio State University, Laboratory for Photogrammetric Computer Vision (PCVLab) yürütücüsü Prof. Dr. Alper YILMAZ ve laboratuvar üyelerine, tüm bölüm içi, bölüm dışı arkadaşlarıma Tez İzleme Komitesi üyeleri Prof. Dr. Cabir VURAL ve Doç. Dr. S. Sinan GÜLTEKİN'e teşekkürlerimi sunuyorum.

Tez çalışmam süresince 2214-A (1059B141601279) ve 2211-C (1649B031603019) programlarıyla destek sağlayarak ülkemizde bilimin gelişmesi ve yeni araştırmacıların yetiştirilmesi için çaba gösteren TÜBİTAK'a teşekkür ederim.

Ayrıca tez çalışmam sırasında ve hayatımın her anında yardımlarını ve anlayışını esirgemeyerek bana sabırla destek veren ve bu günlere gelmemi borçlu olduğum babam, annem, kardeşim, eşim ve çocuklarıma en içten teşekkürlerimle, saygılarımı ve minnettarlığımı sunarım.

Mehmet KORKMAZ
KONYA-2019

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ ve TEŞEKKÜR	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR.....	ix
1. GİRİŞ.....	1
1.1. Tarihçe.....	1
1.2. Tezin Organizasyonu ve Tezde Yapılan Katkılar	5
2. KAYNAK ARAŞTIRMASI	9
2.1. Klasik Dönem SLAM Çalışmaları	9
2.2. Klasik Dönem SLAM Çalışmalarında Karşılaşılan Doğrusallaştırma Problemi ve Algoritma Hızı Üzerine İyileştirmeler	11
2.3. Klasik Dönem SLAM Çalışmaları ile İlgili Gauss Olmayan Gürültü Temsili	12
2.4. Görsel SLAM Çalışmaları (vSLAM).....	13
2.5. Yaya Verileri Kullanılarak Yapılan SLAM Çalışmaları.....	17
2.6. Çoklu Robotlar ve Harita Birleştirme Üzerine Yapılan SLAM Çalışmaları ...	19
2.7. Aktif SLAM	20
3. EŞ ZAMANLI KONUMLAMA VE HARİTALAMA (SLAM)	22
3.1. Giriş.....	22
3.2. Genel SLAM Yaklaşımları	23
3.2.1. Genişletilmiş (EKF) ve Kokusuz Kalman Filtresi (UKF) tabanlı SLAM	24
3.2.2. Parçacık Filtresi (PF) tabanlı yöntemler	31
3.2.3. Graf tabanlı SLAM	32
3.2.4. Görsel SLAM (vSLAM).....	34
3.3. SLAM ile İlgili Diğer Yaklaşımlar	37
3.3.1. Harita birleştirme	37
3.3.2. Aktif SLAM.....	39
4. MATERYEL, YÖNTEM, GELİŞTİRİLEN ALGORİTMALAR VE UYGULAMALAR.....	43
4.1. Giriş.....	43
4.2. Robot İşletim Sistemi (ROS)	43
4.3. Sensör Tabanlı Yöntemler ile İlgili İnceleme	46
4.4. En Uygun Yol Bulma Algoritması ile İlgili İnceleme	50
4.5. Aktif SLAM Algoritmaları ile İlgili Geliştirilen Yöntemler.....	59

4.5.1. Sola yönelimli aktif SLAM algoritması.....	60
4.5.2. CNN tabanlı aktif SLAM algoritması.....	68
4.6. Görsel SLAM Teknikleri ile İlgili Geliştirilen Yöntem.....	73
4.7. Yaya Navigasyonu Başlangıç Problemi için Geliştirilen Yöntem.....	82
5. YENİ BİR ÇOKLU ROBOT HARİTA BİRLEŞTİRME YÖNTEMİNİN GELİŞTİRİLMESİ VE UYGULANMASI	95
6. SONUÇLAR VE ÖNERİLER	108
6.1. Sonuçlar.....	108
6.2. Öneriler	110
KAYNAKLAR	112
ÖZGEÇMİŞ	118



SİMGELER VE KISALTMALAR

Simgeler

$P(A)$: A olayının olasılığı
$P(A B)$: B olayı gerçekleşmişse A olayının olasılığı
m	: Özellik haritası değerleri
C_i	: i . robotun koordinat merkezi
C_0	: Global haritanın koordinat merkezi
J	: Graf SLAM optimallik ölçütü
K	: Kalman Kazancı
P	: Kalman filtresi kovaryans matrisi
x_k	: k anındaki durum vektörü
\hat{x}	: x durumu için kestirim
v_k	: Sıfır ortalamalı gözlem modeline dair k anındaki gürültü vektörü
w_k	: Sıfır ortalamalı hareket modeline dair k anındaki gürültü vektörü
v	: Kalman filtresi inovasyon değeri
S	: Kalman filtresi inovasyon kovaryans matrisi
$p(m_i)$: i . hücrenin meşguliyet olasılığı
$Z_{1:k}$: I . durumdan k . ana kadar ölçümler vektörü
$U_{1:k}$: I . durumdan k . ana kadar kontrol girdileri vektörü
w_i^T	: i . yerel haritanın global koordinatlara aktarımı için gerekli dönüşüm
$\pi_i \oplus \pi_{i+1}$: i . ve $i+1$. yerel haritaların birleşimi
$I_{N \times M}$: $N \times M$ boyutunda bir harita fonksiyonu
$M_{N \times M}$: $N \times M$ boyutunda bir harita fonksiyonu
p	: I haritalarından elde edilen özellikler
t	: Güvenilir özellik
η_i	: Optimal yol hesabı için verim faktörü
$L_{f \times f}$: f filtre boyutundaki karesel bir pencere
μ_{ij}	: Görüntü momenti

Kısaltmalar

A^*	: A yıldız araması
ANOVA	: Tek Yönlü Varyans Analizi
BIM	: Bina Bilgi Modeli (Building Information Model)
CNN	: Evrişimli Sinir Ağı (Convolutional Neural Network)
CEKF	: Sıkıştırılmış Genişletilmiş Kalman Filtresi (Compressed Extended Kalman Filter)
EKF	: Genişletilmiş Kalman Filtresi (Extended Kalman Filter)
ESKF	: Hata Durum Kalman Filtresi (Error State Kalman Filter)
DBN	: Dinamik Bayes Ağı (Dynamic Bayesian Network)
FastSLAM	: A Factored Solution to the SLAM
FAST	: Features from Accelerated Segment Test
GA	: Genetik Algoritma
H_2	: H_2 filtresi

H_{∞}	: H_{∞} filtresi
ICP	: Yinelemeli En Yakın Nokta
IMU	: İçsel Ölçüm Birimi (Inertial Measurement Unit)
KF	: Kalman Filtresi (Kalman Filter)
MCL	: Monte Carlo Konumlama
MonoSLAM	: Tek Gözlü Kameralı SLAM (Monocular SLAM)
ORB	: Oriented FAST and Rotated BRIEF
PF	: Parçacık Filtresi (Particle Filter)
PRM	: Olasılıklı Yol Haritası (Probabilistic Roadmap)
PTAM	: Paralel İzleme ve Haritalama (Parallel Tracking and Mapping)
RANSAC	: Rastgele Örnek Konsensüsü
REMODE	: Düzenleştirilmiş Tek Kamerayla Derinlik Kestirimi (Regularized MONocular Depth Estimation)
RMSE	: Karekök Ortalama Hatası (Root-Mean-Square Error)
ROS	: Robot İşletim Sistemi (Robot Operating System)
RRT	: Hızlı-Kesfeden Rastlansal Ağaçlar (Rapidly Exploring Random Tree)
RSS	: Alınan Sinyal Gücü
Rtab-Map	: Real-Time Appearance-Based Mapping
SEIF	: Seyrek Genişletilmiş Bilgi Filtresi (Sparse Extended Information Filter)
SfM	: İskeletten Yapı İnşası (Structure from motion)
SIFT	: Scale-Invariant Feature Transform
SLAM	: Eş Zamanlı Konumlama ve Haritalama (Simultaneous Localization and Mapping)
SR-UKF	: Karekök Kokusuz Kalman Filtresi (Square-Root Unscented Kalman Filter)
SURF	: Speeded Up Robust Features
UKF	: Kokusuz Kalman Filtresi (Unscented Kalman Filter)
UWB	: Geniş Bant Ötesi
vSLAM	: Görsel Eş Zamanlı Konumlama ve Haritalama (Visual Simultaneous Localization and Mapping)
ZUPT	: Zero Velocity Update

1. GİRİŞ

1.1. Tarihçe

Robot fikri, 3000 yıl öncesine kadar dayanmasına rağmen modern anlamıyla kullanılan “robot” sözcüğü Çekoslovak yazar Karel Čapek tarafından ortaya atılmıştır. Čapek’in kullandığı bu terim, “zorla çalıştırılan işçi” anlamına gelmektedir (1921). 1930’lu yılların sonlarında Westinghouse tarafından “Electro” isimli yürüyüp konuşabilen bir robot geliştirilmiştir. 1940’larda ise Isaac Asimov robot teknolojisinin tümünü kapsayacak bir anlamda “robotik” kelimesini ortaya atmış ve “runaround” isimli yazdığı kitapta robotlarla ilgili üç yasayı tanımlamıştır. Tanımlanan bu kurallar günümüzde bile geliştirilmeye çalışılan robotik, yapay zeka vb. uygulamalarının sınırlarını belirlemek için kullanılmaktadır. Gray Walter tarafından 1950’li yıllarda kaplumbağa görünümü bir robot geliştirilmiştir. Bu robot ışık kaynağına doğru hareket etmekte ve batarya seviyesine göre gerektiğinde şarj noktasına gidebilmekteydi. Bu türdeki robotlar, bir ortamda gezinebilen anlamına gelen *mobil* robotların temellerini oluşturmuştur. 1960’lı yılların sonuna doğru ise “Shakey” isimli, bir ortamda engellere çarpmadan hareket edebilen, komutlarla farklı görevleri yapabilen ve görüntü ekipmanları ile donatılmış bilgisayar kontrollü bir robot geliştirilmiştir (Erden, 2003). Böylece robotik ve yapay zeka konuları birlikte ele alınmaya başlanmış ve günümüzde de hala önemli gelişmelere açık olan bir alan doğmuştur. Bu gibi gelişmeler temelinde günümüze kadar birçok robot veya robotik terminolojisi içerisinde incelenebilen gelişmeler yaşanmıştır. MIT’nin RoboTuna’sı, Sony’nin Aibo robot köpeği, Honda’nın ASIMO’su, Aldebaran firmasının Nao robotu, Boston Dynamics firmasının farklı özelliklerdeki robotları, Willow Garage’nin PR2 robotu, Hanson firmasının Sophia robotu gibi birçok robot geliştirilmiştir ve bu yöndeki çalışmalar günümüzde de devam etmektedir. Bu alandaki çalışmalar incelendiğinde robotlara yaptırılacak işler sayesinde insan hayatını kolaylaştırma, iş verimliliğini artırma, hata payını azaltma gibi birçok fayda sağlanabilmektedir.

Robotlar çok genel manada sabit veya gezgin (mobil) olarak sınıflandırılabilirler. Sabit robotlar genellikle bir noktaya sabitlenerek belirli işleri yapmaktadır. Örneğin kaynak robotları, parça değiştirme için kullanılan robot kollar bu sınıf içerisine dahil edilebilirler. Öte yandan mobil robotlar daha kapsamlı bir kümeyi oluşturmaktadır. Hava, kara, su gibi farklı ortamlarda hareket edebilen türlerinin yanı sıra hareket şekillerine göre

de farklı sınıflar içerisinde incelenebilir. Bu robotların hareket edebilme özelliğinin bir sonucu olarak içerisinde bulunduğu ortamla etkileşime geçmeleri gerekmektedir. Bu yüzden içerisinde bulunduğu ortamı tanıyıp bu ortamda kendisinin nerede olduğunu bilmek mobil robotlarla ilgili en önemli hususlardan birisidir. Bir robotun “*ben neredeyim?*” sorusu literatürde konumlama problemi başlığında incelenmektedir. Robot, sensörleri sayesinde haritasını bildiği bir ortamda çıkarımlar yaparak nerede olduğunu araştırır. Kullanılan robotun türüne göre bu sorunun cevabı değişebilmektedir. Örneğin, iki boyutlu (2B) bir düzlemde kartezyen koordinatlarda genellikle x ve y eksenleri ile kafa açısının (θ) değerleri araştırılırken, üç boyutlu (3B) durum için x , y ve z eksen bilgileri, altı boyut (6B) temsili için x , y , z eksen değerleri ile açısal dönme değerlerinin bilinmesi hedeflenir. Bir mobil robot için konumlama problemi bir Global Konumlama Sistemi (GPS) sensörü sayesinde kolaylıkla çözülebilir. Ancak GPS sensöründen veri alınmadığı birçok durum vardır ve konumlama probleminin farklı bir bakış açısı ile ele alınması gerekmektedir. Bu yüzden robot, GPS sensöründen bağımsız olarak üzerinde bulunan sensörlerle nerede olduğu sorusunun cevabını aramalıdır. Klasik olarak robotun hareket modeli, hareket mesafesi/zamanı, önceki konumu gibi bilgilerle robotun konumu hakkında artırımlı olarak tahmin yapılabilmektedir. Odometri yöntemi olarak bilinen bu yöntem sensör hatalarını barındırmaktadır. Artık hataların birikimli bir şekilde artması sonucunda konum tahmini, robot esas çizgisinden hızlıca uzaklaşabilmektedir. Bu yüzden sonar, lazer gibi mesafe sensörleri ilave dış algılayıcı olarak kullanılmaktadır. Bu sayede bir robotun konumu, hareket modeli/odometri verisi ve mesafe sensör verilerinin birleştirilmesi prosesinin sürekli olarak yinelenmesi ile elde edilmektedir. Haritası bilinen ortamlarda konumlama işlevi literatürde geliştirilen birçok algoritma ile kolaylıkla gerçekleştirilebilmektedir. Bu kapsamda Bayes tabanlı filtreler ve parçacık sürüsü yöntemleri ile gerçekleştirilmiş birçok algoritma vardır.

Konumlama problemine paralel olarak bir robotun bulunduğu ortamda “*çevremdeki dünya neye benziyor?*” sorusu ise haritalama başlığı altında incelenmektedir. Haritalama probleminde robotun konumunun tam olarak bilindiği varsayılmaktadır. Bu sayede robot çevresindeki özellikleri sensörleri vasıtasıyla tanımlayarak ortamın bir haritasını oluşturabilmektedir. Konumlama probleminin aksine haritalama probleminde elde edilen sonuç farklı türlerde olabilmektedir. Örneğin harita çıktısı metrik veya topolojik olabilir ya da metrik haritalar meşguliyet ızgara hücreleri veya özellikler tarafından temsil edilebilir. Ayrıca 2B veya 3B temsillerinde bir diğer çıktı türleridir (Nüchter, 2009). Birçok haritalama algoritması olasılıksal temeller üzerine inşa

edilmiş ve bu çerçevede geliřtirmeler yapılmıřtır. Konumlama problemine benzer řekilde, robotun hareket modeli ve üzerindeki sensörler aracılıęıyla yinelemeli olarak ortam haritası artırımı bir řekilde oluřturulur.

Her ne kadar konumlama ve haritalama problemi literatürde ayrı ayrı ele alınsa da bazı durumlarda robot, ortam haritası ve konumu ile ilgili bilgilerin her ikisinden de yoksundur. Bu tür durumlarda haritalama ve konumlama problemi eř zamanlı olarak ele alınmalıdır. Bu konu literatürde Eř Zamanlı Konumlama ve Haritalama (Simultaneous Localization and Mapping - SLAM) olarak bilinmektedir. Buna göre bir robot, bilmedięi bir ortamda gezinirken bu ortamın haritasını artırımı olarak oluřturmalı ve aynı zamanda bu harita ierisinde kendisini konumlamalıdır. SLAM probleminin çözümlü robota kazandırdıęı otonomi aısından önemlidir ve bu aıdan bakıldıęında SLAM probleminin ortaya konulması ve çözümlü robotik camiasında önemli bir dönüm noktasıdır. Problemin temelleri 1986 yılında yapılan IEEE Robotik ve Otomasyon Konferansında atılmıřtır. Bir dizi arařtırmacı tarafından problemin olasılıksal tabanlı temeller üzerinde incelenebileceęi üzerine tartıřmalar yapılmıř ve bu çerçevede geliřtirilen fikirler ile SLAM probleminin yapı taşları ortaya koyulabilmiřtir (Durrant-Whyte ve Bailey, 2006; Davison ve ark., 2007; Quigley ve ark., 2015). İlerleyen yıllarda problem, dünyanın farklı yerlerindeki arařtırma grupları tarafından ele alınmaya devam etmiřtir. 1990'lı yılların sonu ve 2000'li yılların bařlarında ortaya atılan bir dizi makale, tezler ve yapılan çalıřtaylarla sorunun sınırları tam anlamıyla ortaya koyulabilmiřtir. Problemin ilk yıllarında ortaya konulan çalıřmalar genellikle Bayes tabanlı filtrelerle yapılan çözümlenmelerle olmuřtur. Özellikle Kalman (KF) ve Geniřletilmiş Kalman Filtresi (EKF) tabanlı farklı algoritmalar geliřtirilmiřtir. EKF tabanlı SLAM algoritmaları günümüzde bile geliřtirilen birok çalıřma için bir referans noktası olarak görölmektedir. Daha sonraları, paracık filtresi (PF) tabanlı algoritmaların geliřtirilmesi SLAM çözümlenmelerinin hızını artırmıřtır. SLAM probleminin bu türde çözümlenmeleri "klasik dönem" olarak adlandırılabilir. Bu dönem ierisinde genellikle Bayes tabanlı filtrelerin durum tahmini ve sensör gözlemi ile bu tahminin güncelleme yapıısından faydalanılmıřtır. Bu yapı yinelemeli olacak řekilde bütün bir süreç boyunca kullanılmaktadır. Bu dönem ierisinde sonar, lazer, radar gibi dıř sensörler ile odometri verisi (teker enkoderinden alınan deęerler, hareket modeline göre yapılan tahminler gibi) i sensör olarak kullanılmıřtır.

SLAM problemindeki önemli bir geliřme, kameraların tek dıř sensör olarak kullanıldıęı durumların ele alınması olmuřtur. Bu durum genellikle "modern SLAM"

dönemi olarak adlandırılmakta ve bu algoritmalar görsel SLAM (vSLAM) başlığında incelenmektedir. Bu türdeki yaklaşımlarda ilk olarak Kalman tabanlı filtrelerin kullanıldığı çözümler ortaya çıkmıştır. Ancak kameralardan alınan görüntülerin fazla özellik içermesi algoritmanın zaman performansını açısından pratik olarak uygulanmasını güç kılmıştır. Benzer şekilde kameralarla elde edilen algılama aralığının 360 dereceden az olması da önemli bir eksikliktir. Yine kameraların kullanılması kalibrasyon problemlerinin çözülmesi gerekliliğini ortaya koymuştur. Özellikle tek gözlü kameraların kullanıldığı durumlar için derinlik hesaplamalarının yapılmasına ihtiyaç duyulmuş, fotometrik tutarlılık önemli olmuştur. Konumlama problemi yerine izleme modülü ön plana çıkmıştır. Ayrıca kamera görüntülerinin pasif olması algoritmaların önemini daha da artırmıştır. Özetle SLAM problemi klasik olasılıksal tabanlı zeminden bilgisayar görmesi algoritmalarının kullanıldığı bir temele kaymıştır ve bir nevi gerçek zamanlı hareketten iskelet oluşturma (SfM) problemine dönüşmüştür.

SLAM çözümlenmesi günümüz teknolojisine yansıyan sonuçlar da doğurmuştur. Son yıllarda yavaş yavaş gündelik hayatımıza girmeye başlayan sürücüsüz araba teknolojisinin temelleri SLAM çözümlenmesi ile elde edilmiştir. Bu konuda Amerika Birleşik Devletleri (ABD) Savunma Bakanlığına bağlı bir kuruluş olan Savunma İleri Araştırma Projeleri Ajansı'nın (DARPA) başlattığı sürücüsüz araba yarışları önemli bir hızlandırıcı etki yapmıştır. 2004 yılında düzenlenen yarışta hiçbir araç verilen görevleri tam manasıyla yapamamış ve yalnızca Carnegie Mellon Üniversitesinin aracı 11.78 km'lik bir mesafeyi katetmiştir. Ancak 2005 yılında benzer bir yarış tekrar düzenlenmiş ve birçok takımın görevleri tamamladığı görülmüştür. Beş araç 212 km'lik yarış pistini sürücüsüz bir şekilde tamamlayabilmiştir. 2007 yılında ise yarışmanın boyutu farklı bir mecraya taşınmış ve araçların şehir içi koşullarda sürücüsüz bir şekilde çalışması istenmiştir. Bu yarışta araçlar trafik sinyallerine uyma, çarpışmadan kaçınma, GPS sinyalinden yoksun olma, park edebilme gibi önemli zorlayıcı kurallarla sınırlandırılmıştır. Bu zorluklara rağmen birçok araç verilen görevleri başarı ile tamamlayabilmiştir. Günümüzde ise birçok otomobil üreticisi sürücüsüz araç testlerini başlatmış ve bu konuda farklı hedeflerle ilerleyen yıllar için bütünsel bir çözüm sunma yolunda arayışlara girmiştir. Bu sonuçlar âdeta araç teknolojisini klasik araba üreticilerinden Google ve UBER gibi yazılım firmalarının odağına döndürmüştür.

SLAM konusunun önemli etkilerinden birisi de sağlık alanında yapılan gelişmeler üzerinedir. Klasik bir mide endoskopisi sürecinde hastanın muayenesi midesine doğru ucunda kamera ve ışık bulunan gastroskop isimli yumuşak bir fiberoptik kablo

aracılığıyla yapılmaktadır. Bu yöntemin oluşturduğu problemleri gidermek için içerisinde kamera bulunan kapsüllerin yutulduğu bir sistem geliştirilmiştir. Ancak bu yöntemde sorunlu bölgenin yer tespitinde problemler yaşanabilmektedir. Bunun için SLAM algoritmaları kullanılarak çözümler üretilmeye çalışılmıştır. Bunun sonucunda 3B bir harita ile kapsülün gezdiği bölgelerin haritası çıkarılmış ve konum bilgileri harita üzerine işlenmiştir (Grasa ve ark., 2014).

Son yıllarda giderek her alanda hayatımıza girmeye başlayan artırılmış gerçeklik çalışmalarında en önemli problemlerden birisi, ilgili nesnenin yerinin tespitidir. Bu kapsamda SLAM çözümleri ile artırılmış gerçeklik çalışmalarının birleştirilmesi de önem taşımakta ve bu konuda çalışmalar sürdürülmektedir (Polvi ve ark., 2016).

Özetle SLAM, bir robotun kendi kendine bir ortamın haritasını oluşturması ve bu ortamda kendini konumlaması problemi olarak ifade edilebilir ve günümüzde birçok alanda farklı uygulamalara yol açan teknolojik bir dönüm noktası olmuştur.

1.2. Tezin Organizasyonu ve Tezde Yapılan Katkılar

Bu tez; SLAM konusuna dair tarihçe (Bölüm 1), tez kapsamında yapılan çalışmalarla ilgili bir literatür taraması (Bölüm 2), SLAM ve komşuluğunda incelenen bazı yöntemlere dair açıklamalar (Bölüm 3), SLAM ve vSLAM yöntemleri hakkında benzetim ve gerçek zamanlı uygulamalar üzerinden yapılan incelemeler ve geliştirmeler (Bölüm 4), çoklu robotlar için yeni bir harita birleştirme yönteminin geliştirilmesi ve uygulamaları üzerine çalışmalar (Bölüm 5) ve tez kapsamında yapılan katkıların özetlendiği sonuçlar ve ileride yapılabilecek araştırmaları (Bölüm 6) kapsayan altı bölümden oluşmaktadır.

İlk olarak, klasik SLAM dönemi olarak adlandırılan ve bu çerçevede incelenen Bayes tabanlı filtreler üzerine bir çalışma göz önüne alınmıştır. Bu çalışmada klasik SLAM dönemi içerisinde incelenen ve SLAM algoritmaları için bir mihenk taşı olan EKF-SLAM algoritmasının yine bir diğer Bayes tabanlı Kokusuz Kalman Filtresi (UKF) SLAM algoritması ile karşılaştırılması yapılmıştır. EKF-SLAM yaklaşımlarında SLAM problemi modellenirken Jakobiyen terimlerinden yararlanılmaktadır. Bu da, ilave hatalar oluşturmaktadır. UKF-SLAM yaklaşımında ise parçacık filtresine benzer şekilde ortamdan örnekler alınarak süreç yürütülmekte ve Jakobiyen terimleri kullanılmamaktadır. Bu sayede daha iyi bir doğrusallaştırma adımı gerçekleştirilebilmektedir. Bu düşünceyle, her iki yaklaşımın birbirine göre üstünlüğü ve

eksikliği araştırılmıştır. Yapılan çalışmada harita, konum ve zaman verimlilikleri kıyaslanarak karşılaştırmalı sonuçlar elde edilmiştir. İlgili çalışmaya ait çıktılar Bölüm 3.2.1’de verilmiştir.

Bölüm 4.3’te sonuçları sunulan ikinci çalışmada, SLAM uygulamalarında kullanılan farklı iki sensör, bazı ölçütler bakımından karşılaştırılmıştır. Lazer sensörler, klasik dönem SLAM algoritmaları için büyük bir önem taşımaktadır. Ancak lazer sensörlerin pahalı olması, fazla güç gerektirmeleri ve hantal yapıları sebebiyle alternatif sensörler araştırılmıştır. Bu noktada RGB-D kameralardan yararlanma fikri düşünülmüştür. Bu kameralar görüntü ve derinlik bilgisi verebilmektedir. Bu kameraların nokta bulutu biçiminde verdiği derinlik bilgisi lazer sensör verisi gibi düşünülerek birçok çalışma da kullanılmıştır. Bu tez çalışması kapsamında yapılan bir uygulamada gerçek zamanlı olarak bu verinin bu şekilde kullanılabilmesinin mümkün olup olmadığı araştırılmıştır. Bu kapsamda, gerçek zamanlı bir ortamda lazer sensör ile nokta bulutu verisi verebilen Kinect sensörü, elde edilen haritalar ve zaman açısından karşılaştırılmıştır. Çalışma içerisinde parçacık sayısı parametresinin değişiminin haritalar üzerinde etkisi de irdelenmiştir.

SLAM konusunda en önemli çıktılardan birisi harita verisinin olduğu söylenebilir. Bunun en önemli sebeplerinden birisi robotlardan gerçekleştirilmesi beklenen birçok işlevin haritaları temel almasıdır. Örneğin, robotlara görevler verildiğinde robotlar gideceği yolu harita üzerindeki engeller veya boş alanlara göre hesaplamaktadır. SLAM ile ilgili tez kapsamında yapılan üçüncü bir uygulamada bütün çalışmaların anlamlandırılması adına gerçek zamanlı bir ortamdan elde edilen haritalar üzerinde robotun gideceği en uygun yol nedir ve bununla ilgili hangi parametreler düşünülmelidir sorularının cevabı aranmıştır. Bölüm 4.4’te detayları verilen çalışmanın içeriğinde beş farklı yol planlama algoritması denenerek en uygun yöntemin bulunmasına dair istatistiksel analizler yapılmıştır.

SLAM konusu robotik algoritmalar içerisinde genel bir başlık olarak düşünülebilir. Bu başlık komşuluğunda incelenen yaklaşımlardan birisi aktif SLAM konusudur. Aktif SLAM; robotun harita, konum bilgilerinin yanı sıra navigasyon kısmının da eklenmesi ile elde edilen bir konu bütünlüğüdür. Böylece, robot nereye gitmeliyim sorusuna da kendi algoritmaları çerçevesinde cevap bulmaktadır. Tez kapsamında dördüncü ve beşinci çalışma olarak farklı iki uygulama yapılmıştır. Bölüm 4.5 ve alt başlıklarında detayları sunulan çalışmalarla bu alanda yeni türde algoritmalar geliştirilmeye çalışılmıştır. Geliştirilen çalışmaların birisinde sola yönelimli navigasyon algoritması ile robotun bir ortamda kullanıcı girdisi olmaksızın kendi kendisine nasıl

hareket edebileceğine dair tanımlamalar yapılmış ve sonuçlar irdelenmiştir. Buna ilave olarak son yıllarda sıklıkla birçok alanda kullanılan derin öğrenme yapılarının ilave edildiği yeni türde bütünleşik bir aktif SLAM algoritması da geliştirilmiştir. Derin öğrenmenin bir kolu olan Evrişimli Sinir Ağları (CNN) algoritmaları ile bir robotun bilmediği bir ortamda nasıl hareket edeceğine dair gerçek zamanlı bir çalışma yapılmıştır.

SLAM problemi modern dönemde sadece kamera girdileri ile çözülmeye çalışılan bir problem olmuştur. Bu kapsamda vSLAM başlığı altında incelenen algoritmalar ile ilgili incelemelerde bulunulmuş ve Bölüm 3.2.4'te altıncı bir çalışma olarak farklı iki vSLAM metodunun harita çıktıları karşılaştırılmıştır. Buna paralel olarak, son yıllarda vSLAM konusunda en başarılı algoritmalarından birisi olan ORB özellik tabanlı SLAM tekniği ile ilgili bir çalışma yapılmıştır. ORB-SLAM tekniğinde elde edilen harita özellik tabanlı olduğu için bu metodun derinlik tabanlı bir haritaya dönüştürülmesine bazı uygulamalarda ihtiyaç duyulabilir. Tez kapsamında yapılan yedinci çalışmada ORB-SLAM tekniğinden elde edilen veriler derinlik tabanlı bir haritaya dönüştürülmüş ve bu konuda izlenmesi gereken adımlardan bahsedilmiştir. Bölüm 4.6'da detayları verilen bu çalışma sayesinde literatürdeki bu başarılı tekniğe ilave bir katkı sağlanmıştır.

Bu çalışmaların yanı sıra, robotlar haricinde bir ajan (insan) üzerinde çalışma düşünülmüş ve bu amaçla yaya ayağına yerleştirilen İçsel Ölçüm Birimi (IMU) vasıtasıyla konumlama problemi çözülmek istenmiştir. Bu kapsamda, gerçek zamanlı olarak farklı ortamlardan veriler toplanmıştır. Yinelemesiz Bayes yöntemi ve harita eşleme yöntemleri ile konumlama problemi çözülmürken başlangıç noktasını bulma probleminin çözümü için örnekleme stratejisi üzerinde durulmuştur. Farklı uygulama ortamları için “yürünebilir yollar” tanımlanmıştır. Bu yollar dört farklı seviyede örneklenmiş ve buna bağlı çoklu hipotezler üretilerek sonuçlar elde edilmiştir. Bu sekizinci çalışmanın detayları Bölüm 4.7'de paylaşılmıştır.

SLAM konusu altında incelenen bir diğer konu ise çoklu robotların kullanılması ve bu durumda ortaya çıkan problemlerdir (Saeedi ve ark., 2016). Çoklu robotların kullanılmasının birçok avantajı bulunmasına rağmen en önemli sorunlardan birisi robotlar arası koordinasyondur. Örneğin farklı robotların birbiriyle haberleşmesi, haritaların birleştirilme prosedürleri gibi problemler ayrı ayrı ele alınmalıdır. Bu düşünceyle tez içeriğinde birden fazla robotun kullanıldığı olası bir durum ve bunlardan elde edilen haritaların nasıl birleştirileceği incelenmiştir. Literatürde var olan yöntemler üzerinden çözümler araştırılmıştır. Mevcut yöntemlerin asimetric özellikler barındıran ortamlar için başarılı olabileceği tespit edilmiş ancak simetric bölümler içeren ortamlarda

algoritmaların başarımlarının olumsuz etkilendiđi görülmüştür. Bu kapsamda, Bölüm 5'te de detaylıca anlatılan “*güvenilir özellik*” ismiyle yeni bir tanımlama yapılmış ve “*anlamli harita birleřtirme*” metodu geliřtirilerek buna dair algoritmalar, benzetim ve gerçek zamanlı sonuçlar tez kapsamında yapılan dokuzuncu bir çalıřma olarak sunulmuřtur.



2. KAYNAK ARAŞTIRMASI

Robotik çalışmalar için önemli bir yere sahip olan SLAM algoritması ile ilgili kaydedilen çalışmalar son yirmi yıldır geliştirilmesine karşın bu konudaki fikirler ve ilk çalışmalar 1980'li yılların sonlarına doğru olasılıksal yöntemlerin robotik alanlar üzerine tanımlanması ile ortaya çıkmıştır. Bu literatür taraması SLAM probleminin ortaya çıkışı ve fikrin gelişimi, zaman içerisinde klasik lazer sensör ve Bayes tabanlı filtrelerden kameralar kullanılarak geliştirilen vSLAM algoritmalarına geçiş süreci, sensör ve ajan (robot, insan vb.) türlerine göre geliştirilen SLAM algoritmaları, çoklu robot uygulamaları için SLAM, harita birleştirilmesi ve aktif SLAM yöntemleri ile ilgilidir.

2.1. Klasik Dönem SLAM Çalışmaları

Problemin tanımlanması hususunda yapılan çalışmalarda ortaya çıkan en önemli fikir, harita özelliklerinin konumları ile ilgili yapılan tahminlerin birbirleri arasında bir korelasyona sahip olması ve bu ilintinin peşi sıra gelen başarılı gözlemler ile daha da arttığı keşfedilmesi olmuştur (Smith ve Cheeseman, 1986; Durrantwhyte, 1988). Smith ve ark. (1987) tarafından yapılan çalışmada, robotun pozisyon hatasından dolayı tahminde bulunulan bütün özelliklerin birbiriyle ilişkili olması gerekliliği ortaya konulmuştur. Bu keşif, SLAM algoritmalarında sıklıkla kullanılacak olan durum tahmini ve gözlemlere göre durum tahminlerinin güncellenmesi fikrine yol açacak bir sonucu doğurmuştur. Csorba (1997), doktora tez çalışmasında SLAM konusunda temel yakınsama problemlerini ele alarak problemin tanımlanması hususunda önemli çalışmalar yapmıştır. Bu ve benzeri çalışmalardan elde edilen en önemli sonuç, harita üzerindeki özelliklerin birbiri ile ilintisinin artmasının daha iyi sonuçlar ürettiğinin ortaya konulması ve problemin ilinti arttıkça yakınsadığının keşfedilmesi olmuştur. Buna paralel olarak, Thrun ve arkadaşlarının haritalama ve konumlama problemi için olasılıksal tabanlı geliştirdiği metotlar ve Kalman Filtresi tabanlı SLAM yaklaşımları bu alandaki önemli çalışmalardandır (Thrun ve ark., 1998).

2000'li yılların başlarında SLAM konusundaki yapılan çalışmalar ve Kalman filtresi tabanlı uygulamalarının artırılması ile önemli aşamalar kaydedilmiş ve problem birçok araştırmacı tarafından farklı açılardan ele alınmaya başlamıştır. SLAM probleminde ortam özellikleri ve robotun konumu bir durum matrisi olarak ele alındığında robot her ilerleyişinde bu durum matrisinde bir güncelleme yapılır.

Bahsedilen bu yapı Bayes tabanlı filtreler ile örtüşmektedir. SLAM problemi, yapısında doğrusal olmayan terimler bulundurduğu için problemin çözümünde klasik Kalman Filtresi yerine Genişletilmiş Kalman Filtresi (EKF) ile uygulamalar geliştirilmiştir. Dissanayake ve ark. (2001) çalışmasında SLAM probleminin çözümünün mümkün olduğuna dair detaylı bir yapı verilmiştir. Çalışmanın sonuçlarına göre bilinmeyen bir ortamda konum bilgisi olmaksızın bir ortam haritasının nasıl oluşturulacağı ve bu haritada nasıl konumlama yapılacağına dair bulgular elde edilmiştir. Çalışmada gerçek bir araç üzerine farklı türde sensörler takılarak SLAM algoritması ile ilgili konum ve harita sonuçları elde edilmiştir. Çalışmadan elde edilen harita, özellik tabanlı olup ana sensör olarak radar ve araç tekerleği üzerindeki dönüştürücülerden yararlanılmıştır.

Bailey'in doktora tez çalışmasında, özellikle büyük boyutlu dış ortamlarda SLAM uygulamasının nasıl olacağı ile ilgili noktalar vurgulanmıştır. Bu türdeki alanlarda ortam özelliklerinin karmaşık yapısından kaynaklı problemler dile getirilmiş ve bunların ilişkilendirilmesi ele alınmıştır. Ayrıca büyük ortamlarda ortaya çıkan hesaplama yükü ve birikimli hatanın büyük boyutlu ortamlardaki sorunları ile ilgilenilmiş ve alt haritalara bölme gibi çözümler sunulmuştur (Bailey, 2002).

Nieto ve ark. (2006) çalışmasında, EKF-SLAM yaklaşımında ortam özellikleri ile ilgili karşılaşılan geometrik model tanımlamasının bu algoritmayı sınırladığını, bunun yerine tarama ilişkili yöntemle birleştirilmesinin öneminden bahsedilmiştir. Tarama ilişkilendirilmesi yöntemi ile geometrik modellerden bağımsız özelliklerin kullanımının mümkün olduğunun üzerinde durulmuştur. Sonuçlar deneysel çalışmalarla desteklenmiştir.

Weingarten ve Siegwart (2005) çalışmasında, EKF-SLAM'in 3B uzayda uygulamasının nasıl yapılacağı ve sonuçları üzerinde durulmuştur. Bir düzlem üzerine yerleştirilen 2B bir lazer tarayıcı ile 3B nokta bulutu verileri üretilmiş ve düzlemsel özellikler olasılıksal olarak bu verilerden elde edilmiştir. Veri ilişkilendirilmesinde Mahalanobis uzaklığından yararlanılmıştır. Sonuç olarak, odometri verisinden kaynaklı robot pozisyonundaki birikimli hatanın bahsedilen düzlemsel özellikler kullanılarak düzeltilebileceği vurgulanmıştır.

2.2. Klasik Dönem SLAM Çalışmalarında Karşılaşılan Doğrusallaştırma Problemi ve Algoritma Hızı Üzerine İyileştirmeler

KF veya EKF metotlarının en büyük üstünlüğü çözüme yüksek yakınsama sağlaması ve belirsizliklerle başa çıkmanın nispeten kolay olmasıdır. Ancak bu metotlarla ilgili temel bazı sorunlar bulunmaktadır. Bunlar, bu metotların Gauss (normal) dağılım varsayımı ile hareket etmeleri, doğrusallaştırma hataları ve harita özelliklerinin artışına bağlı olarak üstel bir biçimde algoritma süresinin artışıdır.

Doğrusallaştırma hatalarını gidermek için EKF-SLAM metoduna alternatif çalışmalar yapılmıştır. Kokusuz Kalman filtresi (UKF) tabanlı yaklaşımlar bunların en önde gelenlerindedir. Martinez-Cantin ve Castellanos (2005) çalışması, gerçek zamanlı ve UKF-SLAM'in büyük boyutlu ortamlarda nasıl bir sonuç verdiğini göstermektedir. Çalışmada, EKF-SLAM'in doğrusallaştırma hataları incelenmiş ve buna bağlı olarak UKF-SLAM'in bir çözüm olabileceği vurgulanmıştır. Bu düşünce araştırmacılar tarafından ilgi çekici bulunmuş ve farklı türde geliştirmeler yapılmıştır. Örneğin Li ve Ni (2010) çalışmasında, Karekök Kokusuz Kalman Filtresi tabanlı (SR-UKF) bir UKF-SLAM yaklaşımı geliştirilmiştir. Hem EKF hem de UKF filtrelerinde yuvarlama hatalarına bağlı olarak kovaryans matrisinde simetrisinin kaybolması problemine Karekök Kokusuz Kalman Filtresinin uygulanmasının önemi vurgulanmış ve UKF-SLAM yaklaşımına yeni türde bir bakış açısı getirilmiştir. Çalışmanın sonuçları klasik UKF-SLAM yapısına göre SR-UKF-SLAM yapısının daha iyi olduğunu ortaya koymuştur.

Benzer şekilde zaman verimliliğinin artırılması amacıyla hem EKF yönteminin iyileştirilmesi hem de farklı türde algoritmaların kullanıldığı çalışmalar da yapılmıştır. Paz ve ark. (2008) çalışmasında EKF yönteminde yapılan iyileştirmelerle toplam EKF-SLAM süreç işlem yükü $O(n^3)$ 'ten $O(n^2)$ 'ye düşürülmüştür. Bunun için "böl ve yönet" ismi verilen bir strateji kullanılmıştır. Bu teknik klasik EKF-SLAM yaklaşımında karşılaşılan yeni kısmi haritaların global haritaya peşi sıra eklenmesi yerine hiyerarşik bir yapı ile eklenmesi ile ilgilidir. Çalışmanın uygulama kısmında geleneksel Victoria Park veri seti ile ilgili denemeler yapılmış ve klasik EKF-SLAM yaklaşımına göre hem zaman hem de sonuçların elde edilmesi açısından iyileştirmeler yapılmıştır. Bu stratejinin EKF-SLAM'le ilgili bütün algoritmalara uygulanabileceğini ve bu sayede işlem yükünün azaltılabileceği ifade edilmiştir. Buna alternatif olarak, Sıkıştırılmış Genişletilmiş Kalman Filtresi (CEKF) metodu zaman, hafıza gibi parametreler için önemli bir avantaj sunmaktadır (Guivant ve Nebot, 2001). Bu metot sayesinde daha büyük alanların

işlenmesi mümkün olabilmektedir. Ancak bu yöntemin gürbüz özellik gereksinimi, alt haritaların birleştirilmesi ve buna bağlı olarak verilerin ilişkilendirme problemi gibi olumsuzlukları bulunmaktadır. Tuna ve ark. (2012) çalışmasında EKF, UKF ve CEKF SLAM algoritmalarının işlem süreleri farklı işlemcilerde sahip bilgisayarlar üzerinde denenmiş ve karşılaştırmalı bir çalışma sonuçları sunulmuştur.

2.3. Klasik Dönem SLAM Çalışmaları ile İlgili Gauss Olmayan Gürültü Temsili

Gauss olmayan gürültülerin temsili için de farklı türde algoritmalar geliştirilmiştir. Örneğin, Chandra ve ark. (2010) çalışmasında EKF-SLAM yaklaşımına alternatif yöntemler araştırılmış ve filtre yapısının değiştirilmesi durumu incelenmiştir. EKF filtresinin Gauss olmayan gürültülerden kaynaklı sorunlarından ve doğrusallaştırma hatalarından bahsedilmiştir. Buna binaen EKF'deki Kalman filtresine alternatif H_2 ve H_∞ filtreleri ve bunların karışımı bir filtre türünün SLAM problemine uyarlanması detayları ve sonuçları ele alınmıştır. Düşük değerli Gauss gürültüler için EKF ve H_∞ filtrelerinin benzer sonuçlar verdiğini E_{H_2}/H_∞ filtresinde ise aracın kaybolduğundan bahsedilmiştir. Buna paralel olarak yüksek değerli gürültülerde E_{H_∞} filtresinin EKF yaklaşımından daha iyi sonuçlar verdiği üzerinde durulmuştur. Bu filtre türünün bazı durumlarda EKF-SLAM yaklaşımına bir alternatif olabileceği üzerinde durulmuştur.

Parçacık filtrelerinin kullanılmasıyla Gauss olmayan gürültüler temsil edilebilmiş ve SLAM uygulamaları için önemli bir gelişme kaydedilmiştir. Parçacık filtresinin kullanımı her ne kadar konumlama problemi için uygun olsa da haritalama probleminde yeni tanınan özelliklerin global olarak haritaya eklenmesinden dolayı sürekli olarak hesap yükü artmaktadır. Bu yüzden SLAM problemi için pratik olarak uygulanması neredeyse imkansızdır. Bu noktada, Montemerlo ve ark. (2002) tarafından SLAM probleminin haritalama ve konumlama kısımlarının ayrı ayrı ele alındığı bir ayrıştırılma çözümü (FastSLAM) geliştirilmiştir. FastSLAM olarak adlandırılan çalışma SLAM uygulamaları için önemli bir mihenk taşı olmuş ve EKF-SLAM yaklaşımlarının bu yöne doğru kaymasını sağlamıştır. Bu çalışmaların temelinde Rao-Blackwellization ayrıştırma metodunun SLAM problemine uyarlanmasını vardır. Bu algoritma sayesinde özellik pozisyonlarının robotun yolundan koşullu olarak bağımsız olduğu varsayımı SLAM problemini Monte Carlo konumlama (MCL) ve 2×2 EKF haritaları problemine dönüştürebilmiştir.

Kurt-Yavuz ve Yavuz (2012) çalışmasında, klasik SLAM algoritmaları ile ilgili detaylı bir karşılaştırma yapılmıştır. Çalışmada UKF tabanlı bir FastSLAM algoritması geliştirilmiş ve klasik algoritmalara göre zaman olarak kötü ancak pozisyon doğruluğu olarak daha iyi sonuçlar verdiği vurgulanmıştır.

Ali ve ark. (2018) çalışmasında, FastSLAM 2.0 algoritmasının işlem yükü ile ilgili problemlerine değinilmiş ve son yıllarda yaygınlaşan bulut üzerinden işlemleri yürütmeye dayalı bir çalışma yapılmıştır. Bu sayede işlem yükü, tek bir robotla yapılanaya göre %83.6'ya kadar düşürülebilmektedir.

2.4. Görsel SLAM Çalışmaları (vSLAM)

SLAM problemi için klasik dönem olarak görülen çalışmalar Thrun ve ark. (2005); Bailey ve Durrant-Whyte (2006); Durrant-Whyte ve Bailey (2006) makalelerinde detaylı olarak ele alınmıştır. Her ne kadar klasik SLAM çalışmaları SLAM probleminin anlaşılması ve çözülmesinde önemli bir rol oynasa da son yıllarda SLAM problemindeki çözümler yönünü, kameraların dış sensör olarak kullanıldığı görsel SLAM (vSLAM) algoritmaları noktasına kaydırmıştır.

vSLAM algoritmalarının altında yatan temel düşünce, hantal ve çok fazla güç harcayan lazer sensörler ve doğruluk payı düşük olan sonar sensörlere alternatif bir sensör türü ile SLAM probleminin çözümünün ortaya konulmasıdır. Ayrıca kameralar maliyetlerinin düşük olması ve ortam ile ilgili daha fazla veriyi içermesi açısından önemli bir yer teşkil etmektedir. Bu verilerle, farklı türde algoritmaların yürütme şansına sahip olunması da görsel SLAM çalışmalarını ön plana çıkarmaktadır. Kameraların lazer veya sonar sensörlere göre en büyük üstünlüğü/eksikliği bu sensörlerin pasif olmalarıdır. Lazer veya sonar aktif sensörlerinin zıttına, kameradan elde edilen bilgiler pasiftir. Bu da SLAM konusunda geliştirilen algoritmalara daha çok iş düşmesi anlamına gelmektedir. Bununla birlikte ortam görüntüsü içerisinde bulunabilecek veri yoğunluğunun diğer sensörlere göre daha fazla olacağı aşikardır. Bu sayede robota ileri seviye işler yaptırılmak istenirse bunlar da mümkün olabilmektedir. Yine ucuz, hafif ve düşük güç tüketimleri kameraların önemli üstünlüklerinden olup bu sayede vSLAM konusuna eğilim artmıştır.

vSLAM ile ilgili ilk gerçek zamanlı ve olasılıksal tabanlı çalışma Davison tarafından yapılmıştır (Davison, 2003). Sonraları Smith ve ark. (2006); Davison ve ark. (2007) tarafından yapılan çalışmalarla geliştirmeler yapılmıştır. Davison (2003) çalışmasında, tek gözlü bir kamera kullanılmış ve bilinmeyen bir ortam için 3B

haritalama ve kamera hareketlerinin belirlenilmesi hedeflenmiştir. Verilerin tahmini için klasik SLAM yaklaşımlarındaki gibi EKF metodundan faydalanılmıştır. Davison'un çalışmasında hareket modeli olarak sabit hız ve açıların kullanılması bu çalışmanın gerçek zamanı tam temsil etmeme sorununu ortaya koymuştur. Çünkü kamera hareketlerini sabitlemek gerçek zamanlı birçok çalışma için bir problem teşkil etmektedir. Bununla birlikte global bir koordinat sistemi tanımı için bilinen bir nesnenin özelliklerinden yararlanılmış ve buna göre yeni gözlemlenen özellikler haritaya dahil edilmiştir. Kameranın 6 serbestlik dereceli konumları ve özelliklerin 3B uzaydaki koordinatları, EKF için durum vektörü olarak tanımlanmıştır. Klasik SLAM problemlerinde karşılaşılan fazla sayıda özelliğin olduğu durumlardaki yüksek hesaplama zaman problemi bu yaklaşımda da ortaya çıkmış ve gerçek zamanlı uygulamalarda pratik olarak kullanılabilmesi mümkün olmamıştır.

Smith ve ark. (2006) çalışmasında, ortama ilgili çizgilerin tanımlandığı ve bu düz çizgilerin özellik olarak kullanıldığı bir tek gözlü kameralı SLAM yapısından bahsedilmiştir. Çalışmada bir harita olmadan çizgi tanımlanmasının zor olması ve harita inşa edilirken aynı zamanda bu çizgilerin belirlenmesinin zor olduğu vurgusu yapılmıştır. Çizgilerin noktalara göre daha büyük ortam temsiline kullanılabilecek özellikler olduğundan bahsedilmiş ve ortam temsiline daha etkin olabileceği vurgulanmıştır. Bu özelliklerin tek gözlü kameralı SLAM yapısı içerisinde nasıl ekleneceği gibi hususlar üzerinde incelemeler yapılmış ve yakın kenar çizgilerin olduğu durumlarda algoritmanın gürbüzlüğünün etkilendiğinden bahsedilmiş ve bu sorunlar üzerine ileri çalışmalar yapılması gerekliliği vurgulanmıştır. İlerleyen dönemde bu çalışmalar MonoSLAM (Davison ve ark., 2007) başlığı altında genellenmiştir.

Klasik SLAM yapısında olduğu gibi tek gözlü kameralı SLAM yaklaşımında da EKF yapısının farklı türde algoritmalarla değiştirilmesi hususunda çalışmalar yapılmıştır. Chekhlov ve ark. (2006) çalışmasında, gerçek zamanlı bir video karesinde çalışan, tek bir kamera ile SLAM uygulamasının nasıl yapılacağı ve bunun gürbüzlüğü üzerine bir araştırma yapılmıştır. UKF temelli bir algoritma çerçevesi sunulmuştur. Önceki yöntemlere göre kamera sarsıntısı, stabil olmayan hareketler gibi gürbüzlük konusunda karşılaşılan problemlere bir alternatif sunulmuştur. Ancak EKF karşılaştırması veya zaman değerlendirmesi gibi konulara değinilmemiştir. Buna mukabil Holmes ve ark. (2008) yayımında, bir önceki çalışmanın olumsuzluklarına karşın bazı katkılar sağlanmıştır. Bunun için SR-UKF tabanlı bir MonoSLAM çalışması yapılmış ve çalışmanın sonuçları klasik EKF ve UKF MonoSLAM yapıları ile karşılaştırılmıştır.

Çalışmaya göre SR-UKF algoritması ile elde edilen sonuçlar zaman açısından olumsuz olmasına karşın haritanın geometrik tutarlılığı açısından EKF'ye göre avantajlı olduğu gösterilmiştir.

MonoSLAM algoritmaları ile özellik tabanlı haritalar elde edilebilmektedir. Ancak bu yöntemlerin en büyük problemi zaman kısıtlamasıdır. Bu noktada paralel izleme ve haritalama (PTAM) olarak adlandırılan metodun uygulanması ile özellik tabanlı haritaların oluşturulduğu yöntemlerde önemli bir değişim olmuştur. Bu konudaki ilk çalışmalar Klein ve Murray (2007) makalesinde artırılmış gerçeklik konusunda ele alınmıştır. Temelinde, izleme ve haritalama işlevleri işlemcide farklı iki parçaya ayrılarak ele alınmaktadır. Bu sayede demet optimizasyonu yapılabilmekte ve klasik EKF MonoSLAM yapısında karşılaşılan özellik sayısının artmasına bağlı zaman verimliliğinin düşmesi probleminin önüne geçilmiş ve binlerce özelliklerin olduğu haritalarda bile işlem yapılabilmektedir. Çalışmanın en dikkat çeken yönlerinden birisi videonun normal bir kare oranında gerçekleştirilebilmesidir. Bu çalışmanın bir devamı niteliğinde Klein ve Murray (2009) makalesinde bir cep telefonu kamerasından yararlanılarak PTAM algoritması üzerine denemeler yapılmıştır. Telefon kameralarının dar görüş açısı ve buna bağlı olarak gürbüz bir izleme yapılmasının güçlüğünden bahsedilmiştir. Bu güçlükler rağmen algoritma tam kare oranında çalıştırılabilmiş ve gerçek zamanlı olarak küçük boyutlu haritalar üretilebilmiştir. Sheng ve ark. (2011) çalışmasında, PTAM algoritması kullanılarak lazer tabanlı ölçümler için iyileştirmeler yapılmıştır. Buna göre, mobil bir robot üzerinden lazer sensör bilgisi ve tek gözlü bir kamera ile görüntü bilgisi alınmıştır. Elde edilen veriler birleştirilerek ortam haritası oluşturulmuş ve konumlama yapılmıştır. Çalışmada yalnızca iç ortamlarla ilgili bir haritalama yapılmıştır. Gelecek çalışmalarda ise canlı 3B veri akışı durumunun incelenmesi üzerine vurgu yapılmıştır.

Özellik tabanlı vSLAM algoritmaları ile ilgili önemli bir gelişme ORB tabanlı makalelerin yayınlanması olmuştur. 2015 yılında yayınlanan ilk makalede, ORB-SLAM çerçevesi tanıtılmıştır (Mur-Artal ve ark., 2015). Gerçek zamanlı iç, dış ve büyük ortamlar için uygulamalar yapılmış ve buna göre geliştirilen yöntemin önceki algoritmalara göre zaman açısından büyük bir üstünlük sağladığı vurgulanmıştır. En iyilerin seçimi stratejisi ile önemli bir gürbüzlük sağlanmıştır. Algoritma içerisinde konumlama, haritalama, yeniden konumlama ve kapalı döngüleri elde etme gibi stratejiler başarı ile sağlanabilmiştir. Çalışma, gerçek zamanlı veriler üzerinde ve açık kaynak kodlar kullanılarak doğrulanmıştır. Küçük boyutlu iç ortamlarda 1 cm'nin altında, büyük boyutlu dış ortamlarda ise 1 m'nin altında bir doğruluk elde edilmiştir. Çalışma temeli,

PTAM algoritmasına ORB özelliklerinin eklenmesi üzerine kurulmuştur. Mur-Artal ve Tardos (2017a) çalışmasında ise bu metod hem stereo hem de konumlama konularında iyileştirmeler yapılarak geliştirilmiştir. Sheng ve ark. (2011) çalışmasına göre ORB kelime paketlerinde deęiřtirmeler yapılmıř ve kapalı dngülerden sonra demet dzenlemesi yapılmıřtır. Lv ve ark. (2017) çalışmasında Kinect 2.0 kamerası ile ORB-SLAM tabanlı bir 3B modelleme yapılmıřtır. ORB-SLAM’de elde edilen seyrek grntl ortam haritalarının engelden kama ve navigasyon iin uygun olmadıęından bahsedilmiř ve bu konuda yeni trde bir haritayı ortaya koyarak geliştirme yapmıřlardır. Algoritmada kullanılan kamera el ile ortamda gezdirilerek haritalama ve pozisyon takibi yapılmıřtır. zetle SLAM uygulamaları ierisinde tek gzl kameraların kullanıldıęı ve zellik tabanlı haritaların retildeęi metotlar MonoSLAM ile bařlayıp PTAM ile geliştirilmiř ve ORB-SLAM ile ileri bir seviyeye tařınmıřtır.

ORB-SLAM yapısı tek gzl, stereo ve RGB-D kameraların hepsine uygulanması yn ile bir avantaj sunmasına raęmen literatrde sadece RGB-D kameralar kullanılarak gereklenen vSLAM alıřmaları da vardır. RGB-D kameralar standart bir kamera ile elde edilen grnt zelliklerine ilave olarak derinlik bilgisini de saęlamalarından dolayı bazı alıřmalarda tercih edilmiřtir. Bu kameralar konsol oyunları iin retilmesine raęmen birok farklı robotik uygulamasında ve SLAM algoritmalarında sıklıkla kullanılmıřtır. Sturm ve ark. (2012); Henry ve ark. (2014) alıřmaları bu konulardaki ilk makalelerdendir. Henry ve ark. (2014) alıřmasında bu trde kamelerin i ortamlarda haritalamada nasıl kullanılacaęı zerinde durulmuřtur. Ortam zellikleri ve řekil tabanlı hizalamalardan yararlanılmıřtır. Kapalı dng tanımlamaları iin renkli ve bununla iliřkili derinlik grntlerinden faydalanılmıřtır. Hizalama iřlevi iin Rastgele rnek Konsenss (RANSAC) ve Yinelemeli En Yakın Nokta (ICP) algoritmalarından yararlanılmıřtır. Sturm ve ark. (2012) alıřmasında ise bir robot zerinde veya el ile kameranın tutulduęu durumlara dair incelemeler yapılmıřtır. Bu alıřma, zellikle gerek zamanlı ve kapsamlı bir veri setinin literatre kazandırılması aısından nemli olmuřtur. Ayrıca algoritmanın deęerlendirilmesine dair mutlak iz hatası ve greceli pozisyon hatası gibi farklı iki lt sunulmuřtur.

Bu yaklařımlara ilave olarak grsel-isel SLAM olarak adlandırılan alıřmalarda bulunmaktadır. Bu alıřmalarda dıř sensr olarak kamera ve farklı isel sensrler kullanılmaktadır. Isel sensrler, robotun hareketleri ile ilgili isel bilgilerden olan hız, ivmelenme, odometri deęiřimi hakkında bilgi vermektedir. Bunlar genelde kodlayıcı, ivmeler, jiroskop gibi sensrlerle llr ve grsel SLAM alıřmaları ile entegre

edilebilir. Achtelik ve ark. (2012) çalışmasında, görsel-işsel SLAM ile ilgili bir yapı otonom uçuş yapabilen küçük bir helikopter üzerinde kullanılmıştır. Çalışmada, harici sensör olarak yalnızca tekli bir kamera kullanılmıştır. Kamera görüntüleri ile IMU sensör verileri birleştirilmiş EKF ve PTAM yapısı kullanılmıştır. Çalışma, uçuş hızı 2 m/s seçilerek 0.2~70 m yüksekliğinde ve 350 m uzunluğunda bir yol üzerinde yapılmıştır. Ancak çalışma içerisinde sonuç haritaları ile ilgili görsel harita veya yol izleri paylaşılmamıştır. Fu ve ark. (2013) çalışmasında ise bir İnsansız Hava Aracı (UAV) platformu kullanılmıştır. Çalışmada bulanık denetleyici kontrol yapısı çapraz entropi (CE) tekniği ile optimize edilmiş görsel-işsel SLAM algoritmasının çarpışmadan kaçışı ve güvenli inişi için iyileştirmeler yapılmıştır. İç sensör olarak IMU, dış sensör olarak tekli kameralar ve görsel SLAM tekniği olarak PTAM metodu kullanılmıştır. Çapraz Entropi yapısının klasik bulanık denetleyicisinin başlangıç kural tabanını %64 oranında düşürmesini sağlayacak bir optimizasyon yaptığı vurgulanmıştır. Oriolo ve ark. (2015) çalışmasında ise insansı robotların görsel-işsel sensörler yardımıyla gerçekleştirilmesi üzerine bir SLAM çerçevesi sunulmuştur. Çalışmada işsel veri olarak IMU, eklem dönüştürücüleri, ayak basınç sensörü ve robotun kinematik denklemleri gibi girdiler kullanılmıştır. Sensör veri birleştirilmesi için EKF algoritmasından yararlanılmıştır. Görsel SLAM algoritması için PTAM yapısından yararlanılmıştır. EKF ve PTAM 30 Hz sıklığında çalıştırılmıştır. Mur-Artal ve Tardos (2017b) çalışmasında ise görsel işsel SLAM uygulamalarının klasik kapalı döngü problemleri ile ilgili sorunlarından bahsedilmiş ve buna çözüm olabilecek bir algoritma sunulmuştur. Özellikle daha önceki birçok algoritmanın harita üzerindeki aynı yerleri tekrar gördüğü durumlarda bile hatanın iyileşmemesi probleminden bahsedilmiştir. Bununla birlikte, yeni bir IMU başlangıçlandırma yöntemi uygulanarak genel SLAM yapısının hızlandırılması hususunda gelişmeler kaydedilmiştir. Çalışma, genel erişime açık EuRoC mikro hava araçları veri tabanı üzerinde gerçekleştirilerek sonuçlar doğrulanmıştır. Sensör olarak IMU ve tekli kamera düzeneğinden yararlanılmıştır.

2.5. Yaya Verileri Kullanılarak Yapılan SLAM Çalışmaları

SLAM uygulamaları mobil robotlar, mikro ölçekli hava araçları, araç üzerine entegre edilen sensörler gibi birçok farklı alanda ve farklı türde sensörle gerçekleştirilmiştir. Robotik için birçok çalışmada SLAM başlığı altında çözümlere gidilmesine rağmen yayalar için bunun yapılamamasının en önemli sebeplerinden birisi

harici sensörlerin bu uygulamalar için uygun olmayışıdır. Bu türde fikirler insanların yürüyüşü ile konumlama ve haritalama problemi birlikte ele alınabilir mi düşüncesini ortaya çıkarmıştır. Patrick Robertson ve çalışma arkadaşları tarafından bu düşünceleri temel alan farklı makaleler ileriye sürülmüştür. Angermann ve Robertson (2012) çalışmasında FootSLAM ismiyle bir algoritma ortaya konulmuştur. İnsan ayağına yerleştirilen IMU'lar sayesinde ortam haritası çıkarabilmekte ve konumlama yapılabilmektedir. Çalışmanın temelinde Dinamik Bayesian Ağ (DBN) kullanılması vardır. Makalede de belirtildiği üzere “*İçsel sensörlere bağlı navigasyon sonsuz pozisyon hatasına yol açacak bir sonuca ulaşabilir*” vecizesine meydan okuyan bir yaklaşım geliştirilmiştir. Çalışmanın içeriği FastSLAM olarak bilinen algoritmaya benzemektedir. Ancak oluşturulan haritalar, karesel yerine altıgen ızgara formatında olacak şekilde tanımlanmıştır. Robotik SLAM ile karşılaştırıldığında hareket modelleri ve bulunan kontrol girdileri yayalar için aynı olmayacağından farklı bir şekilde tanımlama yapılmıştır. Bu kapsamda bu çalışmada insanın görsel-bilişsel sistemlerinin etrafı yorumlayıp buna göre hareket etmesi kaidesi üzerinde durulmuş ve DBN modeli buna göre oluşturulmuştur. Uygulamaların gerçekleşmesinde 10’ar dakikalık yürüyüş ve IMU verileri ile denemeler yapılmıştır. Özellikle gelişen IMU teknolojisi ve küçük ölçekli elektro-mekanik (MEMS) tabanlı sensörlerin uygulamaya koyulmasının önemli çalışmalara yol açacağından bahsetmiştir. Bruno ve Robertson (2011) çalışmasında ise yine aynı ekip tarafından FootSLAM olarak bilinen metoda kablosuz bağlantı alanı sinyalleri (Wi-Fi) ve baz istasyonlarından alınan sinyal gücü (RSS) değişkenlerinin eklenmesi ile iyileştirilmelerin yapılması amaçlanmıştır. DBN ile tahmin problemi karakterize edilmiştir. Çalışmada yenilik olarak FootSLAM metoduna RSS benzerlik teriminin eklenmesi olmuştur. Benzetim ve gerçek zamanlı denemeler ile sonuçlar doğrulanmıştır. Bu makaleler temel alınarak Garcia Puyol ve ark. (2014) çalışmasında, çok katlı bina yapıları için FootSLAM uygulaması gerçekleştirilmiştir. Özellikle geniş ölçekli çok katlı alanlar için bu türde yapılan ilk çalışmalardandır. Gentner ve Ulmschneider (2017) çalışmasında ise son yıllarda sıklıkla ele alınan bir konu olan Geniş Bant Ötesi (UWB) sistemlerin yayalar üzerine uygulanan SLAM sistemleri ile birleştirilmesinden bahsedilmiştir. IMU’dan alınan verilerle UWB sisteminden alınan mesafe tahminleri ile birleştirilmiştir. Özellikle yaya konum tahminlerinin gerçek verilere oldukça yakın olduğu ve genelde 1 metrenin altında hata olduğundan bahsedilmiştir.

2.6. Çoklu Robotlar ve Harita Birleştirme Üzerine Yapılan SLAM Çalışmaları

SLAM konusunu içerisinde incelenebilecek önemli başlıklardan birisi de birden fazla robotun kullanıldığı durumlardır. SLAM çalışmaları göz önüne alındığında birçok makalenin bir tek robot veya ajanın kullanımı ile gerçekleştirildiği görülmektedir. Çoklu robotlar ve bunlarla ilgili karşılaşılan problemler ayrı bir konu olarak makalelerde ele alınmaktadır. Çoklu robotlarda karşılaşılan problemlerden birisi farklı robotlardan gelen kısmi haritaların birleştirilmesi ve buna bağlı olarak global haritanın elde edilmesidir. Bu problem genel olarak SLAM konusu içerisinde harita birleştirme alt başlığı altında ele alınmaktadır. Harita birleştirme konusunda ilk makalelerden sayılabilecek olan Konolige ve ark. (2003) çalışmasında, takım halinde çalışan robotların tekli bir robota göre harita birleştirme konusunda daha etkin olabileceğini vurgulamıştır. Bununla birlikte, bu problemin konumlama ve haritalama problemi kadar dikkat çekmediğinin de altı çizilmiştir. Özellikle kapılar, köşeler ve yolların birleştiği kısımlar düşünülmüştür. Kısmi haritalarda bulunan özellik kümeleri benzerlik fonksiyonları ile araştırılmış ve buna göre global haritanın elde edilmesi amaçlanmıştır. Topal ve ark. (2013) çalışmasında ise arama kurtarma bölgelerinde karşılaşılan çoklu robot kullanımları için harita birleştirme problemi üzerinde durmuştur. Çalışmada, harita olarak meşguliyet ızgara haritalarından yararlanılmıştır. Kısmi haritalarda bulunan benzer özelliklerin ve alanların birbirine yakınlığı ölçülmüş ve buna göre birleştirme yerlerinin belirlenmesi amaçlanmıştır. Özellikle felaket bölgesi gibi düzgün olmayan bina, yerleşke, çalışma ortamı vb. kısımların kullanıldığı durumlar için benzetim tabanlı uygulama yapılmıştır. Carpin (2008) makalesinde, harita birleştirilmesi konusunda birçok çalışmanın da temel aldığı meşguliyet ızgara haritaları üzerinde bir çalışma yapılmıştır. Çalışmada geliştirilen yeni türde bir algoritmaya göre haritalardaki birleşim yerlerini bulabilmek için muhtemel dönüşüm noktaları üretilmektedir. Üretilen bu dönüşüm noktaları, geliştirilen bir kabul endeksi ile ağırlıklandırılmaktadır. Bununla birlikte kısmi haritaların aynı ölçekte olması ve belirli bir kesişim bölgelerinin olması gerekliliğinden bahsedilmiştir. Ayrıca büyük boyutlu alanlarda bitiş noktalarındaki bozuk kısımların harita birleştirmesinde hizalama problemleri yaratabileceği gibi sorunlar algoritmanın eksikliği olarak belirtilmiştir. Harita birleştirme probleminde birçok problem meşguliyet ızgara tabanlı haritalar üzerinden ele alınsa da topolojik haritaların birleştirilmesi üzerinede literatürde yapılan çalışmalar vardır. Huang ve Beevers (2005) çalışması bu konu ile ilgili kapsamlı detaylar vermektedir. Çalışmada çoklu robotların ortak bir referans noktası olmaması problemi ve

bunun topolojik haritalarla çözülebileceğinden bahsedilmiştir. Çalışma içerisinde iki topolojik haritanın birleştirilmesi ve bu konuda izlenmesi gereken algoritma ile ilgili bilgiler verilmiştir. Haritaların iskeletine bakarak eşleşme noktaları ile ilgili hipotezler üretilmiş, geometrik dönüşümlerle ilgili bölgeler gruplanarak global harita elde edilebilmiştir. Topolojik kısmi haritaların eşleştirilmesi için resim tescilleme ve en büyükleyen alt graf eşleştirilmesi yöntemlerinden yararlanılmıştır. Erinc ve Carpin (2013) çalışmasında görünüş tabanlı çoklu robotlu bir sistemde topolojik haritaların birleştirilmesi problemi üzerinde durulmuş ve bu konuda bir algoritma geliştirilmiştir. Algoritmalarında geliştirdikleri yöntem kelime sözlüğü (BoW) tabanlıdır. Görüntü tabanlı haritaların birleştirilmesinin değerlendirilmesi ile ilgili bir ölçüt eksikliğinden bahsedilmiş, bu konuda cebirsel bir ölçüt tanımlanmış ve bunun öneminden bahsedilmiştir. Ayrıca robotlar tarafından oluşturulan bu haritaların değerlendirilebileceği bir ölçütün hala eksik olduğu hususunda da vurgular yapılmıştır.

2.7. Aktif SLAM

SLAM'in konu aldığı ve bir nevi bir üst kümesi olan aktif SLAM konusu ise genel itibariyle bir mobil robotun SLAM özelliklerine ilave olarak kendi kendine ortamda hareket edebilmesini sağlayan navigasyon yapısının eklenmesi ile oluşmaktadır. SLAM, her ne kadar haritalama ve konumlama problemlerini ele alsada robotun ortamda nasıl gezdirileceği hangi noktalara gitmesi gerektiği gibi konular bu başlık içerisinde ele alınmamaktadır. Bir robotun ortamda nasıl hareket edeceği, nereye gideceği gibi sorular navigasyon şemasının konusudur. Aktif SLAM çalışmalarında ise robot bir ortamda hem SLAM yapmakta hem de nereye gideceğine dair kararları vermektedir. Bu bakış açısıyla robota tam otonominin ancak ve ancak aktif SLAM yapısı ile kazandırılacağı söylenebilir. Bu düşünceyle SLAM araştırmacıları, navigasyon yapısının bu probleme eklenmesi üzerine de uğraşmaktadır. Aktif SLAM'de genel amaç robotun SLAM yapması ve aynı zamanda ortamda en fazla yeri en az belirsizlikle keşfetmesidir. Bir robotun ortamda otonom keşifler yapmasına dair ilk çalışmalara Yamauchi (1997) makalesinde değinilmiştir. Robotun önündeki açık bölge ile keşfedilmemiş bölgeler arasındaki bölgeler öncül yerler olarak adlandırılmış ve basitçe robotun bu bölgelere yönlendirilmesi ile ortam keşfi yapılabilmıştır. Ortamın keşfinde sensör olarak lazer ve sonar sensörleri birlikte kullanılmıştır. Bu sayede hem 3B keşif yapılabilmiş hem de pahalı bir 3B lazer sensörün kullanımına bir alternatif geliştirilmiştir. Bu çalışmada SLAM ele alınmasada

aktif SLAM konusu ile ilgili çalışmalara yol açıcı olmuştur. Fox ve ark. (2006) ise bu makaleyi temel almakta ve çoklu robotların keşif, haritalama ve aralarında harita sensör bilgileri paylaşımı gibi konularda kapsamlı bir sonuçlar sunmaktadır. Carrillo ve ark. (2012) çalışmasında D optimallik ölçütünün aktif SLAM çalışmaları için belirsizlik hesaplamasında nasıl kullanılacağı üzerinde durulmuştur. Entropi ve A optimallik ölçütlerine karşı D optimallik ölçütünün karşılaştırılması yapılmış ve D optimallik ölçütünün uygunluğu üzerine yorumlar yapılmıştır. Çalışmada EKF tabanlı bir aktif SLAM algoritmasından yararlanılmış ve 25×25 metre boyutundaki bir alanda 0.3 m/adım ile hareket eden bir robot kullanılmıştır. Carlone ve ark. (2013) makalesinde aktif SLAM problemi için RaoBlackwellization Parçacık Filtresi yönteminden yararlanmıştır. Algoritmanın değerlendirilmesi ve yeni keşif veya aynı yerden geçme konusunda verilecek kararlar için Kullback-Leibler ıraksama (ya da göreceli entropi) metodu kullanılmıştır. Aktif SLAM ve keşif sürecinin nihai amacı, bilgi kazancını en büyükleyen bir optimizasyon problemi olduğundan bahsedilmiştir. Mu ve ark. (2016) çalışmasında klasik meşguliyet ızgara haritaları yerine topolojik haritalar temel alınarak bir aktif SLAM şeması sunulmuştur. Özellik tabanlı temsillerden dolayı yol planlamaları klasik durumdan farklı olarak her bir özelliğe fazladan bilgi yüklenecek şekilde ayarlanmıştır. Robotun gerçek zamanlı ayarlanan yol planları için olasılıksal yol harita metodu kullanılmıştır.

Bu tezde, SLAM başlığı altında gerek klasik gerekse modern dönem algoritmaları için benzetim ve gerçek zamanlı uygulamalar yapılmış ve mevcut yöntemler bazı bakımlardan iyileştirilmiştir. Özellikle birçok çalışmanın göz ardı ettiği gerçek zamanlı incelemeler yapılmıştır. Bununla birlikte, çoklu robotlar konusunda harita birleştirme üzerine literatürde iyi irdelenmeyen bazı hususlar etraflıca araştırılarak yeni bir yöntem geliştirilmiştir.

3. EŞ ZAMANLI KONUMLAMA VE HARİTALAMA (SLAM)

3.1. Giriş

SLAM problemi kısaca, eş zamanlı olarak bir robotun ortam haritasını oluşturması ve aynı zamanda kendisini bu haritada konumlandırmasıdır (Durrant-Whyte ve Bailey, 2006; Thrun ve Leonard, 2008; Dissanayake ve ark., 2011; Cadena ve ark., 2016). Haritalama için konum bilgisine ihtiyaç duyulurken, konumlama için harita bilgisine gereksinim vardır. SLAM’de bu iki problem eş zamanlı olarak çözülmesi gerektiği için SLAM, genel olarak robotik araştırmacıları arasında, ilgi çekici ve zor bir problem olarak görülmektedir. SLAM probleminin çözümünde birçok farklı metot bulunmaktadır. Özellikle Bayes tabanlı algoritmalarla önemli sonuçlar elde edilmiştir. Buna mukabil ortam özellikleri ve zaman arasında karmaşık bir ilişki olduğu için zamanla bu metotların üzerine iyileştirmeler yapılarak alternatif algoritmalar geliştirilmiştir. Bunlara ilave olarak, lazer, radar veya sonar sensörler yerine kameraların dış sensör olarak kullanıldığı yaklaşımlar SLAM problemine farklı bir bakış açısı getirmiş, son yıllarda üzerinde ciddi çalışmalar yapılmış ve hala yapılagelmektedir. SLAM problemi, kendi içerisindeki geliştirilen algoritmaların yanı sıra birçok problem için de bir üst şemsiye olarak düşünülebilmektedir. SLAM problemine navigasyon kısmının eklenmesi ile aktif SLAM çalışmaları yapılabilmektedir. Bu sayede mobil bir robota tam otonomi kazandırılabilir. Benzer şekilde, özellikle çoklu robotların kullanıldığı durumlar için farklı haritaların birleştirilme probleminin çözümünde bu problemin ele aldığı konulardan birisidir.

Klasik manada SLAM, gözlem ve robot kontrol girdilerine karşılık harita ve konum bilgisi çıktılarının elde edildiği koşullu bir olasılık problemi olarak tanımlanabilir (Eşitlik (3.1), (Khairuddin ve ark., 2015)).

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (3.1)$$

Eşitlik (3.1)’de x robotun k anında konum ve yönelim açısını, m haritada bulunan özelliklerin konumlarını, $Z_{0:k}$ başlangıçtan k anına kadar yapılan gözlemleri, $U_{0:k}$, ise başlangıçtan k anına kadar uygulanan kontrol girdilerini işaret eder. Robotun başlangıçtan k anına kadar tüm konumları ($x_{0:k}$) göz önüne alınarak bir sonraki adımla

ilgili tahmin yapılırsa tam SLAM, aksi durumlar içinse (x_k) çevrimiçi SLAM şeklinde tanımlamalar yapılmıştır.

Robot için hareket ve gözlem modelleri Eşitlik (3.2) ve (3.3) ile tanımlanabilir.

$$P(x_k | x_{k-1}, u_k) \quad (3.2)$$

$$P(z_k | x_k, m) \quad (3.3)$$

Bu bilgiler ışığında SLAM problemi klasik bir tahmin/düzeltilme yinelemeli döngüsü sırasıyla Eşitlik (3.4) ve Eşitlik (3.5) ile tanımlanabilir.

$$P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) \times P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1} \quad (3.4)$$

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k | x_k, m) P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k | Z_{0:k-1}, U_{0:k})} \quad (3.5)$$

Bazı kaynaklarda başlangıç olarak $k - 1$ yerine k . zaman baz alınmaktadır. Eşitliklerde tutarlılık olması açısından Durrant-Whyte ve arkadaşlarının notasyonu kullanılmıştır. Bu şekilde elde edilen tahmin ve düzeltme basamakları yinelemeli bir yapı oluşturmaktadır. Bu sayede klasik Bayes tabanlı filtreler ile örtüşen bu yapıya farklı türde algoritmalar uygulanabilmektedir (Durrant-Whyte ve Bailey, 2006).

SLAM ile ilgili olarak klasik yaklaşımlar genel olarak Bayes tabanlı filtrelerden oluşmaktadır. Genişletilmiş Kalman Filtresi (EKF), Kokusuz Kalman Filtresi (UKF), Karekök Kokusuz Kalman Filtresi (SR-UKF), Sıkıştırılmış Genişletilmiş Kalman Filtresi (CEKF), Genişletilmiş Bilgi Filtresi (EIF), Seyrek Genişletilmiş Bilgi Filtresi (SEIF) gibi birçok Kalman Filtresi varyasyonları ve parçacık filtresi tabanlı FastSLAM 1.0, FastSLAM 2.0 gibi algoritmalar sıklıkla SLAM probleminin çözümü için uygulanmıştır. Burada literatürde sık kullanılan EKF, UKF, parçacık filtresi tabanlı klasik dönem SLAM algoritmaları, farklı bir yaklaşım olan graf SLAM ve son yıllarda popüler olan görsel SLAM yöntemleri ile ilgili bilgiler verilecektir.

3.2. Genel SLAM Yaklaşımları

Bayes yöntemi tabanlı Kalman filtresi, keşfinden bu yana birçok sistemde kullanılmaktadır. Genel anlamıyla dinamik bir değişimin olduğu durumlarda bilinmeyen bilgilerin, mevcut verilerden hareketle tahmini üzerine dayanmakta olan optimal bir

kestirim filtresidir. Bu sayede bilinmeyen bir veri kabul edilebilir bir varyansla tahmin edilebilmektedir. Bu yönüyle SLAM probleminin çözümü için çok uygundur ve bu kapsamda birçok algoritma tanımlanmıştır. Bunun yanı sıra grafik temsilleri ile ifade edilen çizge tabanlı SLAM algoritmaları tam SLAM probleminin çözümü için birçok algoritmanın altyapısını oluşturmaktadır.

3.2.1. Genişletilmiş (EKF) ve Kokusuz Kalman Filtresi (UKF) tabanlı SLAM

SLAM problemi kendi özünden kaynaklı doğrusal olmayan kısımlar barındırmaktadır. Bu yüzden, klasik Kalman filtresi yerine Genişletilmiş Kalman Filtresi (EKF) bu problemin çözümünde kullanılmıştır. EKF-SLAM algoritmasında durum değişkenleri μ veya x , robot konum bilgisi x ve m harita üzerinde yer alan özelliklerin pozisyonlarıdır (Eşitlik (3.6)). Durum değişkeni μ olarak düşünülürse buna bağlı kovaryans matrisi ise P ile ifade edilir (Eşitlik (3.7)).

$$\bar{x} = \mu_k = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} \quad (3.6)$$

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^T & P_{mm} \end{bmatrix}_{k|k} \quad (3.7)$$

EKF SLAM algoritması için tahmin aşamasında robotun hareket modeli ($P(x_k|x_{k-1}, u_k)$) ve bir önceki durumlar kullanılır (Eşitlik (3.8)).

$$x_k = f(x_{k-1}, u_k) + w_k \quad (3.8)$$

Burada $f(\cdot)$ robot kinematiklerine, w_k sıfır ortalamalı Gauss gürültüsüne işaret etmektedir ($w_k \sim \mathcal{N}(0, Q_k)$). Bu bilgiler ışığında tahmin ile ilgili Eşitlikler (3.9) ve (3.10)'daki gibi düzenlenebilir.

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, U_k) \quad (3.9)$$

$$P_{xx,k|k-1} = \nabla f P_{xx,k-1|k-1} \nabla f^T + Q_k \quad (3.10)$$

Burada ∇ operatörü Jakobiyen işlemini ifade etmektedir.

EKF SLAM algoritmasının ikinci adımı ise tahminin düzeltilmesidir. Buna göre gözlem (ölçüm) modeli ($P(z_k|x_k, m)$) ve gözlemler kullanılır. Ölçüm modeli genel manada Eşitlik (3.11)'deki gibi ifade edilebilir.

$$z_x = h(x_k, m) + v_k \quad (3.11)$$

Burada $h(\cdot)$ ölçüm geometrisini v_k ise sıfır ortalamalı Gauss gürültüsüne işaret etmektedir ($v_k \sim \mathcal{N}(0, R_k)$). Buna göre ölçümle ilgili inovasyon ve inovasyon kovaryansı Eşitlik (3.12) ve (3.13)'teki gibi tanımlanır.

$$V_k = z_k - h(\hat{x}_{k|k-1}, \hat{m}_{k-1}) \quad (3.12)$$

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (3.13)$$

Bu adımlara göre elde edilen yeni durumla ilgili kestirim ve kovaryansı Eşitlik (3.14) ve (3.15)'teki gibi elde edilir.

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_{k|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{m}_{k|k-1} \end{bmatrix} + K_k V_k \quad (3.14)$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \quad (3.15)$$

burada K kalman kazancı olup Eşitlik (3.16)'daki gibi bulunur. (Durrant-Whyte ve Bailey, 2006; Chandra ve ark., 2010).

$$K_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (3.16)$$

Kalman filtresi, modellerin doğrusal olduğunu varsaymaktadır. EKF filtresinde ise doğrusallaştırma işlemi Taylor serisi açılımı ile yapılmaktadır. UKF filtresi bunun daha iyi bir yöntemle yapılıp yapılmayacağı üzerinde durmaktadır. Bu kapsamda, doğrusallaştırma işleminde Taylor serisi ve Jakobiyen matrislerinin yerine sigma noktaları olarak adlandırılan nokta kümesi seçilerek işlemler yapılmaktadır. Bu sayede türev hesabı yapılmasına gerek olmamaktadır.

Klasik bir Kokusuz dönüşüm (UT) için n durum vektör boyutu olmak üzere $2n + 1$ sayısının sigma noktaları ($\chi^{[i]}$) Eşitlik (3.17)'ye göre beklenti değerinin (μ) etrafında seçilir. i indeks değişkeni olup $\chi^{[i]}$ bu değişkene ait sigma noktasını ifade eder.

$$\begin{aligned}\chi^{[0]} &= \bar{x} = \mu \\ \chi^{[i]} &= \bar{x} + \left(\sqrt{(n+\lambda)P}\right)_i, \quad i = 1, 2, \dots, n \\ \chi^{[i]} &= \bar{x} - \left(\sqrt{(n+\lambda)P}\right)_i, \quad i = n+1, n+2, \dots, 2n\end{aligned}\tag{3.17}$$

Burada P ortalamaya ait kovaryansı temsil eder. Bununla birlikte $\left(\sqrt{(n+\lambda)P}\right)_i$ ifadesi genellikle Cholesky ayrıştırma yöntemi kullanılarak bulunur. λ ise ölçekleme parametresi olup Eşitlik (3.18) ile tanımlanır.

$$\lambda = \alpha^2(n+k) - n\tag{3.18}$$

α ve k sigma noktalarının ortalama etrafında yayılmasını belirleyen ölçekleme parametreleri olup α genellikle $10^{-4} \leq \alpha \leq 1$ aralığında, k ise çoğunlukla 0 olarak seçilir. Bu bilgiler ışığında hareket modeli (Eşitlik (3.19)), tahmin ortalaması ve ilgili kovaryans sırasıyla Eşitlik (3.20) ve (3.21)'deki gibi olur.

$$\chi_{k|k-1}^{[i]} = f\left(\chi_{k-1|k-1}^{[i]}, U_k\right)\tag{3.19}$$

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} w_i \chi_{k|k-1}^{[i]}\tag{3.20}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} w_i \left[\chi_{k|k-1}^{[i]} - \hat{x}_{k|k-1}\right] \left[\chi_{k|k-1}^{[i]} - \hat{x}_{k|k-1}\right]^T + Q_k\tag{3.21}$$

Burada w_i i . sigma noktasına ait ağırlığı temsil etmektedir.

UKF'nin ikinci aşaması olan düzeltme adımında ise gözlem ve gözlem modellerinden yararlanır. UKF filtresi parametreleri ilgili adımlara uygulanırsa gözlem modeli Eşitlik (3.22)'deki elde edilir. Buna bağlı inovasyon ve kovaryansı ise Eşitlik (3.23) ve (3.24)'teki gibi elde edilir.

$$Z_{k|k-1}^{[i]} = h\left(\chi_{k|k-1}^{[i]}\right)\tag{3.22}$$

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2n} w_i Z_{k|k-1}^{[i]}$$

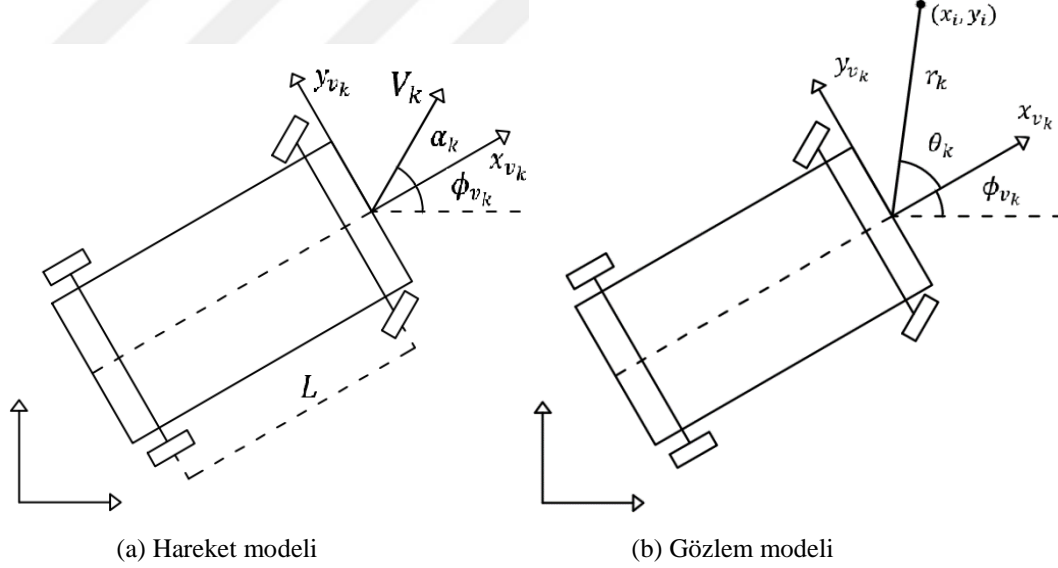
$$V_k = z_k - \hat{z}_{k|k-1} \quad (3.23)$$

$$S_k = \sum_{i=0}^{2n} w_i \left[Z_{k|k-1}^{[i]} - \hat{z}_{k|k-1} \right] \left[Z_{k|k-1}^{[i]} - \hat{z}_{k|k-1} \right]^T + R_k \quad (3.24)$$

Kestirim durumu ve kovaryansı EKF SLAM algoritmasındaki gibi hesaplanır. Burada K Kalman Kazancı ise Eşitlik (3.25)'teki gibidir (Li ve Ni, 2010; Havangi ve ark., 2014).

$$K_k = P_{xz,k|k-1} S_k^{-1} \quad (3.25)$$

UKF yöntemi Jakobiyen türevi içermemesi ve doğrusal olmayan terimlerin tanımlanması açısından EKF algoritmasına göre daha avantajlıdır. Bu düşünceyle SLAM algoritmaları üzerinde bir karşılaştırma yapılmıştır. Benzetim ortamında büyük boyutlu bir alanda hareket eden Ackermann sürme tekniğini temel alan bir araç kullanılmıştır. Araçla ilgili hareket modeli ve gözlem modeli Şekil 3.1'deki gibidir. Aracın taban uzunluğu (L) 4 metre, sensörün maksimum uzaklığı 30 metre, adımlar arası (k ve $k - 1$ durumları arası) süre (ΔT) 0.025 saniye olarak seçilmiştir.



Şekil 3.1. EKF ve UKF SLAM algoritmalarının karşılaştırılmasının yapıldığı deneme için kullanılan araca ait modeller; (a) araç hareket modeli, (b) sensör ölçüm modeli (Bailey ve ark., 2006)

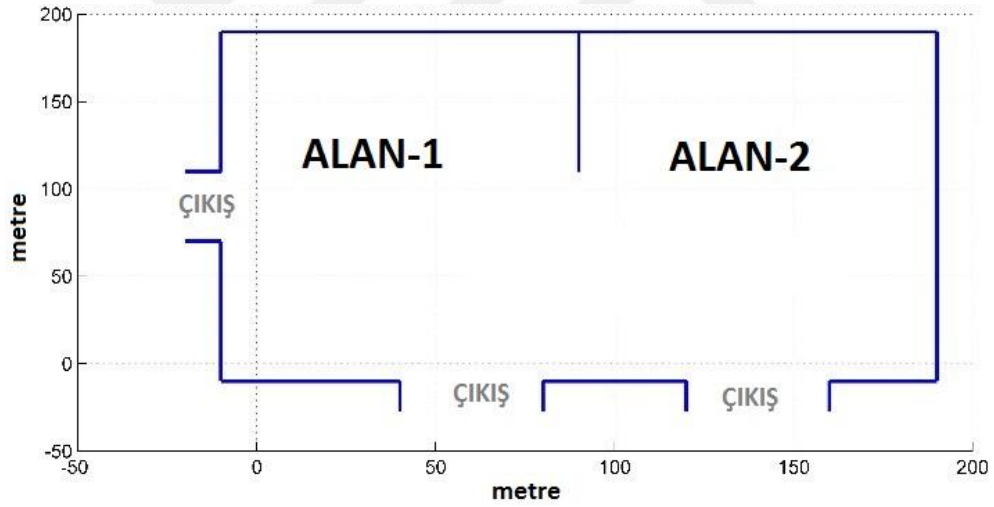
Araç ile ilgili hareket ve gözlem modelleri Eşitlik (3.26) ve (3.27)'deki gibidir.

$$x_{v_k} = f(x_{v_{k-1}}, u_k) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \alpha_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \alpha_k) \\ \phi_{v_{k-1}} + V_k \Delta T \sin(\alpha_k)/L \end{bmatrix} \quad (3.26)$$

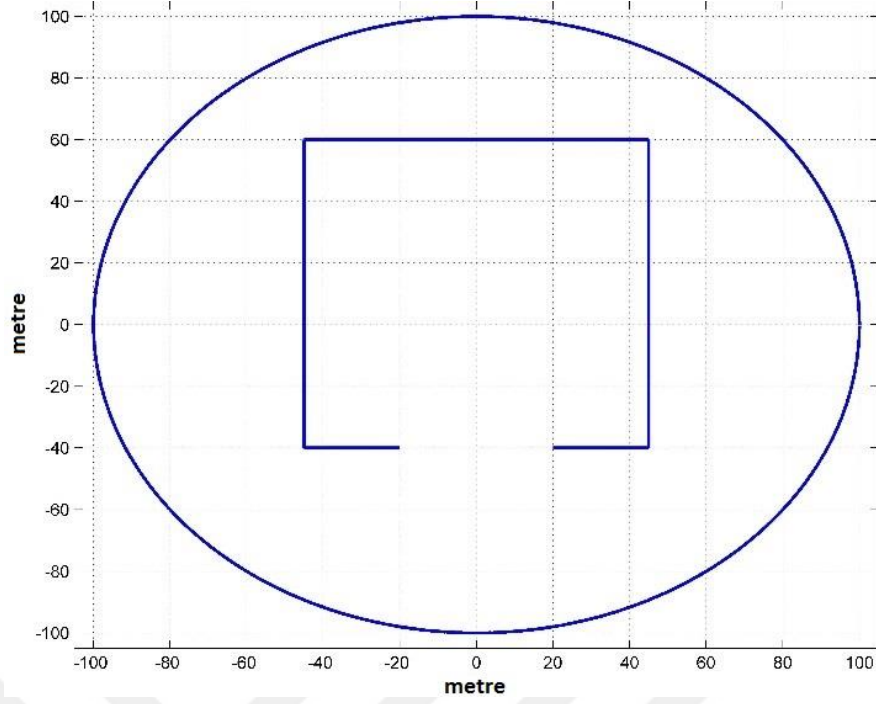
$$z_{i_k} = h_i(x_k) = \begin{bmatrix} \sqrt{(x_i - x_{v_k})^2 + (y_i - y_{v_k})^2} \\ \tan^{-1} \left(\frac{y_i - y_{v_k}}{x_i - x_{v_k}} \right) - \phi_{v_k} \end{bmatrix} \quad (3.27)$$

burada, x_k k . zaman durum vektörünü ifade eder ve o andaki robotun 2B pozisyon vektörü (x_{v_k}) ve ortam özelliklerine ait konumları içerir. Ayrıca u_k kontrol girdisini, V_k araç hızını, α_k teker dönme açısını ifade etmektedir. $[x_i, y_i]$ ise i . harita özelliğine ait konum bilgilerini gösterir. Bu özellikler harita üzerinde durağan olduğu varsayıldığı için herhangi bir zaman ifadesi içermezler (Bailey ve ark., 2006; Chandra ve ark., 2010).

Benzetim çalışması için farklı iki boyut ve şekilde alanlar oluşturulmuştur (Şekil 3.2 ve 3.3).

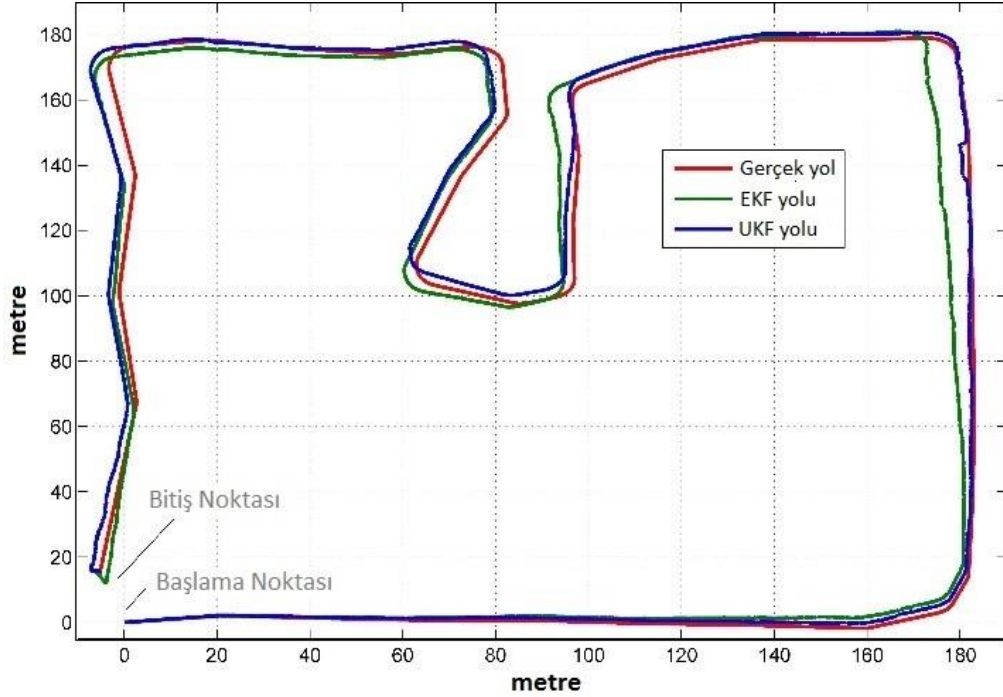


Şekil 3.2. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasında kullanılan kare şekilli hangar alanı (uygulama - 1)

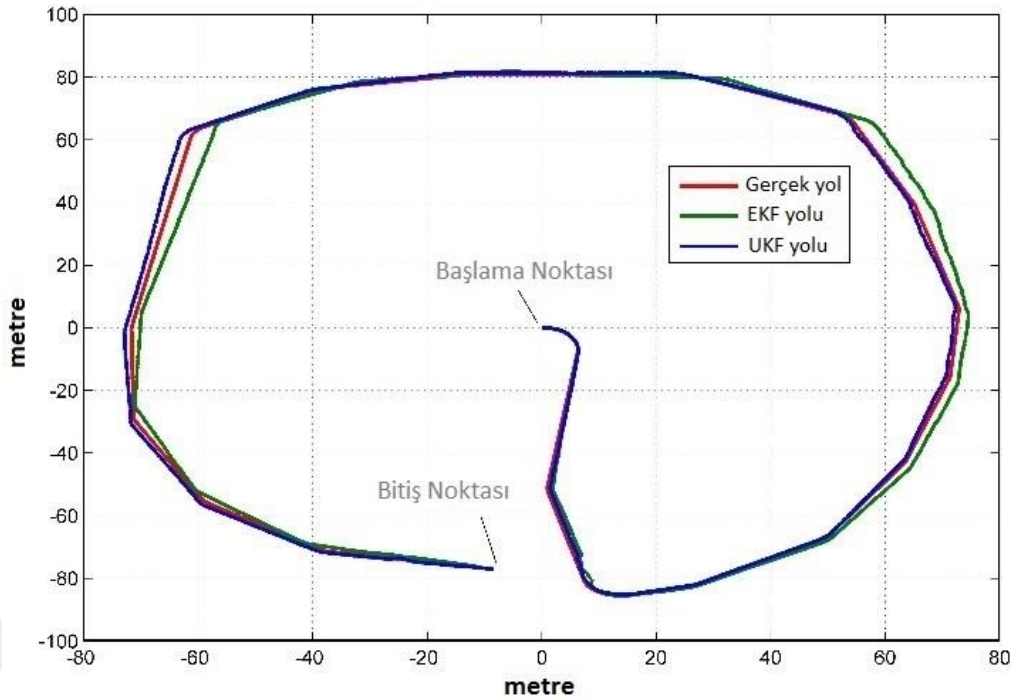


Şekil 3.3. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasında kullanılan daire şekilli hangar alanı (uygulama - 2)

Araç her iki alanda da başlangıç noktasından bitiş noktasına hareket ettirilmiştir. Buna göre EKF ve UKF SLAM algoritmalarından elde edilen yol eğrileri ve gerçek yol eğrileri her iki farklı alan için Şekil 3.4 ve Şekil 3.5'teki gibi olmuştur.



Şekil 3.4. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasına dair gerçek ve algoritma yol eğrileri (kare şekilli hangar alanı, uygulama - 1)



Şekil 3.5. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasına dair gerçek ve algoritma yol eğrileri (daire şekilli hangar alanı, uygulama - 2)

Algoritmaların karşılaştırılmasına dair her iki alan için aracın pisti tamamlama süresi, pozisyon-yönelim açıları kayıt edilmiştir. Hatanın hesaplanabilmesi için gerçek ve algoritma eğrileri değerleri arasındaki hatanın kare kök ortalaması (RMSE) bulunmuştur (Çizelge 3.1 ve 3.2).

Çizelge 3.1. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasına dair sonuçlar: pozisyon, yönelim hataları ve gerçekleşme zamanı (kare şekilli hangar alanı, uygulama - 1)

Algoritma	Pozisyon Hatası	Yönelim Hatası	Zaman (s)
EKF	4,43432	0,78978	62,1163
UKF	2,55331	0,60402	72,1636

Çizelge 3.2. EKF ve UKF SLAM algoritmalarının karşılaştırılması uygulamasına dair sonuçlar: pozisyon, yönelim hataları ve gerçekleşme zamanı (daire şekilli hangar alanı, uygulama - 2)

Algoritma	Pozisyon Hatası	Yönelim Hatası	Zaman (s)
EKF	3,63777	0,54790	46,0049
UKF	1,78697	0,20794	49,6039

Elde edilen veriler ışığında iki algoritma arasında dikkat çekici farklar görülmemiştir. Bununla birlikte, UKF-SLAM algoritmasının gerçek eğrilere daha yakın

sonuçlar verdiği ve hataların daha az olduğu gözlemlenmiştir. Ancak zaman açısından EKF-SLAM algoritmasına göre biraz daha yavaş olduğu görülmüştür.

3.2.2. Parçacık Filtresi (PF) tabanlı yöntemler

EKF-SLAM çözümünde ortamdaki özellik sayısının artması ile kovaryans matrisinin boyutu karesel bir şekilde artmaktadır. Bu yüzden büyük boyutlu veya fazla özellik barındıran ortamlarda algoritma çok yavaş çalışmaktadır. Bununla birlikte diğer bütün Kalman Filtresi ile çalışan algoritmalar Gauss dağılımı varsayımı ile hareket etmektedir. Bu ve benzeri sorunlar göze alındığında parçacık filtresinin kullanılması SLAM problemine farklı bir bakış açısı getirmiştir. Parçacık filtresi, Kalman Filtresi tabanlı yöntemlerden farklı olarak parametrik olmayan bir Bayes filtresi olup Gauss olmayan dağılımların temsilinde de kullanılabilir. Durum kestirimleri ağırlıklandırılmış örnek kümeleri ile ifade edilmekte olup 3 temel basamağı bulunmaktadır. Bunlar örnekleme, ağırlıklandırma ve yeniden örneklemedir. Parçacık tabanlı SLAM algoritması basit bir şekilde $X_t^{[k]}$ örnek yolları, $\mu_{t,1}^{[k]}, \dots, \mu_{t,N}^{[k]}$, N adet 2B Gauss ortalamalarını ve $\Sigma_{t,1}^{[k]}, \dots, \Sigma_{t,N}^{[k]}$ ilgili kovaryansları işaret etmektedir. Burada N ortamdaki özellik sayısını, k ise ilgili bir parçacığı ifade eder (Thrun ve Leonard, 2008). Buna göre, birinci aşama olan örnekleme hareket modelinden, Eşitlik (3.28)'deki gibi elde edilir.

$$x_t^{[k]} \sim p(x_t | x_{t-1}^{[k]}, u^t) \quad (3.28)$$

Ağırlıklandırma aşaması için z_t ölçümü göz önüne alınarak Eşitlik (3.29)'da görüleceği üzere k . parçacık için ağırlık katsayıları hesaplanır.

$$w_t^{[k]} := \mathcal{N}(z_t | x_t^{[k]}, \mu_{t,n}^{[k]}, \Sigma_{t,N}^{[k]}) \quad (3.29)$$

Üçüncü aşama ise yeniden örneklendirme aşaması olup mevcut parçacıklar yeni parçacıklarla değiştirilerek elde edilir. Buradaki temel bakış açısı daha fazla önemli olan parçacıkları tutup az ağırlığa sahip olanları değiştirmektedir. Nihayetinde ise EKF algoritması düzeltme basamağında olduğu gibi z_t ölçümlerine göre yeni ortalama $\mu_{t,n}^{[k]}$ ve kovaryans matrisleri $\Sigma_{t,N}^{[k]}$ elde edilir. FastSLAM algoritmasının en önemli altyapısı Rao-

Blackwellization ayrışımı üzerine dayanmaktadır (Eşitlik (3.30)). Buna göre robot konumu ve özellik yerlerinin birbiri ile ilişkisi olmadığı varsayımı temel alınmaktadır.

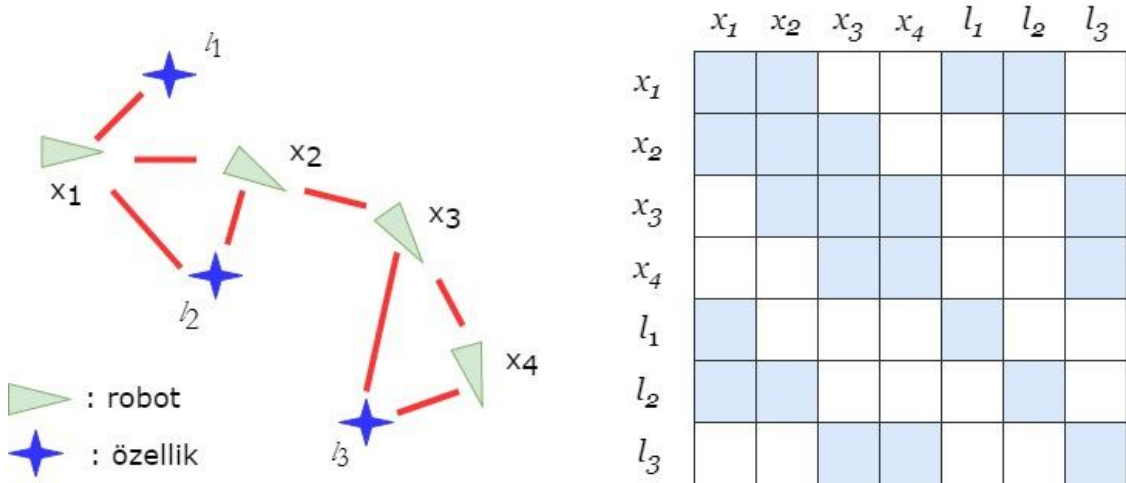
$$p(x_{0:t}, l_{1:N} | z_{1:t}, u_{1:t}) = p(x_{0:t} | z_{1:t}, u_{1:t}) \prod_{i=1}^N p(l_i | x_{0:t}, z_{1:t}) \quad (3.30)$$

Denklemden dikkat edilecek husus, konumlama ve haritalama kısımlarının ayrı ayrı ele alınmış olmasıdır. Böylece Monte Carlo Konumlama (MCL) problemi yapısına benzeyen parçacık filtresi ve 2 boyutlu EKF blokları elde edilmiştir.

FastSLAM 2.0 ise örnekleme ve ağırlıklandırma aşamaları ile FastSLAM 1.0'ın genişletilmiş bir versiyonudur. Bu yapı baz alınarak birçok geliştirme yapılmıştır. Özellikle güncelleme aşamasında yeni örneklerin eklenmesi geçmiş verilerin kaybolmasına sebep olmaktadır. Bunun için literatürde sunulan çözümler vardır (Stachniss ve ark., 2004; Kim ve ark., 2008). Her ne kadar dezavantajları olsa da parçacık filtresi tabanlı FastSLAM algoritmaları ve türevleri klasik SLAM metotları için gelişmiş bir çözüm sunmakta ve pratik olarak dış ve büyük ortamlar için de kullanılabilir.

3.2.3. Graf tabanlı SLAM

Bu teknik, SLAM probleminin çözümünde grafik modelden yararlanmaktadır. Bu türdeki algoritmaların temel dayanağı özellik ve robot konumlarının graf üzerinde düğüm olarak ifade edilmesi ile oluşturulmasıdır. Bununla birlikte düğümler arasındaki kenarlar kısıtları oluşturmakta ve kontrol girdileri, sensör ölçümleri ile ifade edilmektedir (Şekil 3.6).



Şekil 3.6. Graf SLAM tekniği için örnek bir graf ve bunlara karşılık gelen tablo (Dine ve ark., 2015)

Graf temsilli gösterime göre birbiri ile ilintili durumlar dolu karelerle ifade edilmekte, ilişkisi olmayan durumlar ise boş karelerle gösterilmektedir. Peşi sıra gelen robot konum bilgileri bir kontrol girdisi ile birbirine bağlanır. Örneğin, Şekil 3.6 dikkate alınırsa robotun x_2 konumundan x_3 konumuna hareketi için u_{2-3} kontrol girdisi uygulandığında bu veri graf üzerinde bir kenarı oluşturur ve tabloda dolu bölge olarak işaretlenir. Bununla birlikte bu veri tüm problem için bir kısıt oluşturur. Benzer durum ölçüm modelleri için de geçerlidir. x_2 konumunda gözlemlenen l_2 harita özelliği, bir kısıt oluşturur ve tabloda dolu bölge olarak işaretlenir. Bu düşünce ile bir irdeleme yapıldığında logaritmik olarak tam SLAM problemi üç kısımdan oluşur: önceki durumlardan kaynaklı sabit parametre girdileri, hareket ve ölçüm modellerinin oluşturduğu kısıtlar (Eşitlik (3.31)).

$$\log p(X_T, m | Z_T, U_T) = \text{sabit} + \sum_t \log p(x_t | x_{t-1}, u_t) + \sum_t \log p(z_t | x_t, m) \quad (3.31)$$

Graf üzerindeki kenarlar, robotun her bir adımı ($\log p(x_t | x_{t-1}, u_t)$) ve yapılan sensör ölçümleri ($\log p(z_t | x_t, m)$) sonucu oluşmaktadır.

Bu bilgiler ışığında SLAM probleminin çözümü Eşitlik (3.32)'nin modunu bulma problemine dönüşür.

$$X_T^*, m^* = \operatorname{argmax}_{X_T, m} \log p(X_T, m | Z_T, U_T) \quad (3.32)$$

Bu denklemin çözümü Eşitlik (3.33) ile verilen üssel bir fonksiyonun bulunması ile ilgilidir.

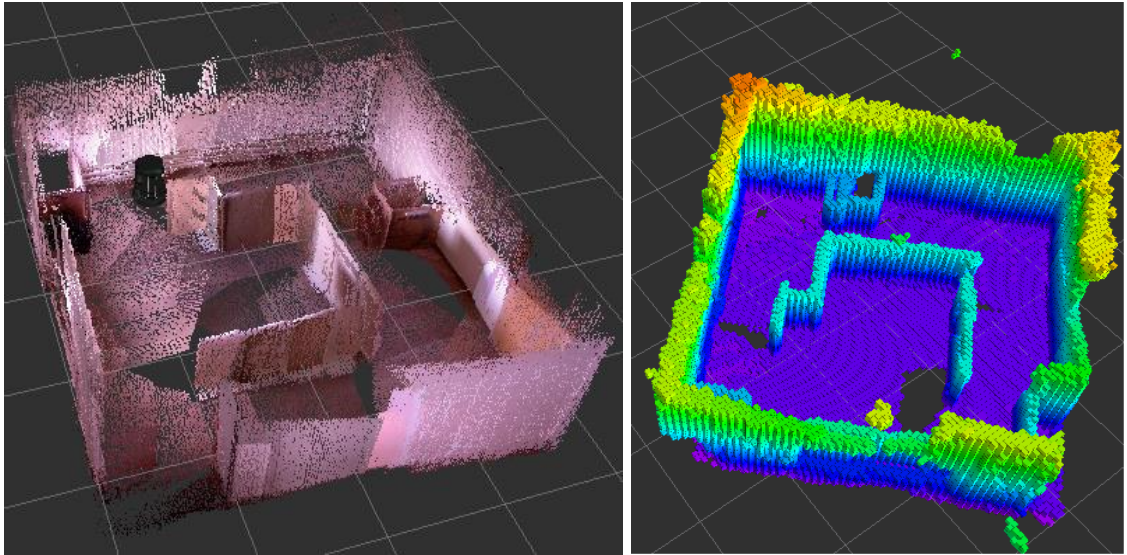
$$\begin{aligned} \log p(X_T, m | Z_T, U_T) = & \text{sabit} + \sum_t [x_t - f(u_t, x_{t-1})]^T R_t^{-1} [x_t - f(u_t, x_{t-1})] \\ & + \sum_t [z_t - h(x_t, m)]^T Q_t^{-1} [z_t - h(x_t, m)] \end{aligned} \quad (3.33)$$

Burada x_t durumu $f(\cdot)$ hareket modelini, z_t ölçümü, $h(\cdot)$ ölçüm modelini, R_t ve Q_t ise sırasıyla hareket ve ölçüm kovaryans matrislerini ifade etmektedir. Buna göre hatayı en küçükleyen düğüm konfigürasyonunun elde edilmesi amaçlanır. Graf SLAM probleminin çözümünde gradyen iniş, karmaşık gradyen gibi birçok optimizasyon metotları kullanılabilir. Bu metot EKF-SLAM gibi yöntemlere göre daha fazla özellik

barındırabilmesinden dolayı daha büyük haritaların elde edilmesine imkan tanır (Thrun ve Leonard, 2008; Grisetti ve ark., 2010).

3.2.4. Görsel SLAM (vSLAM)

SLAM algoritması ilk zamanlarında dönüştürücüler, lazer, sonar, GPS, IMU vb. birçok sensör kullanılarak gerçekleştirilmiştir (Aulinas ve ark., 2008). Erken dönem çözümlerler genellikle olasılıksal bir model çerçevesinde incelenmiş ve durum tahminlerinin gözlemlerle güncellenmesi ile algoritmanın gerçekleştirilmesi sağlanmıştır. Ancak ilerleyen yıllarda kameraların kullanılması ile SLAM problemi farklı bir bakış açısıyla irdelenmeye başlamıştır. Buna göre SLAM çözümünde dış sensör olarak kameraların kullanıldığı algoritmalar görsel SLAM (vSLAM) başlığı altında incelenmeye başlanmıştır (Sola, 2007). Böylece SLAM görüntü işleme ve bilgisayar görmesi alanının ilgi odağı olmuştur (Fuentes-Pacheco ve ark., 2015). Bununla ilgili detaylı bir literatür özetinden kaynak araştırması kısmında bahsedilmiştir. Örnek bir vSLAM algoritması çıktısı Şekil 3.7'deki gibi olabilir (RTAB-Map (Labbé ve Michaud, 2019) ve OctoMap (Hornung ve ark., 2013) yöntemleri).



(a) RTAB-Map algoritma sonuç haritası

(b) OctoMap algoritma sonuç haritası

Şekil 3.7. Korkmaz; ve ark. (2019) gerçek zamanlı vSLAM algoritmalarının karşılaştırılması çalışmasında elde edilen sonuç 3B haritaları; sırasıyla (a) RTAB-Map ve (b) OctoMap algoritma sonuçları

vSLAM algoritması doğrudan, özellik tabanlı ve RGB-D kamera tabanlı başlıklara göre tasnif edilebilmektedir. Bu algoritma, özelinde farklı alt başlıklar içermesine rağmen genellikle başlangıçlandırma, izleme, haritalama, yeniden konumlama ve global harita optimizasyonu basamaklarından oluşmaktadır. Bu başlıkların herbirinin ya da bütününün geliştirilmesine dair birçok araştırma yapılmaktadır (Taketomi ve ark., 2017).

Başlangıçlandırma aşaması genel olarak global koordinat sisteminin tanımlanması ile ilgilidir. Tanımlanan bu koordinat sistemin içerisinde ortamdan alınan veriler işlenerek harita oluşturulur. *İzleme* aşamasında ise oluşturulan haritaya göre kameranın pozisyonu tahmin edilir. Bu aşama klasik yaklaşımdaki konumlama basamağına benzemektedir. *Haritalama* aşamasında ise bilinmeyen ve yeni keşfedilen alanlar başlangıçlandırma aşamasında oluşturulan global koordinat sistemine göre tanımlanarak ortam ile ilgili harita elde edilmiş olur. Bu temel basamaklara ilave olarak algoritmanın daha gürbüz olması birçok durumda vazgeçilmez bir unsurdur. Buna örnek olarak kameranın izlediği özelliklerde oluşan süreksizlik verilebilir. Bu, robot kaçırılma problemi olarak bilinen duruma benzemektedir. Kameranın görüş açısının darlığı ve hareket hızındaki ani değişimler göze alındığında böyle bir duruma karşı gürbüz bir performans sergilemek önemli olacaktır. *Yeniden konumlama* modülü sayesinde bu türde problemlere karşı gürbüzlük sağlanabilir ve izleme aşamasında başarısızlık olsa bile mevcut haritadan konum bilgisi tekrar elde edilebilir. *Global harita optimizasyonu* ise çalışma sürecinde gerçekleşen artık hatalar ve buna bağlı birikimli hataların giderilmesi ile ilgilidir. Bununla ilgili en önemli durumlar kapalı döngülerin elde edilmesi ile daha önce görülen bir yerin yeniden gözlemlenmesidir. Bu sayede birikimli hata azaltılabilir.

Bu bilgiler ışığında vSLAM'in gerçekleşmesi için temel olarak görsel odometri ve hareketten iskelet oluşturma (SfM) problemleri çözülmelidir. Odometri genel manada pozisyon değişimlerinin zamana bağlı olarak sensör verilerinden alındığı bilgidir. Örneğin bir robotun tekerine yerleştirilmiş döner kodlayıcı sensörü robotun ne kadar hareket ettiği ile ilgili bilgiyi verebilmektedir. Benzer şekilde görsel odometri ise kamera hareketlerindeki farkların kullanılması ve bu sayede konum ve yönelim bilgilerinin elde edilmesidir. Bu veri, vSLAM algoritmalarında kullanılan temel bir girdi olup vSLAM bu girdiye ilave olarak global harita optimizasyon modülünün eklenmesi ile elde edilir. SfM ise 2B kamera hareketlerinden fotometrik bir anlamda model çıkarılmasıdır. vSLAM ve SfM'in düşünce yapısı birbirine çok benzemektedir. Her ikisi de lazer sensörler yerine kameraların kullanımı ile haritanın oluşturulması temeline dayanmaktadır. Birçok

çalışmada vSLAM ve gerçek zamanlı SfM algoritmaları arasında kesin bir fark belirtilememiştir. Bu yüzden bazı çalışmalar vSLAM'i gerçek zamanlı SfM olarak da nitelendirmektedir (Taketomi ve ark., 2017). Öte yandan vSLAM yerel bir haritalama çözümü olmasına rağmen SfM daha global ölçekte bir çözüm sunmaktadır. Buna en iyi örnek Google tarafından sunulan cadde görüntüleri olarak verilebilir. Ayrıca SfM'de sıralı bir resim olmasına gerek yokken vSLAM'de sıralı resimlere ihtiyaç duyulur.

Literatürde birçok vSLAM algoritmaları tanımlanarak farklı problemlerin çözümleri elde edilmiştir. vSLAM metotları özellik tabanlı veya yoğunluk tabanlı olarak genel iki kategoriye ayrılabilir. Bunlara ilave olarak, derinlik bilgisi verebilen RGB-D kameralar ile yapılan çözümler de ayrı bir sınıf olarak düşünülebilir.

Özellik tabanlı algoritmalar klasik dönemde yapılan çözümlere benzemektedir. Ortamdan çıkarılan özelliklere bağlı olarak ortamın özellik haritası elde edilir. Bu konudaki ilk çalışmalar tek gözlü kameraların kullanımı ile başlamış sonraları bu metot MonoSLAM olarak adlandırılmıştır (Davison, 2003; Davison ve ark., 2007). Özellik tabanlı vSLAM metotları, PTAM ve onun iyileştirilmiş bir versiyonu olan ORB-SLAM algoritmaları ile geliştirilmiştir (Klein ve Murray, 2007; Mur-Artal ve ark., 2015). Bu algoritmalar da kendi içerisinde filtre yaklaşımı veya demet düzenlemesi yöntemini kullanmasına göre sınıflandırılabilir. Buna paralel bir şekilde doğrudan metotlarda haritalar, yoğunluk bilgisini içerecek şekilde oluşturularak elde edilir. DTAM ve türevleri ile LSD-SLAM bunların en önemli örnekleridir (Newcombe ve ark., 2011; Engel ve ark., 2014). Direkt metotların en önemli özelliklerinden birisi derinlik değerlerinin her bir piksel değeri için anahtar karede tahmin edilmesidir. Bunların yanı sıra vSLAM için üçüncü bir kategori olarak RGB-D kameraların kullanıldığı uygulamalar düşünülebilir. Bu kameralar ilk olarak oyun konsolları için tasarlanmış olmasına rağmen robotik ve bilgisayar görmesi alanlarında fazlasıyla ilgi görmüştür. Microsoft firmasının ürettiği Kinect ve ASUS firmasının ürettiği Xtion sensörleri ve türevleri bu türde kameralara birer örnektir. Bu kameralar görüntü bilgisinin yanı sıra derinlik verisi de sunmaktadır. Bu kameralar sayesinde vSLAM algoritmaları için derinlik tahmini sorunu çözülebilmektedir. Ayrıca 3B olarak metrik bir şekilde koordinat sistemi tanımlaması mümkün olmaktadır. Genellikle iteratif yakın nokta (ICP) algoritmaları ile kamera hareketleri tahmin edilebilmektedir. Ancak bu kameralarla ilgili en büyük handikap dış ortam kullanımlarına müsait olmayışıdır. KinectFusion, SLAM++, RGB-D SLAM algoritmaları bu alanda geliştirilmiş çalışmalardan bazılarıdır (Fuentes-Pacheco ve ark., 2015; Taketomi ve ark., 2017).

3.3. SLAM ile İlgili Diğer Yaklaşımlar

SLAM probleminin ortaya konulması ve çözümlerinin sunulmasından bu yana farklı türde yaklaşımlarda SLAM başlığına konu olmuştur. Örneğin, aynı veya farklı türde çoklu robotların kullanıldığı durumlarda SLAM probleminin çözümü (robot koordinasyonu, harita birleştirme), benzer şekilde robotun nereye gideceğine karar verdiği aktif SLAM yaklaşımı bunlardan bazılarıdır.

3.3.1. Harita birleştirme

Haritalama veya SLAM'e konu olan haritalar farklı yöntem ve algoritmalarla çözülmeye çalışılmıştır. Ancak, çalışmaların birçoğu tek bir robotun problemleri nasıl çözeceği üzerinedir. Gelişen teknoloji ve ihtiyaçlar doğrultusunda birden fazla robotun kullanımına ihtiyaç duyulan birçok durum vardır. Örneğin, bir felaket zamanında robotlar kullanılmak istenildiğinde ortam ile ilgili haritalara ihtiyaç duyulacaktır (Topal ve ark., 2013). Bu haritaların elde edilmesi için tek bir robotun kullanımı yerine birden fazla robotun kullanılması ile zaman kazanılabileceği aşikardır. Ancak her bir robot kendi yerel haritasını üretecek ve bunların global olarak nasıl birleştirilmesi gerektiği üzerine bir metodolojiye ihtiyaç duyulacaktır. Bununla birlikte, bazı durumlarda tek bir robot farklı yerlerin kısmi haritalarını üretebilir ve bu haritalar daha sonra birleştirilmek istenebilir (Carpin ve ark., 2005). Bu açıdan ele alındığında SLAM başlığı altında harita birleştirilmesi konuları farklı çalışmalarda ele alınmıştır. Gerek eş zamanlı gerekse kısmi haritalar elde edildikten sonra haritaların birleştirildiği farklı türde algoritmalar geliştirilmiştir. Bir diğer önemli husus ise elde edilen kısmi haritaların türüdür. SLAM'in sonucunda elde edilen harita özellik, yoğunluk veya meşguliyet ızgara hücreleri tabanlı olabilmektedir. Farklı türdeki haritalara göre algoritmaların içeriği değişebilmektedir. Genel bir şekilde aşağıdaki gibi bir tanımlama yapılabilir.

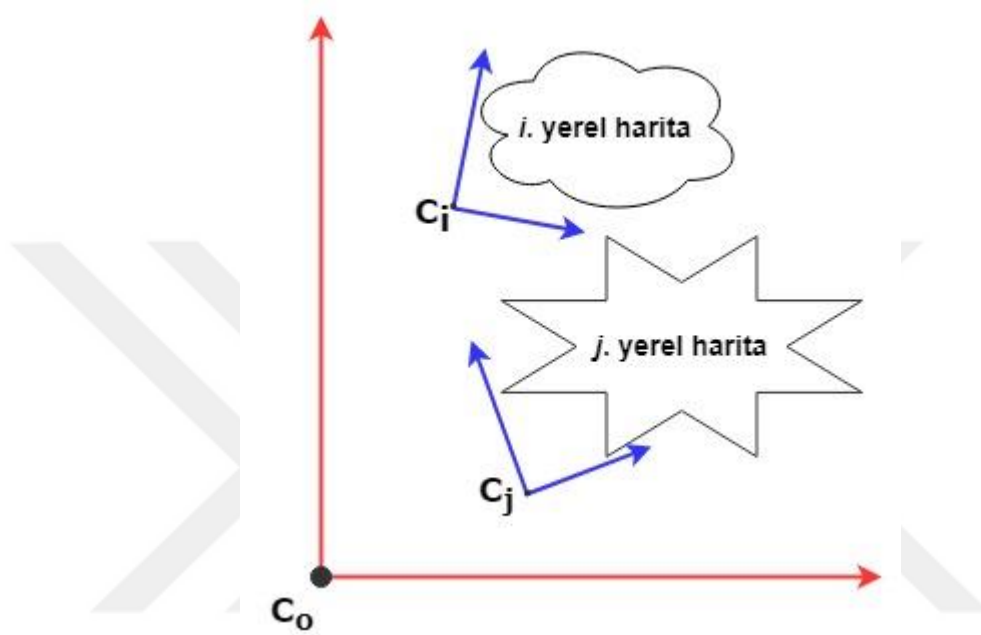
Tanım: N ve M pozitif reel sayılar olmak üzere $N \times M$ boyutlu bir harita fonksiyonu π tanımı, $\pi : [0, N] \times [0, M] \rightarrow \mathbb{R}$ şeklindedir.

Buna göre global koordinat sisteminde yerel harita (Eşitlik (3.34)) ve bunların birleşimi (Eşitlik (3.35)) aşağıdaki gibidir.

$${}^w_i\pi = {}^w_iT\pi_i; i = 1, \dots, n \quad (3.34)$$

$$\pi = \pi_i \oplus \pi_{i+1}, i = 1, \dots, n - 1 \quad (3.35)$$

Burada w_iT C_i merkezli i . yerel haritanın C_o merkezli global koordinatlara dönüşümü için gerekli dönüşüm, \oplus operatörü ise yerel haritaların birleşimi için gerekli olan algoritma veya matematiksel işlemlerdir (Şekil 3.8).



Şekil 3.8. Kısmi haritaların global koordinatlarda gösteriminin temsili

Bu konuda klasik bir tek durumlu EKF-SLAM algoritmasının çoklu duruma genişletilmesi gibi düşünülebilecek bir algoritma “robotların buluşması” başlığı altındaki bir çalışmada ele alınmıştır (Zhou ve Roumeliotis, 2006). Problemin temelinde robotların şans eseri veya bilinçli bir şekilde bir noktada görüşmesi vardır. Böylece birbirleri ile kesişen harita parçacıklarının olma zorunluluğu ortadan kalkmıştır. Robotlar buluşma zamanında birbirleri ile göreceli pozisyonlarını ve bunlarla ilgili belirsizlikleri paylaşırlar ve buna göre haritalar arası dönüşüm hesaplanır. Burada karşılaşılabilecek önemli bir problem, bir özelliğin farklı iki yerel haritada farklı yerlerde gösterilmesidir. Bunun giderilmesi için Mahalanobis mesafe ölçüsü tabanlı bir test uygulanarak eşik değerine göre ilgili özelliğin durum matrisinden çıkarılmasına veya kalmasına karar verilir.

Bunun yanı sıra haritaların bir resim gibi ele alındığı ve buna bağlı olarak resim tescilleme algoritmalarının kullanılması da bir diğer harita birleştirme yöntemidir. Ancak bu yöntemin uygulanabilmesi için yerel haritalar arası örtüşen bölgeler olması

gerekmektedir. Elibol ve ark. (2014) çalışmasında, su altından çoklu robotlarla elde edilen görüntüler birbiri ile birleştirilerek bu yöntemle harita birleştirilmesi yapılmıştır.

3.3.2. Aktif SLAM

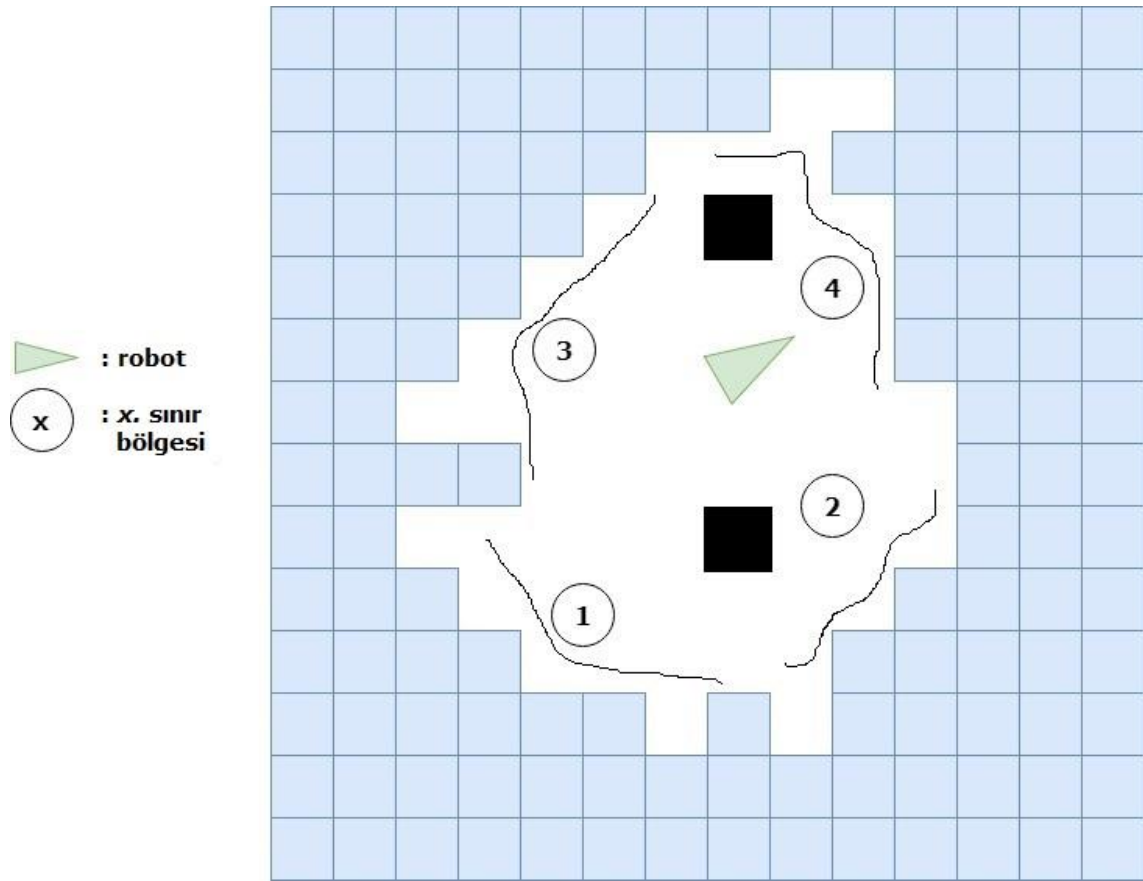
Robotların bir ortamda hareket ettirilmesi genellikle bir kullanıcı girdisi üzerinden olmaktadır. Örneğin, su altı keşif araçları, bomba imha robotları gibi uygulamalarda kablolu veya kablosuz bir şekilde robottan veri alan kullanıcı, robotu istediği yönde hareket ettirerek istenilen görevin yerine getirilmesini sağlar. Ancak bazı durumlarda robotun ortamda kendi kendisine keşif yapması ve robotun çevreyi tanıması beklenebilir. Örneğin, robot ile etkileşimin olmadığı bir felaket durumu veya birden fazla robotun bir görevi yerine getirmesi ile ilgili bir durum söz konusu olduğunda robotların kendi kendine ortamda keşif yapması beklenir. Bu sorun genellikle keşif algoritmaları başlığı altında incelenir. SLAM'den bağımsız olarak bu konudaki ilk çalışmalardan birisi Yamauchi (1997)' dir. Çalışmanın temel fikri robotun çevresi ile ilgili en çok yeni bilgi kazanabileceği yönler yönelmesi üzerinedir. Bunu yapmak için tanımladığı bölgenin sınırlarına doğru hareket etmektedir. Buna göre her bir keşif ile yeni bilinmeyen bölgeler ortaya çıkacak ve buna bağlı olarak bütün ortamların keşfi sağlanmış olacaktır. Şekil 3.9 incelirse robotun ortamda keşif yaptığı ve buna bağlı olarak bazı ortamları keşfettiği (boş veya koyu alanlar) bazı ortamlar ile ilgili bilgi sahibi olmadığı görülmektedir (mavi renkli hücreler). Robotun keşfettiği boş/dolu alan ile keşfedilmemiş bölgeler arasında şekilde gösterildiği gibi farklı sınır bölgeleri oluşmaktadır. Buna bağlı olarak makalede önerilen metoda göre robotun en yakınındaki keşfedilmemiş sınır bölgesi olan 4 numaralı alana doğru hareket edecektir. Robot üzerindeki lazer, sonar veya kızıl ötesi sensörler sayesinde yeni hücreleri keşfedecek ve peşi sıra yaptığı keşifler ile kendi kendisine bir ortamda hareket etmesini sağlayacaktır. Her bir hücreye başlangıç olarak 0.5 olasılık değeri atanmış ve keşiflerle bu olasılığın değişip değişmediği araştırılmaktadır. Buna bağlı olarak hücre açık, meşgul veya bilinmeyen olarak işaretlenmektedir.

Açık: $p(m_i) < 0.5$

Meşgul: $p(m_i) > 0.5$

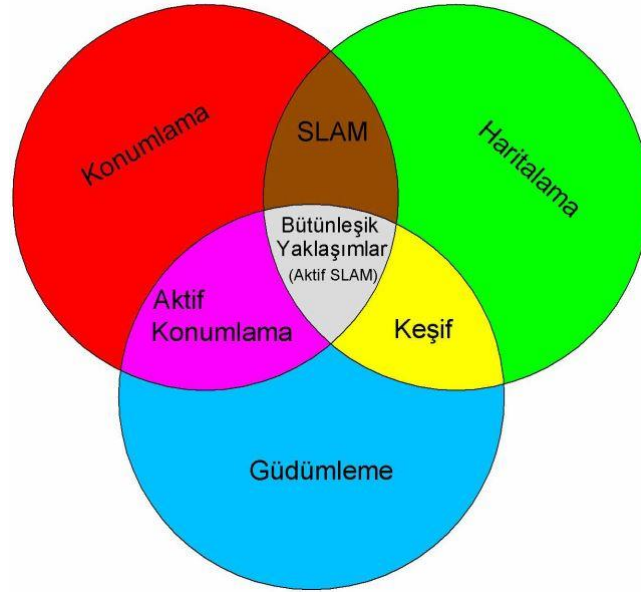
Bilinmeyen: $p(m_i) = 0.5$

Burada $p(m_i)$ i . hücrenin meşgulliyet olasılığını göstermektedir.



Şekil 3.9. Bir ortamda robot tarafından yapılan keşifler ve farklı sınır bölgeleri

Gezgin robotlar için SLAM algoritmalarının keşfi önemli bir gelişme olmuştur. SLAM sayesinde robotlar otonomi kazanmış ve bilmediği bir ortamda kendisini konumlamayı ve aynı zamanda ortamın haritasını elde etmeyi başarabilmiştir. Ancak robotlara tam otonomi SLAM algoritmalarına ilave olarak navigasyon ve keşif algoritmalarının da eklenmesi ile sağlanabilmektedir. Çünkü SLAM algoritmaları robota tam olarak nereye ve nasıl gideceği ile ilgili bir bilgi sağlamamaktadır. SLAM probleminde kullanıcı girdisi, rassal hareket veya önceden seçilmiş geçiş noktaları ile yol bilgisi tanımlanmaktadır. Haritalama, konumlama ve navigasyon-keşif bileşenlerinin bir arada olması ile robot bir ortamda kendisini konumlayabilmekte, haritayı elde edebilmekte ve nereye nasıl gideceği sorularının cevabını bulabilmektedir (Şekil 3.10). Bu üç alanın kesişimi literatürde bütünleşik yaklaşımlar bir diğer bakış açısıyla aktif SLAM olarak adlandırılmıştır (Stachniss, 2009). Aktif SLAM problemi SLAM algoritmalarına göre daha güncel bir alan olup günümüzde de birçok yönüyle ele alınmaktadır (Zamora ve Yu, 2013; Cadena ve ark., 2016).



Şekil 3.10. Haritalama, konumlama ve hareket kontrol başlıkları arasındaki ilişki (Makarenko ve ark., 2002)

Aktif SLAM alanında genellikle SLAM algoritmasının belirsizliğinin azaltılması, robot tarafından keşfedilen ortamın en fazla olması ve görevin yerine getirilmesi ölçütleri göz önüne alınır ve buna bağlı bir maliyet analizi yapılır (Eşitlik (3.36)) (Carrillo ve ark., 2012).

$$J = \sum_i \alpha_i U_i + \sum_i \beta_i T_i \quad (3.36)$$

i : Planlanan yol sayısı

U_i : Sistem parametrelerindeki belirsizlik değeri (kovaryans matrisinin izi, determinanı gibi skaler bir değer)

T_i : Diğer beklenti değerleri (yol uzunluğu, güdüm süresi, enerji değeri vb. olup belirtilen değerlere göre skaler bir değer)

α, β : Ayar parametrelerini kapsayan ağırlık katsayıları

Eşitlik (3.34)'te J amaç fonksiyonu olup en iyi değerinin bulunması istenir. α ve β ise kullanıcı tarafından isteğe bağlı olarak seçilen ve eşitliği iki parçaya ayırarak ilgili kısımların farklı etkiler vermesini sağlayan ağırlık katsayılarıdır (Mihaylova ve ark., 2003). Bu tanımlamalara göre aktif SLAM problemi J fonksiyonunu en iyi yapmaya çalışan π reçeteleri olarak düşünülebilir. π reçetelerinin değerlendirilmesi için optimal

deney tasarım teorisinden (TOED) faydalanılabilir. Bu teori, temelinde bilinmeyen gürültülere karşı çıkış parametrelerinin varyansını hesaplar ve buna bağlı olarak en iyi π reçetesi elde edilebilir. Burada çözüm olarak *A-opt* (matrisin izini ve buna bağlı toplam varyansı en küçükleme), *D-opt* (matris determinantının logaritma değerini en büyükleme), *E-opt* (matrisin tersinin özdeğerlerini en küçükleme) gibi ölçütlerden yararlanan farklı çalışmalar vardır.

Bu başlıkta tanımlanan aktif SLAM yapısı içerisindeki SLAM bloğu, olasılıksal tabanlı yapıları temel almıştır. Halihazırda geliştirilen birçok vSLAM algoritması ile entegre edilmesi gereken aktif SLAM algoritmaları için farklı bir strateji düşünülmesi gerekebilir.

Bölüm 3'te genel olarak klasik ve modern dönem SLAM yöntemleri üzerine tanımlamalar yapılmış ve detaylı açıklamalar verilmiştir. Sonraki iki bölümde, bu tez kapsamında yapılan uygulamalar, geliştirilen yöntemler ve yapılan çalışmalardan elde edilen sonuçlar üzerinde durulacaktır.

4. MATARYEL, YÖNTEM, GELİŞTİRİLEN ALGORİTMALAR VE UYGULAMALAR

4.1. Giriş

Önceki bölümlerde SLAM problemi ile ilgili tarihçe, matematiksel altyapı ve genel problemler tanımlamıştır. Bu bölümde tez kapsamında yapılan farklı çalışmalar geliştirilen algoritmalar ve uygulamaların açıklanması ile bunlar için kullanılan materyallerin incelenmesi üzerinde durulacaktır.

4.2. Robot İşletim Sistemi (ROS)

Robot İşletim Sistemi (ROS)¹ her ne kadar tanımında bir işletim sistemi olarak belirtilse de aslında genel manada bir yazılım sistemi olarak tanımlanabilir. ROS'un altında yatan temel düşünce “tekerleği yeniden keşfetmeme” üzerine kuruludur. Birçok araştırmacı robotik problemler üzerine çalışmaktadır. Bunun için birçok algoritma ve ilgili yazılımlar geliştirilmektedir. Geliştirilen algoritmaların kaynak kodları paylaşılsa bile yeni başlayan birçok araştırmacı için bu kod parçacıklarının anlaşılması ve üzerine geliştirmeler yapılması çok uzun süreler alabilmektedir. Çoğu zaman araştırmacılar kendi kodları ve çözümlerini geliştirilmekte ve tekrar tekrar aynı problemle ilgili çözümler üretmektedir. Bu düşünceyle, geliştirilen yazılımların bir paket haline getirilmesi buna göre yeni kullanıcının bile küçük değişikliklerle ilgili çözümlere aşina olması ve geliştirmesi fikri ROS platformu için bir temel oluşturmuştur. Genel bir tanımlama ile ROS, robot yazılımları ile ilgili esnek bir çerçeve sağlayarak farklı araçlar ve kütüphaneler kullanılmasına izin veren bir yazılım sistemidir. Karmaşık robotik sistemlerin çözümü için belirli teamüller çerçevesinde çözümler sunmaktadır.

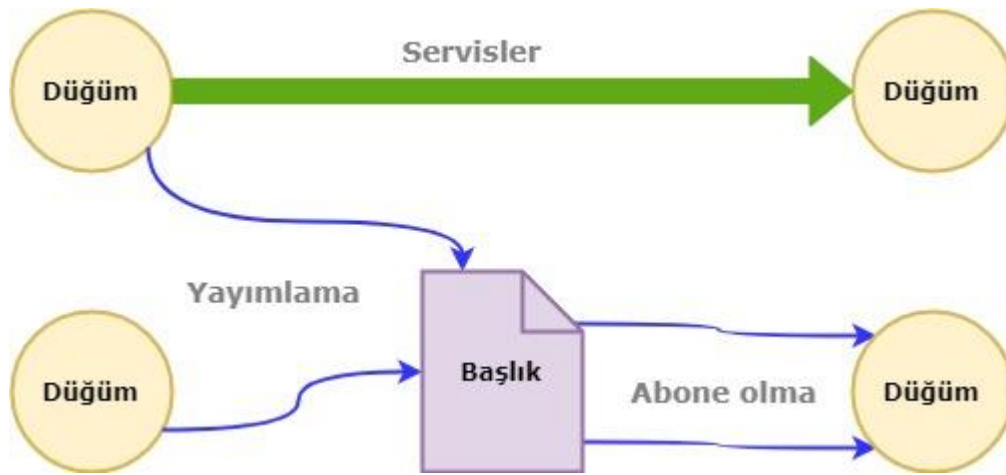
ROS yazılım sistemi ilk olarak 2007 yılında Standford Yapay Zeka Laboratuvarında ortaya konulmuş ve daha sonraları Willow Garage robot araştırma enstitüsü ve OSR vakfi bünyesinde devam ettirilmiştir. “Tekerleği yeniden keşfetmeme” fikri birçok araştırmacı tarafından benimsenmiş ve ROS adeta robotik platformlar için bir temel haline gelmiştir. Birçok firma ürettiği robotları veya sensörleri ROS uyumlu hale getirmeye başlamış ve yeni ürünlerini ROS uyumlu olarak çıkarmaya başlamıştır. Her

¹ <https://www.ros.org/>

geçen gün ROS uyumlu robot, sensör sayısı hızla artmaktadır. 2010 yılında “Box Turtle” sürümü ile başlayan dağıtımların çok kısa sürede farklı versiyonlarla yenilenmiş ve “Melodic Morenia” ile mevcut koşullarda 2023 yılına kadar destek alacak sürümü yayınlanmıştır. 2012 yılından bu yana her yıl farklı bir ülkede ROSCon isminde ROS ile ilgili son gelişmelerin aktarıldığı ve farklı seviyeden ilgililerin katıldığı çalıştaylar yapılmaktadır. Bununla birlikte ROS aktif bir topluluğa sahip olup soru cevap platformunda 40 binin üzerinde sorunun cevabı verilmiştir. Özellikle Raspberry Pi gibi küçük bilgisayarların geliştirilmesi ve bunlar üzerine ROS’un kurulmasıyla birçok yeteneğe sahip robotların geliştirilmesi mümkün olabilmektedir. Ayrıca ROS’un akademik, ticari, endüstriyel ve hobi amaçlı birçok kullanıcıya hitap etmesinden dolayı yakın gelecekte robotik için temel bir standart haline gelmesi kuvvetle muhtemeldir.

ROS’un en büyük üstünlüklerinden birisi kullanıcıları herhangi bir yazılım dilini kullanmaya zorlamamasıdır. C++, Python, Java gibi diller sıklıkla kullanılmaktadır. Bunun yanı sıra bir kod parçası Python diliyle yazılmışken bir diğer kod parçası başka bir dilde yazılmış olabilir ve bunların birbiriyle çalışması konusunda herhangi bir problemle karşılaşmamaktadır. Bir diğer manada ROS, robotik yazılımlarını modüler bir yapıya kavuşturmuş ve bu sayede lego parçaları gibi ihtiyaç duyulan kısmın kullanılmasına imkan tanımıştır. Ayrıca Rviz modülü sayesinde sensör verisi, robotun bilgileri gibi farklı durumların görselleştirilmesi mümkündür. Genellikle ROS’un tavsiye edilen çalışma ortamı Linux/Ubuntu üzerinde olmakla birlikte son yıllarda MATLAB programında da bazı ROS uygulamalarının kullanılmasına başlanılmıştır. Ayrıca Windows 10 üzerinde Linux alt sistemi olarak (WSL) ROS’un kullanımına imkan tanınmaya başlanılmıştır.

ROS genel manada herhangi bir görevi yerine getiren düğümler ve bu düğümler arası yapıların oluşturduğu bir çalışma ortamıdır (Şekil 4.1). Düğümler, farklı hesaplamalar yapabilirler. Örneğin, bir düğüm yüz tanıması ile ilgili iken diğer düğüm sensör verilerinin okunması ile ilgili olabilir. Bu düğümler arası haberleşme ise farklı türde olabilir; servisler ve başlıklar. Başlıklar, düğümler arası iletişimi sağlarlar ve bir düğüm bir bilgi yayımlarken diğer bir düğüm eğer bu bilgiye ihtiyaç duyarsa ilgili başlığa abone olarak o bilgiyi alabilir (Quigley ve ark., 2015).



Şekil 4.1. ROS'un çalışma altyapısına dair genel bir çalışma süreci

Örneğin “*usb_cam*”² ROS paketi usb kameralarla ilgili bir ROS sürücüsüdür. Bu paket “*usb_cam_node*” isimli düğümden usb girişlerine bağlanmış kameraların görüntülerini almaya yarar ve “*<kamera_ismi>/image*” başlığından ilgili veriyi “*sensor_msgs/Image*” tipinde yayımlar. Bir diğer paket olan “*image_rotate*”³ ise kendisine gelen görüntüleri kamera koordinatıyla eş düzlemde olacak şekilde hizalamaya çalışır. Bunun için “*image_rotate*” isimli bir düğümüne sahip olup bu düğüm bir önceki paketten yayımlanan “*<kamera_ismi>/image*” isimli başlığa abone olarak verileri işler ve “*rotated/image*” isimli farklı bir başlıktan “*sensor_msgs/Image*” türünde veriyi yayımlar. Bu şekilde bu veri de farklı bir paketin girişi olabilir ve ihtiyaç duyan düğüm bu veriye abone olup istenilen herhangi bir işlevi gerçekleyebilir.

Çoğu zaman gerçek uygulama denemelerinden önce gerek görselleştirme gerekse yazılımın kontrolü yapılmak istenmektedir. Bunun için V-Rep, Webots, Gazebo, Microsoft Robotics Developer Studio gibi birçok farklı robot simülatörleri bulunmaktadır. Bu benzetim araçlarından birisi olan Gazebo, gerek birçok sensör ve nesnelere temsili gerekse ROS ile uyumlu olması açısından son yıllarda sıklıkla robotik araştırmacıları tarafından tercih edilmektedir. ROS-Gazebo sayesinde bir robotun kinematikleri, çarpışma sınırları, ağırlık merkezi vb. birçok bilginin simüle edilmesi mümkündür. Gazebo “*gazebo_ros_pkgs*” paketleri sayesinde ROS üzerinde gerekli veri alışverişini sağlayabilir. Bu tez çalışmasında yapılan uygulamalarda ROS, Gazebo ve Rviz yazılımlarından faydalanılmıştır.

² http://wiki.ros.org/usb_cam

³ http://wiki.ros.org/image_rotate

4.3. Sensör Tabanlı Yöntemler ile İlgili İnceleme

SLAM çalışmalarında sensörlerden alınan veriler önemli bir rol oynamaktadır. Bu sensörler iç ve dış algılayıcılar olarak gruplandırılabilir. Özellikle dış algılayıcı sensörler sayesinde filtre tabanlı yöntemler için tahminler güncellenerek durumlarla ilgili kestirimler yapılmaktadır. Bu kapsamda klasik dönem SLAM algoritmaları için sonar, 2B ve 3B lazerler, kamera vb. sensörler kullanılmaktadır. Durum tahmini ve tahminlerin güncellenmesi ile ilgili detaylı açıklamalar Bölüm 3'te verilmiştir.

Lazer sensörler birçok uygulamada mesafe bilgisi için yüksek doğruluklu bir ölçüm sunmaktadır. Ancak lazer sensörlerin pahalı olması, fazla güç gereksinim gerektirmeleri gibi olumsuzlukları robotik uygulamalarda farklı türde sensör arayışlarına yol açmıştır. Bu noktada en ilginç yaklaşımlardan birisi RGB-D sensörlerin geliştirilmesi ile ortaya çıkmıştır (Şekil 4.2).



(a) Rplidar A1M8

(b) Microsoft Kinect versiyon 1.0

Şekil 4.2. Korkmaz ve ark. (2018) çalışmasında yararlanılan farklı iki sensör; (a) 360° derece 2B tarama yapabilen lazer sensör ve (b) RGB-D kamera

RGB-D sensörler görüntü bilgilerinin yanı sıra ortamla ilgili derinlik bilgisi de sunmaktadır. Bu özellikleri sayesinde robotik alanında önemli çalışmalara imza atılmıştır. Bu kameralarla ilgili bir diğer fikir ise derinlik haritasının lazer sensör verisi olarak kullanılması ile ilgilidir. Elde edilen nokta bulutu verisi gerekli filtre ve düzeltme işlemleri ile örneklenmiş bir 2B lazer verisi gibi kullanılmaktadır. Görüş aralığı, fiyat gibi parametreler göz önüne alındığında Slamtec firmasına ait bir lazer ve Microsoft firmasına ait bir RGB-D sensör karşılaştırması Çizelge 4.1'de paylaşılmıştır.

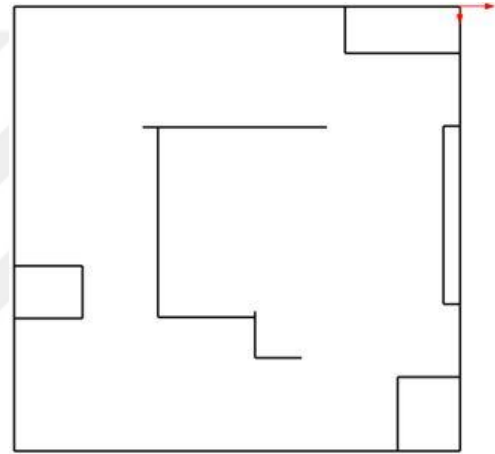
Çizelge 4.1. Çalışmada kullanılan lazer sensör ve RGB-D kamerasının genel bir karşılaştırılması

Özellik	Rplidar	Kinect
Fiyat	~ \$450	~ \$150
Görüş mesafesi	~ 6 metre	~ 5 metre
Görüş açısı (FoV)	360 derece yatay	43 (düşey) & 57 (yatay) derece

RGB-D kameralarla elde edilen derinlik verilerinin 2B lazer verilerine dönüştürülmesi fikrinin gerçek zamanlı SLAM uygulamalarında nasıl çalışacağı ile ilgili tez kapsamında bir deneme yapılmıştır. Şekil 4.3a’da gösterilen bir ortam, robotun görüş mesafesi seviyesinde farklı bölümlere ayrılmıştır.



(a) Bahsedilen çalışmaya ait uygulama ortamı

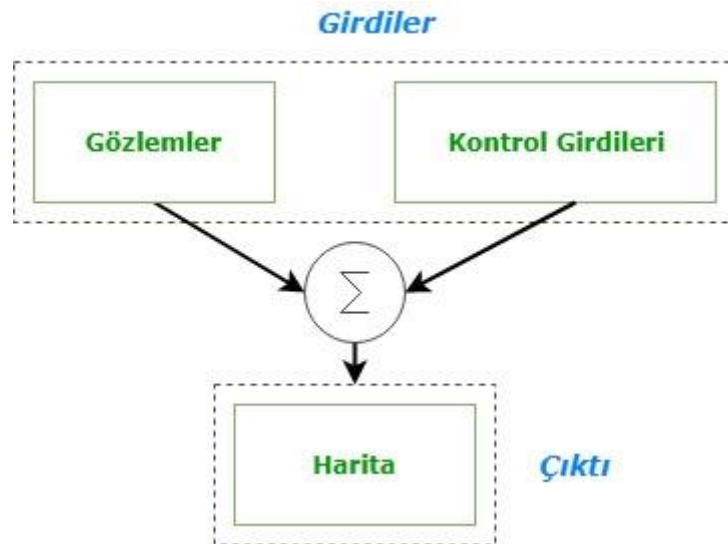


(b) Aynı ortam için plan

Şekil 4.3. Bir SLAM algoritması üzerinde farklı türdeki sensörlerin karşılaştırılması uygulamasına dair denemelerin yapıldığı ortamla ilgili resimler; (a) Uygulama ortamı görüntüsü ve (b) üstten görünüm

Denemeler için SLAM uygulamalarında sıklıkla kullanılan parçacık filtresi tabanlı ve meşguliyet haritası çıktısı üreten *gmapping*⁴ ROS paketi, SLAM algoritması olarak kullanılmıştır. İlgili karşılaştırma çalışmasına dair genel algoritma taslağı Şekil 4.4’te gösterilmiştir. Gözlem girdileri olarak ilk uygulamada lazer sensörden alınan veriler kullanılmıştır. İkinci uygulamada ise Kinect sensöründen alınan veriler ölçüm değerleri olarak algoritmaya verilmiştir. Klasik SLAM çalışmalarında olduğu gibi robot hareket modeli ve kontrol girdileri ile durum tahmini yapılmış bu tahminler gözlemlerle güncellenerek nihai durum elde edilmiştir.

⁴ <http://wiki.ros.org/gmapping>



Şekil 4.4. Bir SLAM algoritması üzerinde farklı türdeki sensörlerin karşılaştırılması uygulamasına dair algoritma girdi ve çıktılarına ait blok diyagram

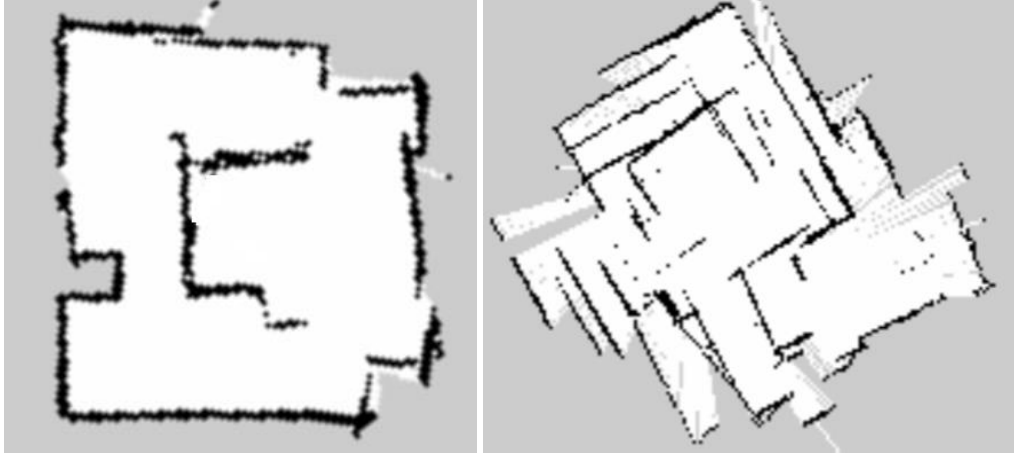
Gözlem bilgisi iki farklı sensörden alınmakla birlikte her ikisi de 2B örnekleme lazer verisi gibidir. RPLidar sensörü ile bu veri 360° görüş açısı ile elde edilirken Kinect sensörü ile daha kısıtlı bir alanda oluşmaktadır. Kinect sensöründen elde edilen nokta bulutu kümesi, gerekli yazılımlarla örnekleme bir 2B lazer verisi gibi algoritmanın girişlerine uygulanmıştır.

Bunların yanı sıra algoritmanın başarımını etkileyen parçacık sayısı parametresinin değişimi de göz önüne alınmıştır. Bu kapsamda iki farklı parçacık sayısına göre algoritmalar test edilmiştir. Her iki deneme için de bir kullanıcı tarafından robotun ROS *turtlebot_teleop*⁵ paketi kullanılarak yönlendirilmesi sağlanmıştır. Çalışmada Turtlebot-II⁶ olarak bilinen robot kullanılmıştır. Turtlebot-II robotu düşük maliyeti, mobilite özelliği, modüler yapısı, farklı algoritmaların denenebilmesi için sağladığı platform ve en önemlisi de açık kaynak kodlu uygulamalara imkan vermesi gibi özelliklerinden dolayı robotik camiasında sıklıkla kullanılan bir robot olup diferansiyel sürüş tekniğine sahiptir (Nourbakhsh ve Siegwart, 2004). İki adet motor ile hareket edebilir ve sarhoş tekerleri sayesinde kendi etrafında 360° dönebilme kabiliyetine sahiptir.

İlk olarak algoritmanın parçacık sayısı 80 olarak seçilmiş ve buna göre her iki farklı sensör de kullanılarak ortam haritaları elde edilmiştir (Şekil 4.5).

⁵ http://wiki.ros.org/turtlebot_teleop

⁶ <https://www.turtlebot.com/turtlebot2/>

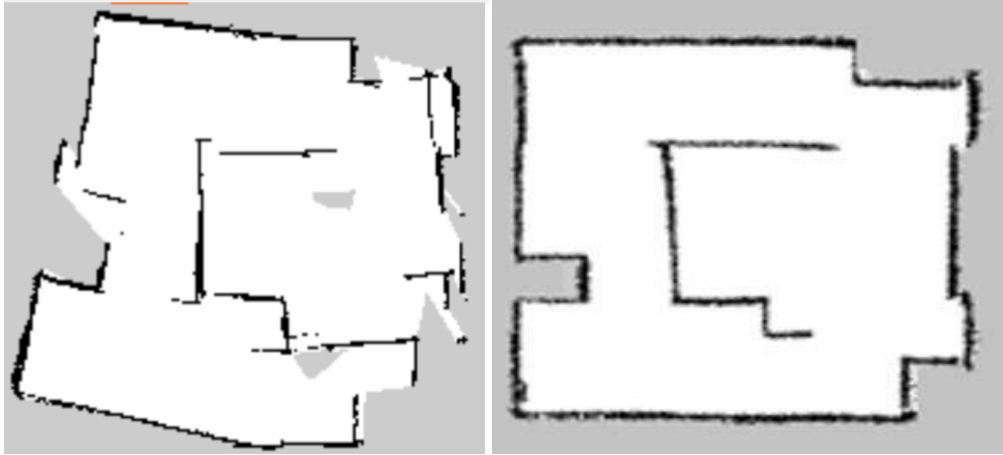


(a) Kinect sensörü sonuç haritası

(b) Lazer sensörü sonuç haritası

Şekil 4.5. Bir SLAM algoritması üzerinde farklı türdeki sensörlerin karşılaştırılması uygulamasına dair parçacık sayısının 80 seçildiği durumlar için elde edilen sonuçlar; (a) Kinect sensörü kullanılarak elde edilen nihai harita, (b) Lazer sensör kullanılarak elde edilen nihai harita

Her iki sensör kullanılarak elde edilen haritalar incelendiğinde sonuç haritalarının gerçek görünümünden oldukça uzak olduğu görülmüştür. Sonuç başarımını iyileştirmek için ortamda çözümleri temsil eden parçacık sayısı artırılmıştır (300). Bununla ilgili algoritma sonuçları Şekil 4.6’da paylaşılmıştır.



(a) Kinect sensörü sonuç haritası

(b) Lazer sensörü sonuç haritası

Şekil 4.6. Bir SLAM algoritması üzerinde farklı türdeki sensörlerin karşılaştırılması uygulamasına dair parçacık sayısının 300 seçildiği durumlar için elde edilen sonuçlar; (a) Kinect sensörü kullanılarak elde edilen nihai harita, (b) Lazer sensör kullanılarak elde edilen nihai harita

Parçacık sayısının artması doğal olarak ortamda daha fazla çözümü temsil eden varsayımların olmasını ve bu bağlamda haritanın iyileşmesini sağlamıştır. Rplidar sensörünün 360° tam tarama yapması nedeniyle beklendiği üzere sonuç haritası daha hızlı elde edilmiş ve gerçek değerine yakınsamıştır. Rplidar sensörü ile 92 saniyede ortam

haritası oluşturulmasına rağmen Kinect sensörü ile ancak 210 saniyede sonunda sonuçlar elde edilmiştir. Bu sonuçlar, Kinect sensörünün lazer sensör olarak kullanılması fikrinin gerçek zamanlı ortam denemesinde iyi haritalar üretmediğini ortaya koymuştur. Gerçekleştirilen çalışmada algoritma girdisi olarak sadece parçacık sayısının değişmesine dair parametre göz önüne alınmıştır. Parçacık sayısının yanı sıra *gmapping* algoritmasında farklı parametrelerde bulunmaktadır. Bu parametrelerin değişimini ve büyük ölçekli ortam denemelerini kapsayan detaylı bir çalışmada yapılabilir. Bu veriler ışığında, RGB-D sensörlerin 2B lazer sensör olarak kullanılması düşünülen gerçek zamanlı çalışmalarda parametre ayarlarının iyi yapılmasının gerekliliği ortaya çıkmıştır. Aksi durumda Şekil 4.5 ve 4.6'da görüleceği üzere küçük boyutlu ortamlar için bile harita gerçek değerlerinden uzakta olabilmektedir. Yapılan bu çalışmanın sonuçları Korkmaz ve ark. (2018)'de yayınlanmıştır.

4.4. En Uygun Yol Bulma Algoritması ile İlgili İnceleme

SLAM veya haritalama probleminde elde edilen haritaların nihai amaçlarından birisi robotlara görevler verilmesi ve ilgili görevlerin gerçekleştirilmesidir. Mobil robotların kendilerine atanan bu görevleri gerçekleştirmek amacıyla harita üzerinde farklı noktalara hareket etmesi gerekebilir. Ancak robotun kendisine atanan bir görevi gerçekleştirmek için görev noktasına ulaşabileceği farklı rotalar olabilir. Bu kapsamda, robotun hedef noktasına en optimal bir şekilde ulaşması istenir. Optimallik ölçütü problemin türüne veya kullanıcının ihtiyacına göre farklılık gösterebilir. Örneğin, zaman parametresinin önemli olduğu durumlarda en optimal çözüm en kısa zamanda nihai hedefe giden algoritma olabilirken peşi sıra gelen hedeflerin olduğu durumlar göz önüne alındığında en kısa zamanda rotayı bulan algoritma tercih edilebilir ve bu durumda en optimal çözüm en kısa zamanda rotayı belirleyen algoritma olabilir. Bu gibi farklı yaklaşımlara göre optimallik ölçütü tanımlanabilir ve en uygun yol bulma algoritmaları kullanılarak istenilen çözüm elde edilebilir.

A yıldız arama algoritması (A*): Bu algoritma, navigasyon probleminde bir noktadan diğer noktaya olan en kısa yolun bulunması ile ilgilidir. Algoritmada $f(n)$ fonksiyonunun küçültülmesi amaçlanmaktadır. $f(n)$ maliyet fonksiyonu Eşitlik (4.1)'deki gibi ifade edilir.

$$f(n) = h(n) + g(n) \quad (4.1)$$

burada n yol veya graf içerisindeki bir sonraki noktayı veya düğümü, $h(n)$ bitiş düğümüne karşılık gelen sezgisel maliyet ve $g(n)$ ise başlangıç durumuna göre olan maliyeti ifade etmektedir (Saian ve ark., 2016). Çalışma süresince robotun bir birimlik, dört ana (ileri, geri, sağ, sol) ve dört köşegen (sağ-ileri, sağ-geri, sol-ileri, sol-geri) doğrultularında hareket ettiği varsayılmıştır. Bununla birlikte harita çözünürlüğünün bu algoritmanın hesaplama zamanı üzerinde etkisi önemli bir parametredir. Yapılan irdelemelerin sonucu olarak harita çözünürlüğü X ve Y doğrultularında 100 olarak seçilmiştir.

Genetik Algoritma (GA): Optimizasyon gereken birçok alanda kullanılan bir yaklaşımdır. Özellikle doğrusal bir çözümü olmayan karmaşık sistemlerin en iyi parametrelerini elde etmede kullanılır. Navigasyon probleminde en kısa yolun bulunması için bu algoritmadan yararlanılabilir. Doğal seçim sürecini taklit eden ve en iyi bireylerin elde edilmesi ile ilgili bir çözüm olup sezgisel bir yaklaşımdır. Ebeveyn kromozomlarının çaprazlanması, çeşitlilik katılması ve yerel optimum noktalardan kurtulmak için mutasyon işlemi ile yeni bireylerin eklenmesi gibi farklı basamaklardan oluşmaktadır. Çözüm kümesindeki bireyler sürekli olarak amaç fonksiyonuna göre araştırılır ve en iyi çözümün elde edilmesi amaçlanır. İstenilen kriter veya iterasyon değerine göre arama sonuçlandırılır (Sonmez ve ark., 2015). Bu çalışmada kullanılan GA yapısı için popülasyon sayısı 50 olarak seçilmiştir. Bununla ilişkili farklı denemeler yapılmasına rağmen çözümün kaydadeğer bir şekilde değişmediği görülmüştür. Bir ceza puanı tanımlanarak bulunan yolun haritada engelin olduğu kısımlardan geçmemesi sağlanmıştır. Çalışmanın gerçekleştirilmesi için MATLAB programının hazır kütüphane fonksiyonu genetik algoritma optimizasyon aracından yararlanılmıştır.

Hızlı-Kesfeden Rastlansal Ağaçlar (RRT): Bir yol planlama algoritması olup özellikle holonomik olmayan (robotun tekerlerine dik yönde hareket edememesi) kısıtlara sahip hareket problemlerinin çözümleri ile ilgilenir. Genel olarak çözüm uzayının artırımsal bir şekilde bir ağaç oluşturulması metodolojisini kullanır. Ağacın *inşaası* ve *genişlemesi* için farklı fonksiyon kullanılarak çözüm uzayı oluşturulur. Bu yaklaşıma benzer şekilde geliştirilen bir diğer metotta ise iki yönlü olarak arama yapılır (**bRRT**). Bu algoritmada ağaç başlangıç ve hedef noktasından itibaren iki yönlü olarak oluşturulur. Ağaçlar

birleşim noktasına geldiğinde çözüm bulunmuş olur. Bu türde bir yaklaşımla algoritmanın zaman verimliliğinin artırılması sağlanmaktadır (Sun ve ark., 2018). RRT ve bRRT algoritmalarında önemli parametrelerden birisi adım sayısıdır. Bu değer artması ile hızlı ancak daha keskin dönüşlerin olduğu sonuçlar elde edilebilir. Aksine adım sayısının azaldığı durumlarda, daha iyi yol eğrilerinin olduğu ancak yavaş sonuçların üretildiği sonuçlar oluşturulabilir. Yapılan bir dizi denemeye göre bu çalışma için adım sayısı olarak 20 değeri seçilmiştir. Her iki algoritma için en fazla deneme sayısı 10000 olarak belirlenmiştir. Eğer deneme sayısı bu değeri geçerse algoritmalar yolun bulunamadığına dair bir sonuç üretmektedir.

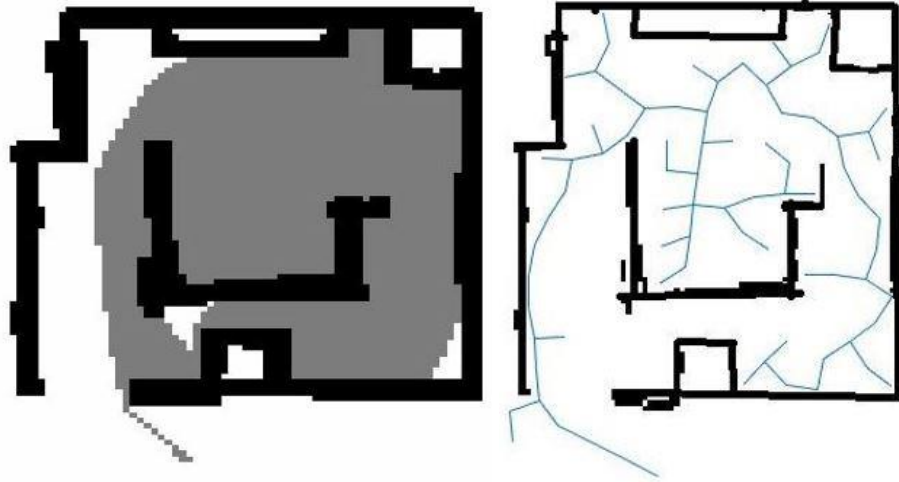
Olasılıklı yol haritası (PRM): Bir diğer tür en optimum yol bulma yöntemi olup olasılık tabanlıdır. Harita üzerinde rasgele oluşturulan düğümler ve bu düğümlerin bağlandığı kenarlar üzerinden yapılan bir hesaplama sayesinde en uygun yol bulunur. Bir diğer deyişle yol haritası için $G(E, V)$ şeklinde bir graf üretilir. İlk olarak engellerin olmadığı noktalardan örnekler alınarak düğümler oluşturulur. Oluşturulan bu düğümler birbiri ile ilintilenerek çözüm için en uygun yol hesaplanır. Düğümlerin birbirine göre uzaklığı için bir metrik gereklidir ve genelde Öklit uzaklığı ile hesaplamalar yapılabilir (Sudhakara ve ark., 2017). Çalışmada, yapılan bir dizi denemeler sonucunda başlangıç düğümü parametresi olarak 50 olarak seçilmiş ve uzaklık ölçütü olarak Öklit mesafesi kullanılmıştır.

Bu kapsamda gerçek zamanlı bir ortamın ROS *gmapping* paketi ve Turtlebot-II robotu kullanılarak haritası elde edilmiştir. Elde edilen haritaya göre beş farklı algoritmanın sonuçları karşılaştırılarak en optimal çözüm aranmıştır. Çalışmalar Intel i7 4720HQ 2.60 GHz işlemcili ve 16 GB RAM'i bulunan bir bilgisayar üzerinde yapılmış ve algoritma sonuçları elde edilmiştir.

Haritalar üzerinde rasgele seçilen başlangıç ve bitiş noktaları referans düğümler olarak kabul edilmiş ve algoritmalar çalıştırılmıştır.

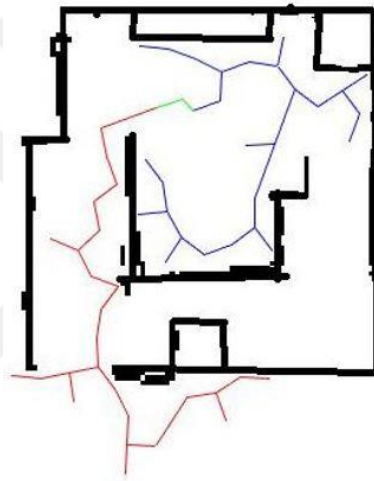
- Başlangıç noktası [X, Y] : [89, 237]
- Hedef (bitiş) noktası [X, Y] : [353, 152]

A* algoritmasının sezgisel yaklaşımı, ağaç algoritmalarının dallanmaları, PRM algoritmasının oluşturduğu graf ve GA'nın en iyi bireyi bulma grafiklerine dair yöntemlerin gelişim süreçleri Şekil 4.7'de gösterilmiştir.

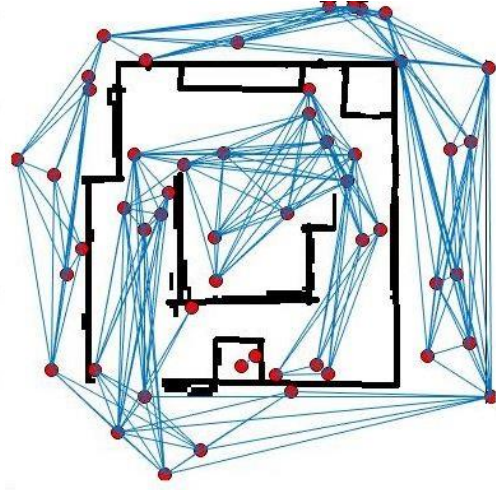


(a) A*

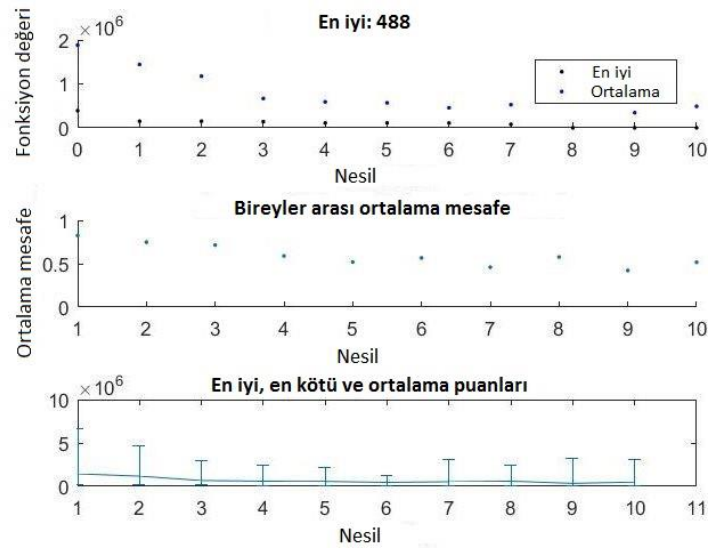
(b) RRT



(c) bRRT



(d) PRM



(e) GA

Şekil 4.7. En uygun yol bulma problemi çalışmasında kullanılan algoritmalara ait gelişim aşamaları; (a) A*, (b) RRT, (c) bRRT, (d) bRRT, (e) GA

Elde edilen çözümlerin sezgisel sonuçlar içermesinden dolayı her bir çalışma sonunda algoritmalar farklı sonuç üretebilmektedir. Bu yüzden sonuçlar istatistiksel bir irdelemeye tabi tutulmalıdır. Buna göre N boyutlu çözüm uzayının içerisinde 10 farklı sonuç (Çizelge 4.2) seçilmiştir. Bu 10 deneme için bütün algoritmalar aynı koşullarda çalıştırılmış ve herhangi bir algoritma parametresi değiştirilmemiştir. Bu denemelere ait sonuçların zaman ve mesafe açısından değerlendirilmesi yapılmıştır. Değerlendirme için tek yönlü varyans analizi (ANOVA) yöntemi ele alınmıştır (Chien ve ark., 2010). Bu testin amacı en iyi sonuçları veren algoritmaları değerlendirmek üzerinedir. İncelemenin yapılabilmesi için temel olarak farklı iki optimallik tanımı vardır: (i) sonucun bulunması için gereken zaman, (ii) başlangıç-bitiş noktaları arasındaki yol uzunluğu. Giriş kısmında da belirtildiği üzere birden fazla bakış açısı ile optimallik tanımı yapıldığında hangi yöntemin daha iyi sonuç vereceği veya hangi yöntemin kullanıcıya hitap edeceği kullanıcının ihtiyaçlarına göre değişebilir. Bazı durumlarda en kısa zamanda sonucu bulan bazense en kısa yolu bulan algoritmanın tercihi problemin türüne bağlıdır. Buna göre algoritmaların başlangıç-bitiş noktaları arasındaki yolu bulma zamanı ve bu yolun uzaklığına ilişkin kriterler göz önüne alınarak, ANOVA sonuçları elde edilmiştir (Çizelge 4.3'ten 4.7'ye kadar).

Çizelge 4.2. En uygun yol bulma problemi çalışmasında kullanılan algoritmaların on kez çalıştırılması sonucunda elde edilen algoritmalara ait farklı zaman ve mesafe değerleri

	Algoritma	Zaman (s)	Yol Uzunluğu (br)		Algoritma	Zaman (s)	Yol Uzunluğu (br)
	Arama - 1	GA	24.62		524	Arama - 2	GA
	A*	76.96	368		A*	76.96	368
	RRT	2.38	446		RRT	2.19	491
	bRRT	1.98	434		bRRT	1.46	500
	PRM	0.95	397		PRM	1.07	386
	Algoritma	Zaman (s)	Yol Uzunluğu (br)		Algoritma	Zaman (s)	Yol Uzunluğu (br)
	Arama - 3	GA	25.60		493	Arama - 4	GA
	A*	76.96	368		A*	76.96	368
	RRT	1.69	436		RRT	1.96	454
	bRRT	2.18	437		bRRT	1.64	435
	PRM	1.06	382		PRM	1.10	370
	Algoritma	Zaman (s)	Yol Uzunluğu (br)		Algoritma	Zaman (s)	Yol Uzunluğu (br)
	Arama - 5	GA	25.47		461	Arama - 6	GA
	A*	76.96	368		A*	76.96	368
	RRT	1.62	443		RRT	2.10	455
	bRRT	2.15	451		bRRT	1.66	464
	PRM	0.94	370		PRM	0.98	412
	Algoritma	Zaman (s)	Yol Uzunluğu (br)		Algoritma	Zaman (s)	Yol Uzunluğu (br)
	Arama - 7	GA	25.40		566	Arama - 8	GA
	A*	76.96	368		A*	76.96	368
	RRT	2.87	590		RRT	1.63	456
	bRRT	1.89	414		bRRT	1.52	537
	PRM	1.01	369		PRM	0.88	389
	Algoritma	Zaman (s)	Yol Uzunluğu (br)		Algoritma	Zaman (s)	Yol Uzunluğu (br)
	Arama - 9	GA	24.69		507	Arama - 10	GA
	A*	76.96	368		A*	76.96	368
	RRT	2.90	201		RRT	2.20	467
	bRRT	2.40	472		bRRT	1.60	431
	PRM	1.06	388		PRM	1.04	370

ANOVA incelemesine göre mesafe ve zaman değişkenleri baz alınıp inceleme yapıldığında sonuçların dört farklı guruba ayrılabilirdiği görülebilir (Çizelge 4.4 ve 4.6). Gruplar içerisinde A grubu en kötü değerleri veren algoritmaların oluşturduğu grup, D grubu ise en iyi değerleri veren algoritmaların bulunduğu gruptur. İlk olarak zaman değişkeni ele alınıp bu kritere göre en iyi yöntemin hangisi olduğu sonucu istatistiksel olarak araştırılmıştır. Bu değişken ve ANOVA analizine ait sonuçlar Çizelge 4.3 ve 4.4'teki gibidir. Çizelge 4.3'te analize ait farklı sonuçlar bulunmaktadır. DF serbestlik

derecesini, SS kareler toplamını, MS kareler toplamı ortalamasını, F frekansı, P ise istatistiksel anlamlılığı göstermektedir. Çizelge 4.4'teki sonuçlar ise algoritmaların ortalama, standart sapma ve gruplamaya dair bilgilerine işaret etmektedir. Buna göre başlangıç-bitiş noktaları arasında optimal yolu bulan algoritmalar zaman açısından bir değerlendirmeye tabi tutulursa, *D* grubu içerisinde yer alan bRRT ve PRM algoritmalarının en iyi, *A* grubu içerisinde yer alan A* algoritmasının en kötü sonucu verdiği söylenebilir. Eğer kullanıcı zaman kıstasını düşünürse bRRT veya PRM algoritmalarını kullanarak en iyi sonuçları elde edebilir.

Çizelge 4.3. En uygun yol bulma problemi çalışması için zaman değişkeninin baz alındığı duruma dair yapılan ANOVA analizi sonuçları

Kaynak	DF	SS	MS	F	P
Algoritma	4	42753,73	10688,43	9252,04	0,000
Hata	45	51,99	1,16		
Toplam	49	42805,72			

Çizelge 4.4. En uygun yol bulma problemi çalışması için zaman değişkeninin baz alındığı duruma dair Fisher metoduna göre gruplama ve grupların standart sapma değerleri (%95 güven aralığı)

Algoritma	N	Ortalama	StDev	Grup
A*	10	76,960	0,000	A
bRRT	10	1,848	0,320	C , D
GA	10	25,886	2,335	B
PRM	10	1,009	0,070	D
RRT	10	2,154	0,464	C

İkinci olarak başlangıç-bitiş noktaları arasındaki algoritmaların bulunduğu toplam mesafe değişkeni ele alınıp ANOVA analizi yapılmış ve bu kritere göre en iyi yöntem hangisidir sonuçları istatistiksel olarak araştırılmıştır. Bu değişken ve ANOVA analizine ait sonuçlar Çizelge 4.5 ve 4.6'daki gibidir. Çizelge 4.5 bir önceki çalışmaya benzer şekilde algoritmalar arası farklı parametreleri göstermektedir. Çizelge 4.6 ise gruplama sonuçlarını işaret etmektedir. Bu sonuçlara göre, başlangıç-bitiş noktaları arasında optimal yolu bulan algoritmalar mesafe açısından değerlendirilirse, en iyi yöntem *D* grubu içerisinde yer alan A* ve PRM algoritmaları, en kötü sonuç ise *A* grubunda yer alan GA yöntemi olarak yorumlanabilir. Eğer kullanıcı mesafe kıstasını göz önüne alırsa A* veya PRM algoritmalarını kullanabilir.

Çizelge 4.5. En uygun yol bulma problemi çalışması için mesafe değişkeninin baz alındığı duruma dair yapılan ANOVA analizi sonuçları

Kaynak	DF	SS	MS	F	P
Algoritma	4	252422	63106	10,80	0,000
Hata	45	262926	5843		
Toplam	49	515349			

Çizelge 4.6. En uygun yol bulma problemi çalışması için mesafe değişkeninin baz alındığı duruma dair Fisher metoduna göre gruplama ve grupların standart sapma değerleri (%95 güven aralığı)

Algoritma	N	Ortalama	StDev	Grup
A*	10	368,00	0,000	D
bRRT	10	457,50	37,26	B
GA	10	568,90	135,44	A
PRM	10	383,30	14,18	C, D
RRT	10	443,90	96,34	B, C

Optimallik kullanıcının gereksinimine göre belirleyeceği bir durum olup ihtiyaca göre en uygun algoritma seçilebilir. Yapılan bu çalışmaya göre kullanıcı, zaman ve mesafe değişkenlerinin her ikisini de göz önüne alırsa her iki analizde de en iyi sonuçları veren PRM algoritmasını tercih edebilir.

Her ne kadar bu tür analizlerin yapılması sonuç değerlendirmeleri için önemli olsa da pratik açıdan bu kontrolleri sürekli olarak tekrar etmenin zaman alacağı aşikardır. Bu düşünceyle tez kapsamında genellikle en kısa zamanda en kısa mesafeyi bulan algoritma hangisidir düşüncesinin pratik bir testi için bir verimlilik ölçütü tanımlanmıştır (Eşitlik (4.2) ve (4.3)).

$$\varepsilon_i = t_i \cdot x_i \quad (4.2)$$

$$\eta_i = \text{inv}(\varepsilon_i \div \max|\varepsilon_i|) \quad (4.3)$$

$i = 1, 2, 3, \dots, n$ indeks değeri ve n ise toplam algoritma sayısıdır. Örneğin bu çalışma için beş farklı algoritma kullanılmıştır ($n = 5$). 1. algoritma GA, 2. algoritma A*, 3. algoritma RRT, 4. algoritma bRRT ve 5. algoritma PRM yöntemleridir. t_i , i . yöntemle ait robotun gideceği yolun bulunması için geçen algoritma zamanını, x_i bu yöntemin bulunduğu başlangıç ve bitiş noktaları arasındaki mesafeyi göstermektedir. η_i ise i . algoritmanın verimlilik puanını gösterir ve algoritmalar arasında en optimalin hangisi olduğuna dair bir kıyas ölçütü gibi düşünülebilir. Bu ölçüt, algoritmaların zaman ve

mesafe deęişkenlerini göz önüne alarak birbirlerine göre verimlilięini gösteren bir puan üretmektedir. *inv* operatörü ise parantez içerisindeki ifadenin tersini alan bir işlemdir.

Verimlilik hesabının yapılması için bütün algoritmalara dair çözüm uzayından bir sonuç kümesi seçilir. Ortaya atılan verimlilik ölçütüne göre en kötü algoritma skoru 1 değerine eşit olacak şekilde diğer algoritma değerleri normalize edilir ve verimlilik puanları bulunur. Bu kapsamda Çizelge 4.7 kontrol edilirse PRM algoritmasının zaman ve mesafe değerleri baz alındığında en yüksek verimi verdiği görülmüştür. Bu sonuç ANOVA sonuçları ile tutarlı olup geliştirilen verimlilik kriteri pratik olarak ön denemelerin değerlendirilmesinde kullanılabilir.

Çizelge 4.7. En uygun yol bulma problemi çalışması için zaman ve mesafe deęişkenlerinin baz alındığı duruma dair algoritmaların verimlilik değerleri

Algoritma	Zaman (t)	Mesafe (x)	Verim (η)
A*	76.96 s	368 br	1
bRRT	1.98 s	434 br	32.95778
GA	24.62 s	524 br	2.195298
PRM	0.95 s	397 br	75.09288
RRT	2.38 s	446 br	26.68094

Sonuç olarak seçilen başlangıç/bitiş noktaları için en iyi yollara yol eğrileri Şekil 4.8'deki gibi elde edilir.



Şekil 4.8. En uygun yol bulma problemi çalışması için farklı algoritmalara dair başlangıç ve bitiş noktaları arası elde edilen yollara ait eğriler; (a) A*, (b) GA, (c) RRT, (d) bRRT, (e) PRM

SLAM algoritmaları sonucunda elde edilen haritalar robotlar tarafından farklı görevlerde kullanılır. Bu kapsamda, özellikle bir noktadan diğer bir noktaya gitme görevlerinde izlenecek yolun içeriği ve bunun bulunması bir önem teşkil etmektedir. Bu alt bölüm içerisinde açıklanan ve sonuçları irdelenen metotlar kullanılarak uygun bir yöntem seçilip navigasyon şeması SLAM kısmına entegre edilebilir. Farklı optimallik ölçütlerine göre en uygun yol planlaması elde edilebilir. Bu çalışmadan elde edilen veriler Korkmaz ve Durdu (2018) çalışmasında yayınlanmıştır.

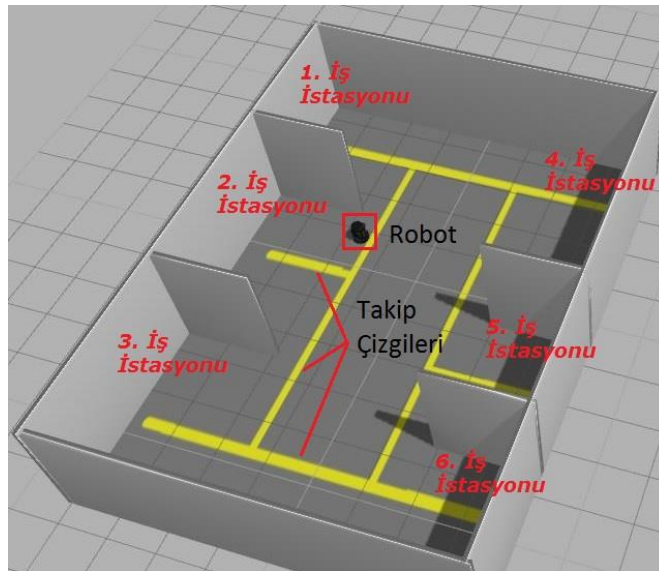
4.5. Aktif SLAM Algoritmaları ile İlgili Geliştirilen Yöntemler

SLAM algoritmasında bir robotun aynı anda ortamın haritasını çıkarması ve konum bilgisine erişmesi problemi incelenmektedir. Bölüm 3'te de bahsedildiği üzere

robotun nereye gideceği problemi ise navigasyon ve keşif algoritmaları başlığı altında incelenmektedir. Bu düşünceyle bazı durumlarda SLAM ve navigasyon problemi birlikte ele alınmalıdır. Bu algoritmalar genel olarak aktif SLAM başlığı altında değerlendirilmektedir. Bu tez çalışmasında mevcut aktif SLAM çalışmalarına ilave olarak iki farklı türde algoritma geliştirilmiştir. İlk olarak sola yönelimli bir çizgi izleme algoritma çerçevesi geliştirilmiş ve bunun nasıl uygulanacağı ve örnek bir çalışmaya ait sonuçlar irdelenmiştir. İkinci bir çalışma ise son yıllarda birçok alanda uygulamaları artan derin öğrenme tabanlı bir algoritma taslağı olup gerçek zamanlı bir uygulama üzerinde sonuçlar irdelenmiştir.

4.5.1. Sola yönelimli aktif SLAM algoritması

Aktif SLAM başlığı altında sola yönelimli bir çizgi izleme algoritması ile bir robotun herhangi bir kullanıcısı girdisine gerek kalmaksızın ortamda hareket ettirilmesi amaçlanmıştır. Denemelerin yapılabilmesi için Şekil 4.9'daki gibi fabrika çalışma koşullarına benzer küçük ölçekli bir ortam Gazebo⁷ ortamında tasarlanmıştır. Ortam içerisinde bölmelerle ayrılmış farklı iş istasyonları bulunmaktadır. Ayrıca bütün fabrika ortamlarında bulunabilen ve bölümleri birbirinden ayırma, kılavuzlama vb. işlerde kullanılan takip çizgileri ile çevre betimlenmiştir.



Şekil 4.9. Geliştirilen sola yönelimli aktif SLAM algoritmasında kullanılan çalışma ortamı, robot ve iş çalışma çizgileri

⁷ <http://gazebosim.org/>

Çalışmada robot, kendi kendine hareketi için ortam çizgilerini geliştirilen algoritma ile takip etmektedir. Bununla birlikte, ortamın haritası da çıkarılarak konumlama, haritalama ve navigasyon problemleri birlikte ele alınmıştır.

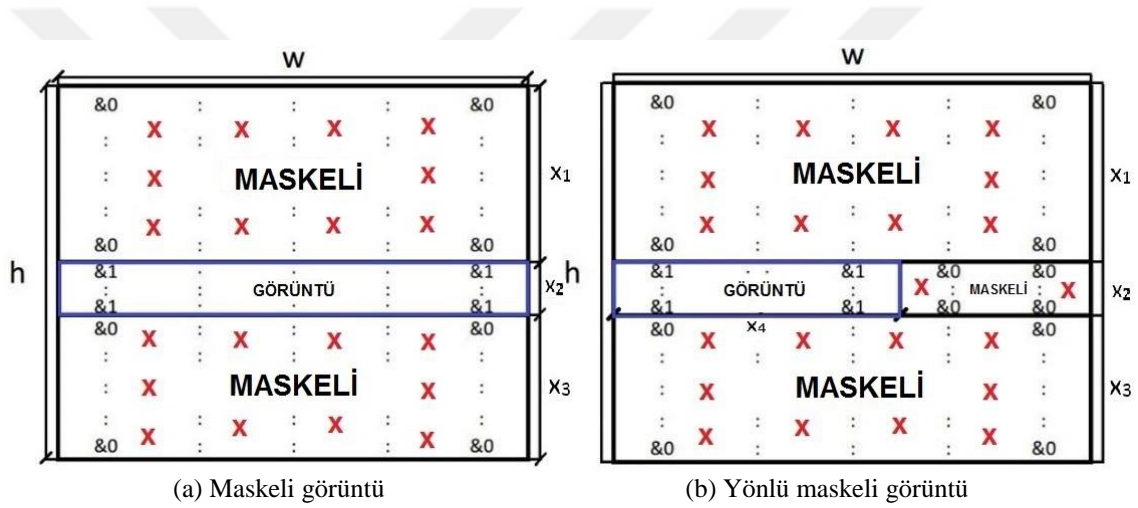
Robotun izleyeceği çizgilerin belirlenmesi için robot üzerinde bulunan kameradan yararlanılmıştır. Kameradan robotun gördüğü bölgeye dair görüntüler alınmaktadır. Robotun istenilen yoldan gidebilmesi için sarı ile betimlenen takip çizgilerinin belirlenmesi bu algorithmada önemli bir yer tutmaktadır. Bu çizgiler, normal şartlar altında kırmızı, yeşil, mavi ana (RGB) renkleri üzerinden ilgili yol çizgisi renginin karşılığı bulunarak bu renkle ilgili bir maskeleye yapılması ile kolayca belirlenebilmektedir. Ancak benzetim ortamında çalışabilecek bu düşünce gerçek zamanlı çalışmalarda aynı performansı veremeyebilir. Bunun en önemli sebeplerinden birisi, RGB renklerinin nesnenin rengi ve parlaklığını içeren bir temsil olmasıdır. Gerçek zamanlı çalışma koşullarında renklerin ortam ışığına göre yoğunluğunun değişmesi, gölgeli veya parlak bölgelerin oluşması rengin doğru belirlenememesine sebep olmaktadır. Her ne kadar benzetim çalışmasında bu türde gürültüler göz ardı edilebilsede ROS ve Gazebo'nun sunduğu çalışma koşulları sayesinde bu parametreleri çalışma koşullarına dahil etmek mümkün olmakta ve gerçek zamanlı çalışma koşullarına yakın bir deneme şansı elde edilmektedir. RGB renk uzayında karşılaşılan bu problem, rengin özü, doygunluğu ve parlaklığını içeren (HSV) bir temsil kullanılarak çözülebilir. Bu sayede, örneğin bu uygulama için sarı çizgiye yakın renklerin filtrelenmesi sağlanabilir. Çalışma boyunca kullanılan RGB ve HSV temsilleri için OpenCV⁸ kütüphanesi baz alınmıştır.

Bu kapsamda robotun ortamdaki çizgileri bularak takip etmesi amaçlanmıştır. Robotun takip çizgileri dışına çıktığı durumda oluşan hata, bir oransal denetleyici (P) yapısı üzerinden kontrol girdisi olarak robota uygulanmıştır. Bu sayede robot, takip çizgilerini merkezleyecek biçimde hareket edebilmektedir. Robotun çok yakınındaki mesafeler baz alınarak çizginin takip edilmesi istenirse (yaklaşık 1 metrenin altı) denetleyici yapısı gerekli düzeltmeyi uygulamak için ihtiyaç duyduğu zamanı bulamayabilir. Buna paralel şekilde, çok uzak bölgedeki çizgiler baz alınırsa (yaklaşık 2 metrenin üstü), kontrol girdileri beklenenden daha erken uygulanır ve robot takip çizgisinden uzaklaşabilir. Bu yüzden iyi bir denetleyici yapısı sunmak için robotun gördüğü alan içerisinde yaklaşık olarak bir metreye karşılık gelen bir bölgenin

⁸ <https://opencv.org/>

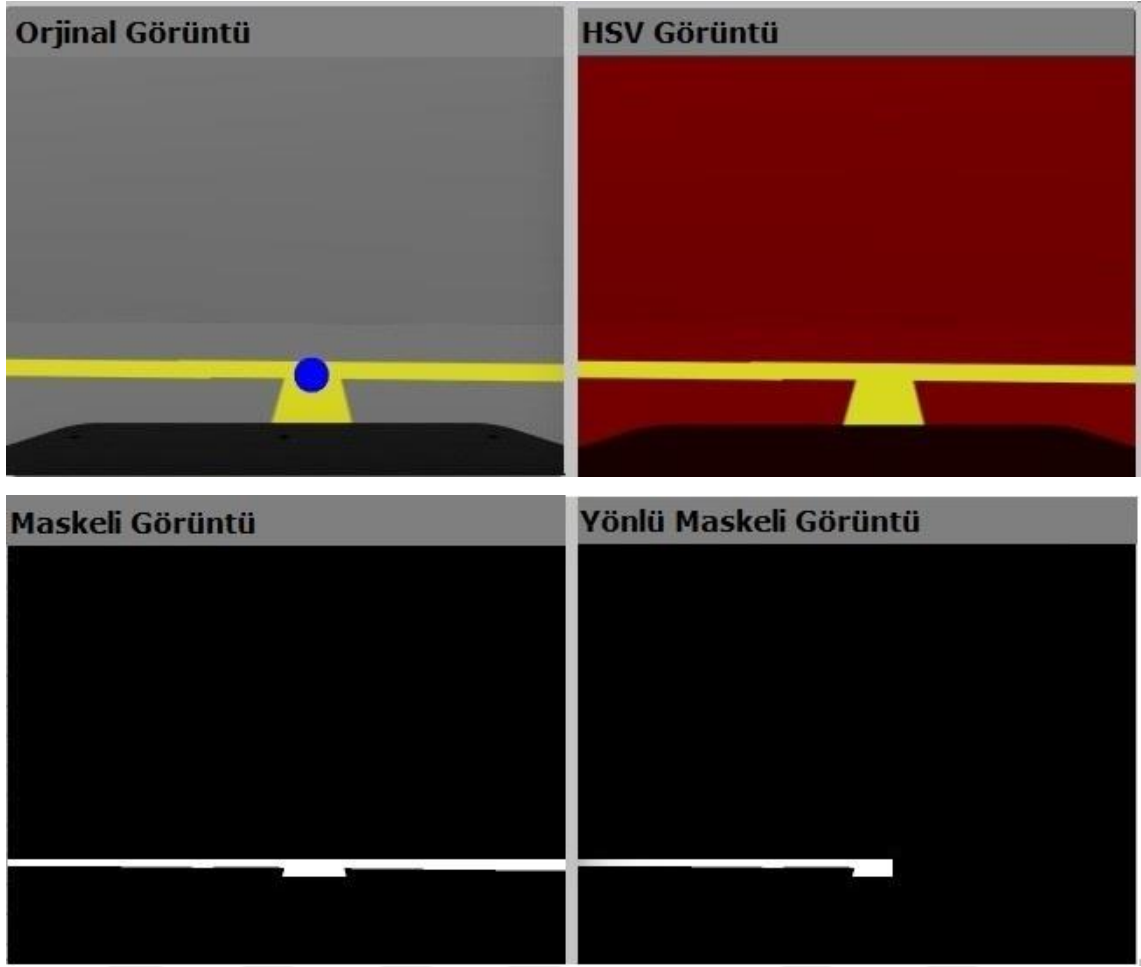
belirlenmesi amaçlanmıştır. Bu düşüncüyü gerçeklemek adına robot kamerasından alınan h yüksekliğinde ve w genişliğindeki bir görüntü x_1 , x_2 ve x_3 olacak şekilde üç farklı alt parçalara bölünmüştür. x_1 ve x_3 ile belirlenen kısımlar çok yakın veya uzak bölgeleri temsil ettiği için lojik sıfır değeri ile çarpılarak maskelenmiştir. Robotun takip etmesi gereken çizgiler yaklaşık olarak bir metrelik bir bölgeyi temsil eden x_2 bölgesindedir. Bu yapıyı temsil eden maske Şekil 4.10a'daki gibidir.

Bir diğer husus ise sola dönüş bölgelerinin belirlenmesidir. Sola dönüş olan kısımların belirlenmesi için birinci maske uygulandıktan sonra ortaya çıkan görüntü, x_2 , ilave bir x_4 parametresi ile iki kısma ayrılmıştır. Bu sayede sola dönüş olan bölgeler belirlenebilmektedir. Bu türde yönelimleri bulmak için Şekil 4.10b'de görülen maske yapısından yararlanılmıştır.



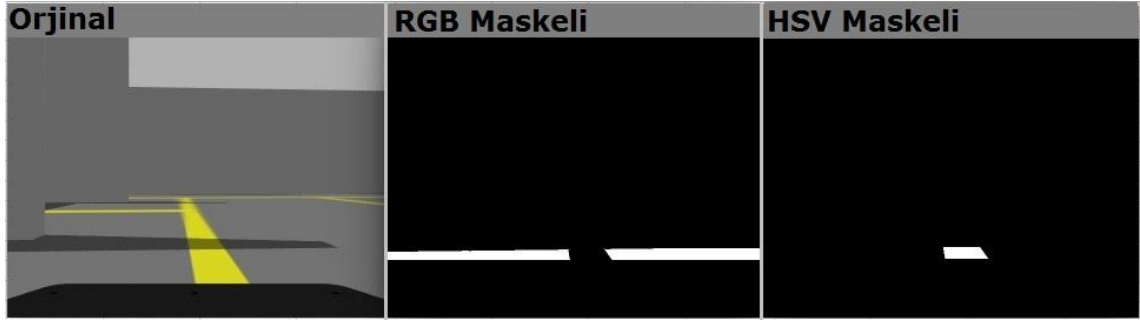
Şekil 4.10. Robot kamerasından alınan görüntülere uygulanan lojik 0 ve 1 maskeleme işlemlerine ait içerikler; (a) Robotun yaklaşık olarak bir metre önündeki mesafeyi belirlemek için kullanılan maske, (b) Robotun sola dönüş bölgelerini bulmak için kullanılan sol maskeli görüntü

Mesafelerin belirlenmesinde $x_1 = 3 * h/4$, $x_2 = 20 \sim 30 [px]$, $x_3 = h - (x_1 + x_2)$ ve $x_4 = \frac{w}{2} + \alpha$ denklemleri kullanılmıştır. Örnek bir 480×360 çözünürlüğündeki görüntü için $x_1 = 270$, $x_2 = 25$, $x_3 = 65$, $x_4 = 300$ olarak seçilebilir. Robot kamerasından alınan RGB uzayındaki görüntü, HSV dönüşümü yapılan görüntü, robotun çizgileri takip etmek için kullanacağı maskeli görüntü ve sola yönelimin belirlenmesi için yararlanılan yönlü maskeli görüntüler Şekil 4.11'de verilmiştir.



Şekil 4.11. Sola yönelimli aktif SLAM algoritma uygulamasına dair robot kamerasından alınan ve algoritma içeriğinde yapılan işlemlere ait görüntüler; sırasıyla RGB renk uzay görüntüsü, HSV dönüşümü yapılmış görüntü, maskeli ve yönlü maskeli görüntüler

Yukarıda da bahsedildiği üzere gölge veya daha fazla parlak olan yerleri gerçek zamanlı bir ortama benzer şekilde ROS/Gazebo altyapısı ile tanımlamak mümkün olmaktadır. Şekil 4.12 incelendiğinde kameradan alınan görüntü (Şekil 4.12a) üzerinde bulunan takip çizgisi eğer RGB ana renkleri üzerindeki değerine göre filtrelenirse yanlışlıklar olabilmektedir. Örneğin Şekil 4.12b’de de görüleceği üzere takip çizgisi üzerine gölge düşen bir bölge RGB renkleri üzerinden maskelenirse asıl izlenmesi gereken yol bulunamamaktadır. Bunun yerine gölgeli veya farklı kısımların, izlenilmesi gereken yol gibi ortaya çıktığı görülmektedir. Buna karşın aynı bölgeye karşılık gelen HSV uzayına dönüştürülmüş görüntü üzerinden maskeleme yapılırsa Şekil 4.12c’deki gibi esas izlenecek yolun görülmesi sağlanmaktadır.



(a) Kameradan alınan görüntü

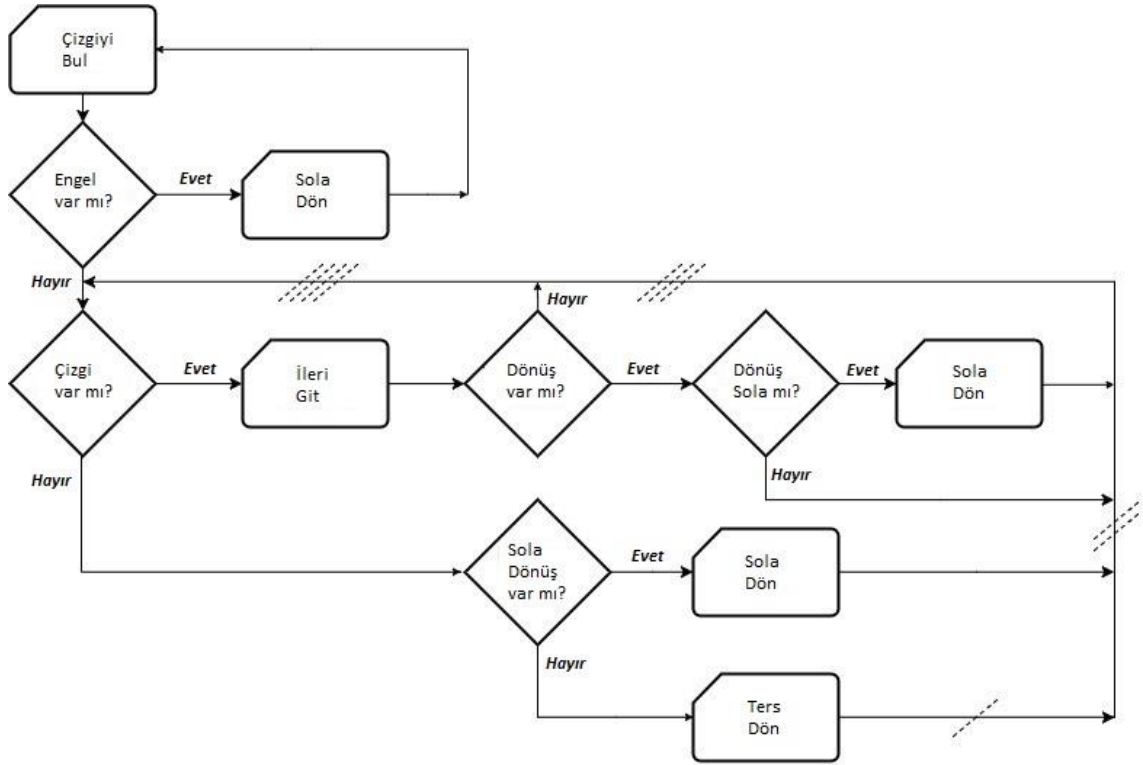
(b) RGB renk uzayına göre maskelenmiş görüntü

(c) HSV renk uzayına göre maskelenmiş görüntü

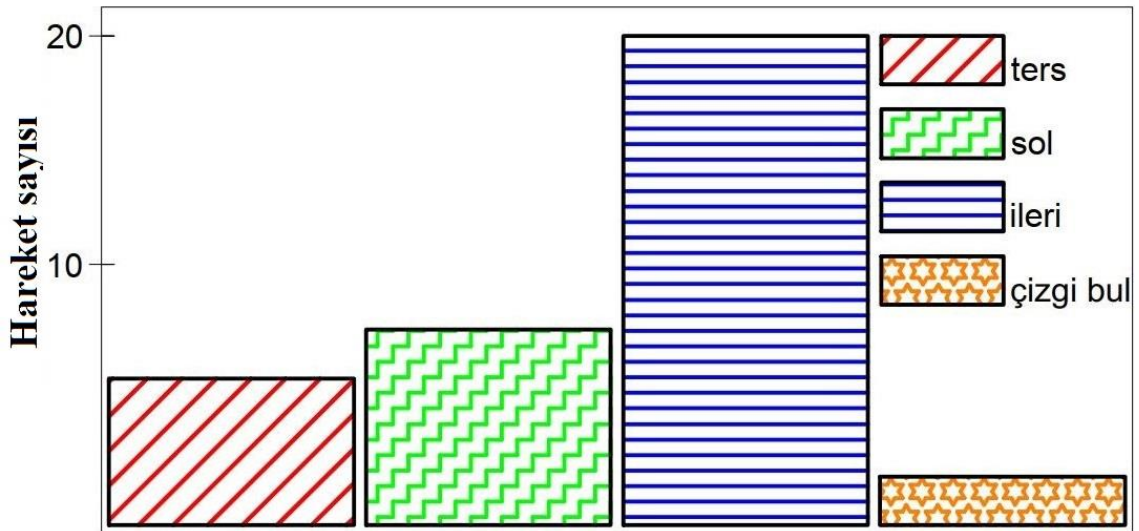
Şekil 4.12. Sola yönelimli aktif SLAM algoritma uygulamasına dair ortamdan alınan örnek bir görüntü; (a) Kameradan alınan RGB renk uzayındaki bir görüntü ve gölgeli ortamların görünüşü, (b) RGB renk uzayına göre filtreleme yapıldığında ortaya çıkan takip çizgilerine dair görüntü, (c) HSV renk uzayı temel alınarak filtreleme yapıldığında robotun takip edeceği bölgeye dair görüntü

Bu bilgiler ışığında Şekil 4.13'te blok diyagramları görülen sola yönelimli bir navigasyon algoritması ile robotun ortam içerisinde kendi kendine hareketi amaçlanmıştır.

Algoritma ilk olarak çizgiyi bulmayı amaçlamaktadır. Çizgi bulunduktan sonra çeşitli basamaklar üzerinden çizginin takibi yapılmaktadır. Burada saf bir çizgi izleme algoritması dönüş noktaları, çizginin bittiği kısımlar vb. durumlarda kararsız sonuçlar üretebileceği için sola yönelimli bir algoritma taslağı üzerinden takip yapılmıştır. Örneğin her iki yönde de çizgi bulunduğu durumlarda (Şekil 4.11) robot saf bir çizgi izleme kuralı ile hareket ederse hangi yöne döneceğine dair kararsız kalacak ve alan içerisinde ulaşılmayan bölgeler olabilecektir. Bu yüzden tek yöne hareket eden bir algoritma doğru sonuçlara ulaşabilecektir. Bu kapsamda dört farklı aksiyon ile robotun tüm hareketleri betimlenebilmektedir: ileri gitme, sola dönüş, ters dönme ve çizgiyi bulma. Buna göre robotun yaptığı aksiyonların sıklığı Şekil 4.14'te belirtilmiştir.



Şekil 4.13. Sola yönelimli aktif SLAM algoritma uygulamasına dair geliştirilen ve navigasyon kısmını temsil eden algoritma parçası



Şekil 4.14. Sola yönelimli aktif SLAM algoritma uygulamasına çalışması sırasında robotun yaptığı hareketlerin sıklık çizelgesi

İleri aksiyonları için robot 0.2 m/s hızla ve P denetleyicisi ile; sola dönüşler için 0 m/s ileri ve 90° sola; ters dönüşler içinse 0 m/s ileri 180° geriye dönme hareketleri yapılmaktadır. Tüm çalışma düzeni için farklı iki fonksiyon üzerinden sistem yapısı

oluşturulabilmektedir. Bu kapsamda robotun ortamda hareketi için çalışmada geliştirilen navigasyon algoritması ve ortamın haritasının çıkarılması için ROS *gmapping* paketi kullanılmıştır. Sistem ile ilgili bütün algoritma şeması Algoritma 4.1 içerisinde açıklanmıştır.

Robotun takip ve dönüş hareketi için resim momentleri kullanılmaktadır. Genel bir görüntü momenti bilgisayar görmesi başlığı altında Eşitlik (4.4)'teki gibi tanımlanabilir (Wang ve Cho, 2008).

$$m_{ij} = \iint_{R(t)} x^i y^j I(x, y) dx dy \quad (4.4)$$

Çalışma için kullanılan ifade ise Eşitlik (4.5)'teki gibidir.

$$\mu_{ij} = \iint_{R(t)} (x - c_x)^i (y - c_y)^j I(x, y) dx dy \quad (4.5)$$

Burada $c_x = m_{10}/m_{00}$ ve $c_y = m_{01}/m_{00}$ iki boyutlu bir görüntünün merkezleridir. Görüntü merkezleri ve çizgi merkezi arasındaki fark hata bilgisi olarak kullanılır. $e = c_x - c_i$ denkleminde göre robotun hareketi esnasında çizgiden ne kadar saptığına dair bir kontrol bilgisi ile P denetleyicisi tarafından çizgide kalması sağlanmaktadır. Daha iyi bir denetleyici yapısı sağlamak adına oransal, türevsel ve integral (PID) denetleyicisi de denenmiştir. Ancak bu türde bir denetleyici kullanıldığında çizgi takibi için başarımların yüksek olmadığı aksine sistemin karmaşıklığının arttığı görülmüştür. Sistem karmaşıklığını artırmamak adına basit bir P denetleyicisinin yeterli olduğu gözlemlenmiş ve bu yapı hata gideriminde/çizgi takibinde kullanılmıştır.

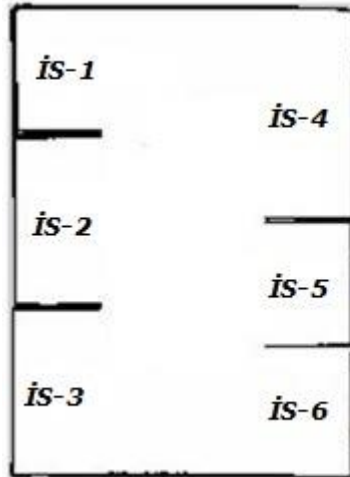
Bu bilgiler ışığında, aynı ortam için robotun kullanıcı girdisi ve geliştirilen algoritma ile yönlendirilmesine dair farklı iki durum incelenmiş ve sonuç haritaları karşılaştırılmıştır. Şekil 4.15a'da hiç bir kullanıcı girdisi olmadan robotun önerilen metotla yönlendirilmesi sonucu elde edilen harita bulunmaktadır. Benzer şekilde Şekil 4.15b'de ise kullanıcı girdisi ile yönlendirilen robot ile elde edilmiş harita bulunmaktadır. Bu sonuçlara göre her iki haritada birbirine oldukça yakındır. Ancak birisinde robot kullanıcı girdisi ile yönlendirilmiş diğerinde ise önerilen algoritma ile yönlendirilmiştir.

Algoritma 4.1. Tüm sistemin algoritma şeması

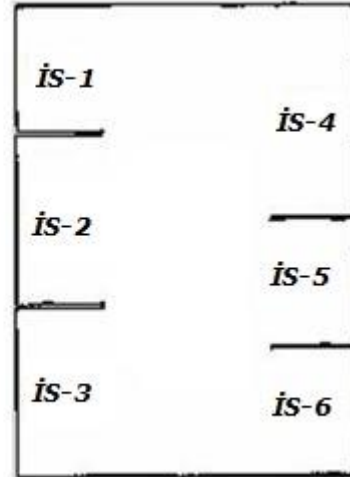
```

fonksiyon navigasyon() // Şekil 4.13'e bkz
fonksiyon bitir
fonksiyon GLOBAL Harita ← SLAM()
  while derinlik görüntüsü varken yap
    derinlik görüntüsünü al
    Lazer verisine dönüştür
    Sıralı tarama ile eşleştir
     $\{x_t^{(i)}\} \leftarrow \{x_{t-1}^{(i)}\}$  ve  $\pi(x_t|z_{1:t}u_{1:t})$  kullanarak yeni
      parçacık üret
     $w^{(i)} = \frac{p(x_t^{(i)}|z_{1:t}u_{1:t})}{\pi(x_t^{(i)}|z_{1:t}u_{1:t})}$  denklemi ile parçacıkları
      ağırlıklandır
    ağırlığı yüksek olanlar ile düşük olanları yer
    değiştir // (yeniden örnekleme)
     $x_t^{(i)}$  ile ilgili her bir parçacık için harita
      tahminini yap  $p(m_t^{(i)}|x_{1:t}z_{1:t})$ 
  While bitir
fonksiyon bitir
while sistemAktifse yap
  SLAM() // 1. kısım: çevre haritası oluşturma
  Navigasyon() // 2. kısım: otonom navigasyon
While bitir

```



(a) Önerilen metot



(b) Kullanıcı girdisi

Şekil 4.15. Sola yönelimli aktif SLAM algoritma uygulamasına dair elde edilen sonuçlar (a) Geliştirilen metot kullanılarak oluşturulan sonuç haritası; (b) Kullanıcı girdisi ile robotun güdümlenmesi durumu için elde edilen sonuç haritası (İS: İş istasyon bölgesi)

Literatürde birçok metot ortam ile ilgili meşgul olmayan hücrelerin keşfedilmesi ve buna göre robotun ilgili bölgeye yönlendirilmesi düşüncesinde şekillenmektedir. Bu

tez kapsamında incelenen bu çalışmada robotun yönlendirilmesine dair farklı bir bakış açısı sunulmuştur. Görüntü işleme metotları ve alt işlemler ile robotun yönlendirilmesi önerilmiştir. Elde edilen sonuçlar çerçevesinde önerilen metodun literatürde yer alan metotlara göre bir alternatif olabileceği ortaya çıkmaktadır. Çalışmanın sonuçları Durdu ve Korkmaz (2019)'da yayınlanmıştır.

4.5.2. CNN tabanlı aktif SLAM algoritması

Makine öğrenmesi, bilgisayarlı sistemlere eldeki veri kümesinden *öğrenme* yetisini kazandıran yapay zekanın bir alt bilim dalıdır. Klasik programlama sistemlerinden farklı olarak açık algoritmik adımlar yerine öğrenme ve buna bağlı bir yapı oluşturma düşüncesi bulunmaktadır. Çok genel manada üç ana sınıfta incelenir; gözetimli, gözetimsiz ve takviyeli öğrenme.

Gözetimli öğrenmede, eldeki verilerin ait olduğu sınıf bellidir ve amaç bu verilerden bir öğrenme modeli oluşturmak ve buna göre yeni gelen verinin hangi sınıfa ait olacağına dair karar vermektir. Naive Bayes, k-en yakın komşu (k-NN), karar ağaçları, destek vektör makinesi (SVM) gibi birçok algoritma ile bu işlem yapılabilmektedir. Ayrıca çıktıların kesikli bir durumu ifade eden sınıflar yerine süreklilik gerektiren olayları incelemesi için de regresyon yöntemleri kullanılır. En bilinen yöntemler doğrusal regresyon, polinomal regresyon, karar ağaçları gibi tekniklerdir.

Gözetimsiz öğrenme ise eldeki veri sınıflarının bir etiketinin olmadığı durumları kapsar. Gözetimsiz öğrenme, mevcut verilere dair kaç farklı sınıfın gerekli olduğunun bilinmediği durumlar için önemli ve kullanışlı olmaktadır. Kendi içerisinde farklı sınıfları olmakla birlikte en temel manada kümeleme ve birliktelik olarak ayrıştırılabilir. Kümeleme için k-ortalamları en bilinen algoritmalarındandır. Buna göre eldeki veri için optimal bir küme sayısı belirlenebilir. Birliktelik kavramı ise değişkenler arası farklılıkları inceler. Kümelemeye göre en büyük farkı veri noktaları yerine bunların nitelikleri arasındaki ilişkinin incelenmesidir.

Bu metotlara ilave olarak takviyeli öğrenme makine öğrenme alanının ilgilendiği farklı bir yöntemdir. Takviyeli öğrenmede temel düşünce yapılan davranışların ödüllendirilmesi veya cezalandırılması ve buna bağlı olarak bir öğrenme yapısının oluşturulmasıdır. Bu yönüyle canlıların öğrenme ve davranış psikolojisine benzetilebilir. Yapı, ceza aldığı durumları tekrar etmez bunun aksine ödül aldığı davranışı devam ettirmeye çalışır ve buna bağlı olarak sürekli öğrenme yetisini devam ettirebilir.

Klasik makine öğrenmesi bahsedilen bu başlıklar altında uzunca süredir incelenmiş ve günümüzde de araştırmalara devam edilmektedir. Ancak klasik makine öğrenmesi yöntemleri birçok manada yerel çözümler sunmakta biraz daha genelleştirmek istendiğinde sonuçlar doğru değerlerden kolayca sapmaktadır. Günümüz şartlarında veri miktarlarının büyümesi ile yeni çözümlere ihtiyaç duyulmuştur. Bu kapsamda derin öğrenme yapıları son zamanlarda aktif bir şekilde araştırmacılar tarafından incelenmektedir. Derin öğrenme makine öğrenmesinin bir alt dalı olup karar verme işlevleri için insan beynini taklit eden çok katmanlı bir modeldir. 2010 yılından bu yana derin öğrenme ile ilgili birçok araştırma yoğun bir şekilde devam ettirilmektedir. Bunda en önemli paylardan birisi GPU ile yapılan hesaplamaların daha hızlı olmasıdır. Derin öğrenme için genel olarak Tensorflow, Theano, Caffe, Pytorch gibi platformlar bulunmaktadır.

Evrışimsel Sinir Ağları (CNN veya ConvNet) ise derin öğrenmenin bir türüdür. Her ne kadar ilaç dizaynı, cümle modelleme gibi örneklerde de CNN'in kullanıldığı görülsede genellikle görüntü analizleri üzerinde kullanılmakta ve bilinen birçok yöntemle göre avantajlı sonuçlar ortaya koymaktadır. CNN yapılarında resimler üzerinden otomatik veya belirli desenler seçilerek özellikler oluşturulur. Bunun için elde edilen filtre/desen resme uygulanır ve çapraz ilintiye göre bu desenle resmin ilgili kısmının ne kadar ilintili olduğu bulunur. Konvolüsyon işlemi yerine korelasyonun kullanılması konvolüsyon işlevinde kaynaklanan evirme, çekirdek ağırlıkları gibi fazladan gelebilecek karmaşıklık azaltmaktadır (Cheong ve Park, 2017). Elde edilen sonuçlar genellikle ReLu katmanına tabi tutulur ve buna göre negatif değerler giderilir (Eşitlik (4.6)).

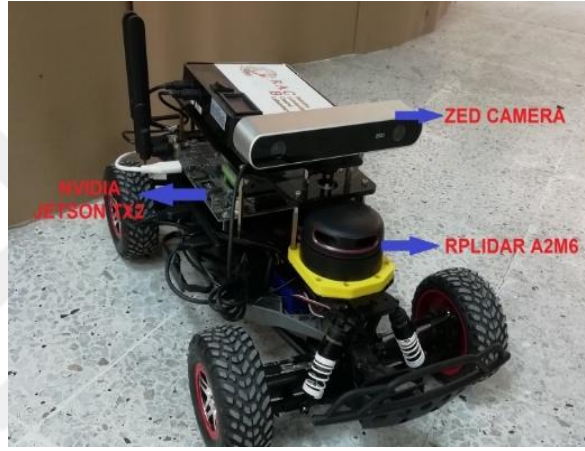
$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4.6)$$

Havuzlama aşamasında ise elde edilen görüntülerin boyutunun düşürülmesi amaçlanır. Bu sayede bir sonraki aşamanın giriş boyutu azaltılmış olur. Genellikle elde edilen filtrenin maksimum değeri alınarak sadeleştirme işlemi yapılır. Burada oluşan veri kaybı sonucunda ağır veriyi ezberlemesinin önüne geçilebilir. Konvolüsyon, ReLu ve havuzlama katmanlarının nasıl sıralanacağı, kaç adet katmanın olması gerektiği, kaydırma değerinin kaç seçileceği gibi birçok farklı parametre değiştirilerek CNN mimarilerinin başarıları değiştirilebilir. Elde edilen nihai konvülüsyon katmanları tam bağlantılı katmana uygulanarak ağ yapısı oluşturulur. Fazla verinin olduğu ağlarda bazen

DropOut katmanı ile ağın ezberlemesi ve verinin azaltılması için ağıdaki bazı düğümler kaldırılır (Ran ve ark., 2017).

Bu bilgiler ışığında bu tez çalışmasında son yıllarda popülaritesi artan ve birçok sisteme eklenmesi ile iyi sonuçlar veren CNN yapısının SLAM algoritmaları ile entegrasyonu üzerinde durulmuştur. SLAM problemi içerisinde aktif SLAM şemalarında CNN yapısının nasıl bir avantaj sağlayacağına dair çalışma yapılmıştır.

Şekil 4.16’da görülen bir mobil robot ile gerçek zamanlı bir çalışma yapılmıştır. Robot üzerinde NVIDIA Jetson TX2 kartı ile, zed kamera ve lazer sensör bulunmaktadır. Robotun hareket modeli Bölüm 3.2.1’deki modele benzerdir.



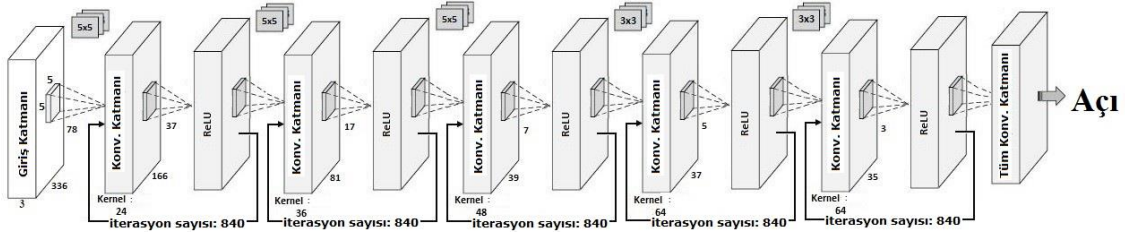
Şekil 4.16. CNN tabanlı aktif SLAM çalışmasında kullanılan robot

Aktif SLAM düşüncesi çerçevesinde, robotun kendi kendine hareketinin sağlanması ve hareket süresince ortamın haritasını çıkarması bu çalışmanın esas amacıdır. Bunu gerçekleştirebilmek için ortamdaki görüntüler ve robotun bu görüntülere karşılık teker manevra açıları kayıt edilerek bir veri tabanı oluşturulmuştur. Bu amaçla Şekil 4.17’de görülen eğitim ortamında robot bir kullanıcı tarafından hareket ettirilmiştir.



Şekil 4.17. CNN tabanlı aktif SLAM uygulamasında ağı eğitilmesi için yararlanılan ortam

Robotun hareketi esnasında zed kameradan alınan görüntüler girdi ve kullanıcı tarafından sağlanan robotun teker manevraları çıktı olacak şekilde CNN tabanlı bir ağ eğitilmiştir (Şekil 4.18).



Şekil 4.18. CNN tabanlı aktif SLAM çalışmasında kullanılan CNN yapısı

Bu şekilde bir çalışmanın amacı robotun kendi kendine ne zaman ileriye veya dönüşe karar vermesi gerektiği ile ilgili bir düşünceye kavuşmasını sağlamaktır. Aynı ortamda denemelerin yapılması gerek ağı test edilmesi gerekse ezberlenen bölgelerin olması ihtimalinden dolayı uygun olmayacağı için geliştirilen ağ farklı bir ortamda denemelere tabi tutulmuştur (Şekil 4.19). Test ortamı oluşturulurken eğitim ortamına göre farklı hareket bölgelerinin, farklı dönüşlerin olması gibi koşullar göz önünde tutulmuştur.

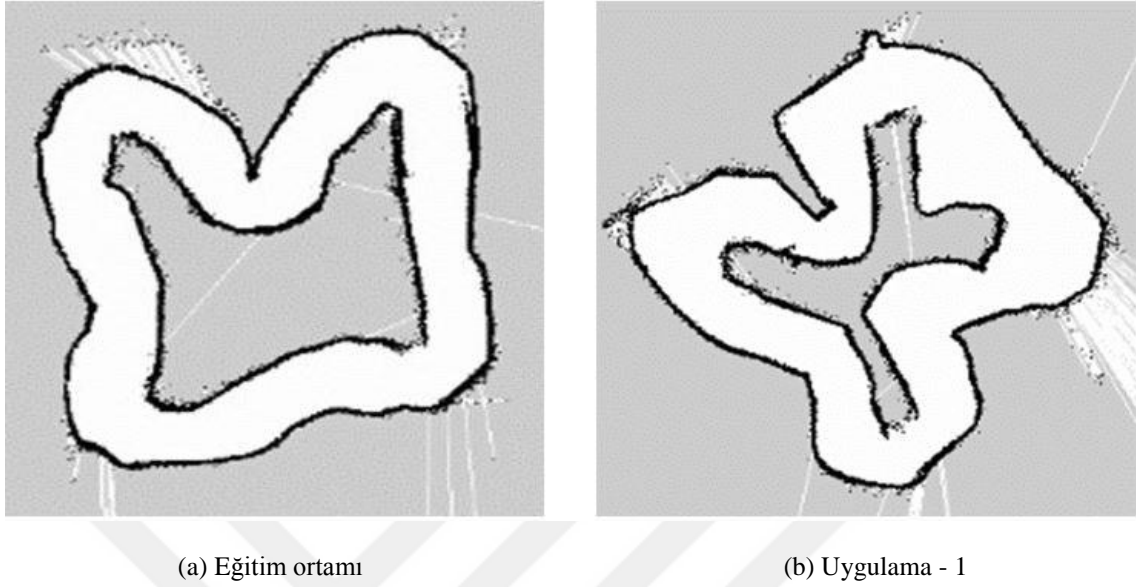


Şekil 4.19. CNN tabanlı aktif SLAM uygulamasına dair denemelerin gerçekleştirildiği birinci test ortamı (uygulama – 1: çoklu dönüş veya ileri hareket durumlarını içermeyen bir ortam)

Buna göre eğitim verileri ile oluşturulan ağ sayesinde robotun ortamda otonom bir şekilde hareketi sağlanmıştır. Bu hareket esnasında SLAM algoritması da çalıştırılarak aktif SLAM algoritmalarının gereği olan hem otonom hareket hem de ortam haritalaması yapılabilmektedir.

Eğitim ortamı ve test ortamı ile ilgili haritalar Şekil 4.20'deki gibidir. Görüldüğü üzere her iki ortamda birbirinden göreceli olarak farklı bölümler bulunmaktadır. Test

ortamı eğitim ortamına göre farklı bölgeler içermesine rağmen eğitilen ağ sayesinde robot kendi kendine hareket edebilmiştir.



Şekil 4.20. CNN tabanlı aktif SLAM uygulamasına dair haritalar; (a) CNN tabanlı aktif SLAM çalışması uygulamalarının tümünde kullanılan ağın eğitim işleminin yapıldığı ortamın haritası, (b) Eğitilen ağın doğrulanması için kullanılan birinci test ortamının haritası

Test boyunca kullanılan CNN yapısının oluşturulması için kullanılan parametreler Çizelge 4.8'deki gibidir.

Çizelge 4.8. CNN tabanlı aktif SLAM algoritması navigasyon kısmı için oluşturulan CNN ağının eğitilmesinde kullanılan parametreler

Parametreler	Sayı/Yöntem
Eğitim resim sayısı	5376
Test resim sayısı	1344
Batch Size	8
Aktivasyon fonksiyonu	RELU
Kayıp fonksiyonu	Ortalama Karekök Hatası
Optimizatör	Adam
Epok sayısı	40

Şekil 4.19'daki test ortamı tek yönlü bir veri içermekte olup robotun dönüş hareketlerini yapacağı zamanlar için farklı bir karar mekanizmasına ihtiyaç duyulmamaktadır. Ancak bazı durumlarda robotun hangi yöne gideceği gibi konularda da karar vermesi gerekebilir. Bu türde problemlerin denenmesine dair Şekil 4.21a'daki gibi ikinci bir deneme ortamı oluşturulmuştur.



(a) Test ortamı - 2

(b) 2. Ortama ait sonuç haritası

Şekil 4.21. CNN tabanlı aktif SLAM uygulamasına dair denemelerin gerçekleştirildiği ikinci uygulama; (a) ağın denemesi için oluşturulan ikinci test ortamı, (b) Eğitilen ağ kullanılarak elde edilen ortam haritası (uygulama – 2: dönüş ve ileri hareket durumlarının her ikisini birden aynı anda içeren bir ortam)

Elde edilen sonuç haritası incelendiğinde (Şekil 4.21b) işaretçi noktaların bulunduğu kısımlarda robot mevcut güzergah eğrisine göre *sola dönüş* veya *ileri hareket* aksiyonlarından birini seçmelidir. Bu seçim işlevi için ilgili işaretçi noktalara karekodlar yerleştirilmiştir. Robot gördüğü karekod değerine göre yaptığı aksiyonu belleğine kayıt etmekte ve bir sonraki adımda aynı noktadan tekrar geçerken farklı bir aksiyonu seçmektedir. CNN tabanlı metotlar günümüzde birçok araştırmaya farklı bir bakış açısı getirmektedir. Bu tez çalışmasında, bu başlıkta CNN yöntemlerinin SLAM çalışmalarında nasıl bir etki vereceği üzerine bir araştırma yapılmıştır. Gerçek zamanlı uygulamalar üzerinde sonuçlar irdelenmiş ve aktif SLAM yapısında karşılaşılan navigasyon şeması kısmının CNN tabanlı yöntemler ile de oluşturulabileceği görülmüştür.

4.6. Görsel SLAM Teknikleri ile İlgili Geliştirilen Yöntem

SLAM yöntemleri son yıllarda klasik lazer sensör ve Bayes tabanlı filtre kombinasyonlarının kullanımından kamera ve geliştirilen yeni türde algoritmalarından faydalanmasına doğru bir eğilim göstermektedir. 2015 yılında ortaya atılan ve yayımlandığı dergide en iyi makale ödülünü alan ORB-SLAM yayını üzerine çalışmalar tez kapsamında yapılmıştır (Mur-Artal ve ark., 2015). Bu metotta karşılaşılan özellik tabanlı haritaların yoğunluk tabanlı bir şekilde görülmesinin sağlanması amaçlanmıştır ve buna bağlı gerçek zamanlı bir çalışma ile sonuçlar elde edilmiştir.

ORB-SLAM, tek gözlü kameraların kullanıldığı özellik tabanlı haritalar üreten ve temelinde ORB özelliklerin kullanan bir vSLAM algoritmasıdır. ORB-SLAM2 diye adlandırılan bir üst versiyonunda stereo ve RGB-D kameraların kullanımı da eklenmiştir (Mur-Artal ve Tardos, 2017a). Bölüm 3'te özellik tabanlı harita üreten yöntemlerden bahsedilmiştir. ORB-SLAM bu yöntemler içerisinde son yıllarda ortaya konulan en iyi metotlardan birisi olarak görülmektedir.

ORB-SLAM yönteminde çözüm için üç ana paralel silsile vardır. (i) İzleme modülünde kameradan alınan görüntülerden ORB özellikleri çıkarılır durum ile ilgili tahmin veya yeniden konumlama yapılır ve gerektiğinde yeni anahtar karelerin eklenmesi veya eklenmemesine karar verilir. Çıktısı, ikinci modül olan (ii) lokal haritalamaya verilir. Bu modülde eğer yeni bir anahtar kare geldiyse bunun entegrasyonu sağlanır ve farklı ayrıştırma basamakları ile (iii) döngü kapama modülüne girer olarak verilir. Döngü kapama modülü her bir eklenen yeni anahtar kare ile mevcut durumda bir eşleşme olup olmadığını kontrol eder. Eşleşmenin olduğu durumlarda benzerlik dönüşümü ile süreçte meydana gelen kaymalar hesaplanır ve buna göre düzeltme yapılır. Bunların yanı sıra kapalı döngü tanımlanması ve yeniden konumlama işlemleri için algoritma kelime sözlüğü (BoW) yöntemini kullanmaktadır. BoW'a göre elde edilen görsel sözlük büyük bir resim veri setinden çevrimdışı elde edilmiştir. Eğer ortamla ilgili fazla bir değişiklik yoksa aynı görsel sözlüğü farklı uygulamalarda da kullanmak mümkündür. Harita başlangıçlandırılması için düzlemsel görüntü homografisi, aksi durumlar için temel matris varsayımları kullanılmıştır.

Düzenleştirilmiş Tek Kameralı Derinlik Kestirimi (REMODE) ismiyle adlandırılan teknik ise gerçek zamanlı alınan görüntülerden ortamın yoğunluk haritasının çıkarılması ile ilgilidir (Pizzoli ve ark., 2014). Yöntemin temelinde tek gözlü bir kameradan alınan görüntülerin Bayes tahminleri ve dış bükey optimizasyon tekniklerinin birleştirilmesi ile yoğunluk haritasını elde etme esası bulunmaktadır. Bu işlem yapılırken piksel tabanlı tahminler üzerinden yoğunluk haritası oluşturulmuştur. vSLAM tekniklerinde elde edilen özellik tabanlı haritaların robotik uygulamalarında engelden kaçınma, rota planlama gibi durumlar için tam olarak sorunlara cevap veremeyeceği düşüncesi bu çalışmanın temelini oluşturmuştur. Bu sayede robotik uygulamalar için yoğunluk tabanlı bir haritanın daha anlamlı olacağı düşünülmüştür.

ORB-SLAM yöntemini her ne kadar görsel SLAM metotları için hızlı bir çözüm sunsa da elde edilen sonuç haritası özellik tabanlı olup bazı durumlarda derinlik tabanlı haritalara ihtiyaç duyulur. Özellikle ortamın nihai üç boyutlu temsili istendiğinde bu türde

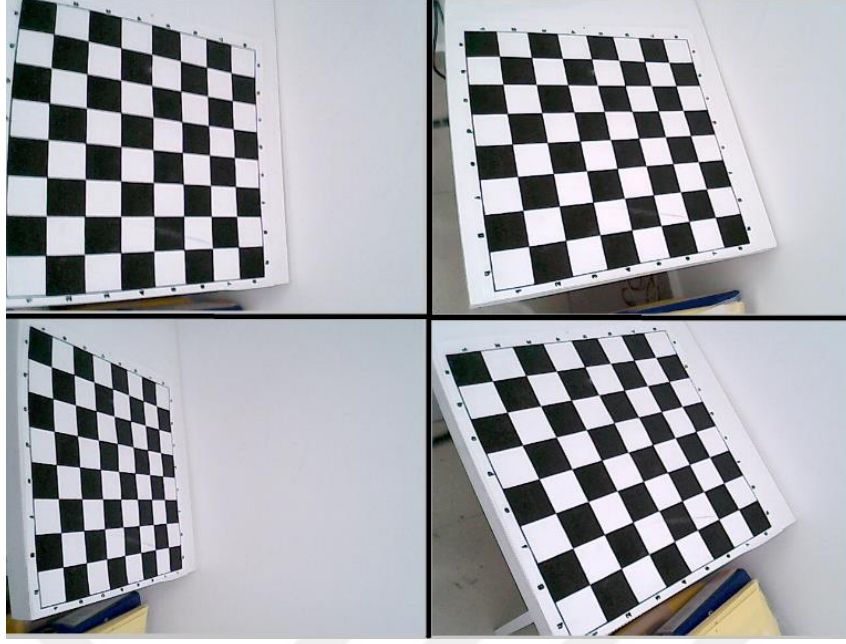
bir yöntem eksik kalmaktadır. Özellik tabanlı çıktı üreten yöntemlerin bu olumsuzlukları REMODE çalışmasında da özellikle dile getirilmiştir (Pizzoli ve ark., 2014). Bu eksiklikten yola çıkılarak tez kapsamında ORB-SLAM metodu bir diğer üç boyutlu görselleştirme yöntemi olan ve REMODE olarak adlandırılan bir yöntemle birleştirilerek ORB-SLAM çalışması için bir katkı sağlanması amaçlanmıştır.

Bu kapsamda çalışmanın gerçekleşmesi ve yöntemin etkinliğinin ortaya koyulabilmesi için ucuz, çözünürlüğü düşük bir tek gözlü kameradan yararlanılmıştır (Şekil 4.22).



Şekil 4.22. Görsel SLAM tekniklerinin kullanıldığı çalışmada kullanılan kamera

Kamera maksimum 640×480 video çözünürlüğü, 350K CMOS sensör, 8 MP çözünürlük, $F = 2.4$, $f = 3.0$ mm lens özelliklerine sahiptir. Kameranın uygulamada kullanılması için kalibrasyon yapılması ve ilgili parametrelerin yazılıma girdi olarak verilmesi gerekmektedir. Buna göre kalibrasyon için klasik satranç tahtası kalibrasyon düzeneği kullanılmıştır. Şekil 4.23'te görüleceği üzere kamera ile ortamdan bazı görüntüler alınmıştır.



Şekil 4.23. Kamera kalibrasyonu için bazı standart kalibrasyon görüntüleri

Kamera kalibrasyonu yapıldıktan sonra ilgili parametreler OpenCV formatında elde edilmiştir. Kamera kalibrasyon ve bozulma değerleri aşağıdaki gibidir.

$$K = \begin{bmatrix} 658.9585 & 0 & 0 \\ s & 659.2656 & 0 \\ 328.5526 & 268.1391 & 1 \end{bmatrix}; D = [0.2080, -0.9396, 0, 0, 0]$$

burada K iç matrisi D (Eşitlik (4.7)) ise bozulma parametrelerini ifade etmektedir.

$$\text{Radyal düzeltme} \rightarrow \begin{cases} x_c = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_c = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases} \quad (4.7)$$

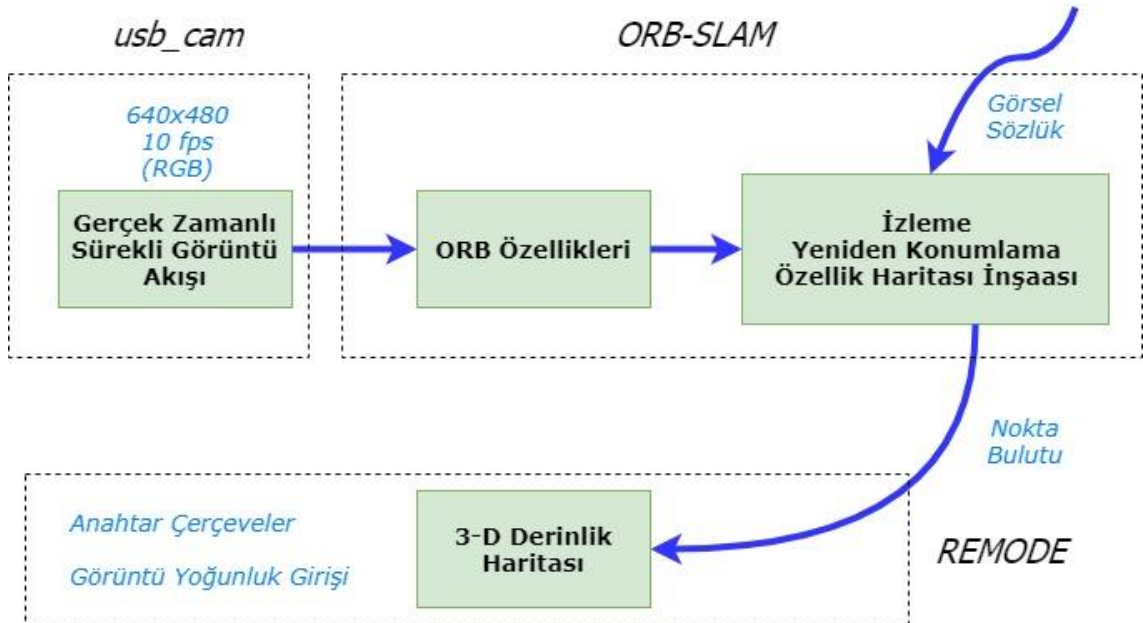
$$\text{Teğetsel düzeltme} \rightarrow \begin{cases} x_c = x + (2p_1xy + p_2(r^2 + 2x^2)) \\ y_c = y + (2p_2xy + p_1(r^2 + 2y^2)) \end{cases}$$

Elde edilen kalibrasyon parametreleri ve geliştirilen algoritma çerçevesinde Şekil 4.24'te görülen bir deneme ortamı için çalışma yapılmıştır.



Şekil 4.24. Görsel SLAM tekniklerinin kullanıldığı çalışma için hazırlanan ve haritalaması yapılacak olan sahneye dair bir görüntü

Oluşturulan ortamda denemelerin yapılabilmesi için Şekil 4.25'te blok diyagramı görülen bir algoritma oluşturulmuştur. Buna göre ortamdan Şekil 4.22'de gösterilen kamera aracılığıyla sürekli bir şekilde RGB düzeyinde görüntüler alınmıştır. Bu görüntülerden ORB özellikleri elde edilmiş ve ORB-SLAM algoritmasının özelliklerinden olan görsel sözlük ile işleme tabi tutulmuştur. Buradan elde edilen çıktılar REMODE yöntemine gönderilerek ortam ile ilgili derinlik haritası oluşturulmuştur.



Şekil 4.25. Görsel SLAM tekniklerinin kullanıldığı çalışmaya dair genel girdi ve çıktıları gösteren algoritma blok diyagramı

Şekil 4.25'te blok diyagramı verilen algoritma ORB-SLAM2⁹ ve REMODE¹⁰ olmak üzere farklı iki yöntemin birleştirilmesiyle oluşmaktadır. Algoritmanın gerçekleşmesi için çalışma içerisinde bahsedilen kamera kullanılmış ve Bölüm 4.2'de değinilen ROS *usb_cam* paketinden yararlanılmıştır. Kameradan alınan görüntüler bu pakette de varsayılan değer olarak kullanılan 640×480 boyutundadır. Bununla birlikte saniyede alınan kare sayısı (fps) paket varsayılan değeri 30 fps olmasına rağmen algoritma hızını artırmak için 10 fps olacak şekilde seçilmiştir. *usb_cam* paketi kameradan aldığı görüntüleri *kamera_ismi/Image* başlığından *sensor_msgs/Image* türünde yayımlamaktadır. Daha sonra bu başlığa abone olunarak kameradan alınan ortam görüntüleri ORB-SLAM algoritmasına uygulanmıştır. ORB-SLAM algoritmasında tanımlanan saniyede alınan görüntü sayısı, resimdeki özellik sayısı, en küçük başlangıç özellik sayısı vb. gerçek zamanlı bir uygulama sonucunu etkileyebilecek birçok parametre vardır. Uygulamanın yapıldığı ortam ve kameranın türüne göre farklı değerlerin kullanılması gerekebilir. Bu uygulama insan eliyle hareket ettirilen düşük çözünürlüklü bir kamera kullanılarak bir iç ortam sahnesinin haritasının elde edilmesi ile ilgilidir. Bu yüzden saniyede alınan görüntü sayısı, insan eliyle yapılan hareketler ve kamera çözünürlüğünün başarımı etkileyeceği aşikardır. İnsan eli hareketleri göz önüne alındığında yüksek özellik sayısının seçilmesi peşi sıra gelen görüntülerdeki özelliklerin takip edilmesini kolaylaştıracaktır. Bununla birlikte kamera çözünürlüğünün az olması başlangıçlandırma aşamasını geciktirebileceği için az bir özellikle başlama koşulunun oluşturulması gereklidir. Bunların zıttına, bir iç ortamda çok fazla görüntü değişiminin olmadığı göz önüne alınırsa düşük fps değerlerinin seçilmesi çok fazla bir problem oluşturmayacak ve çalışma hızını artıracaktır. Örneğin bu uygulamada, saniyede alınan kare sayısının 30 ve 10 olarak seçildiği durumlar incelenmiştir. 30 fps koşulunda çalışma hızı azalmasına rağmen sonuç başarımı çok fazla değişmemiştir. Benzer şekilde resim özellik sayısının 500 ve 2000 seçildiği denemeler yapılmıştır. 500 özellik seçildiğinde kamera hareketleri yapılırken ORB-SLAM algoritmasının peşi sıra gelen görüntülerin takibinde başarısızlıklar olduğu görülmüştür. Buna paralel bir şekilde başlangıç özellik sayısının 7 seçildiği durumlarda kamera çözünürlüğünün düşük olmasından dolayı başlangıçlandırma işleminin yapılamadığı gözlemlenmiştir. Bu düşünceyle ORB-SLAM parametreleri bu uygulama için bir dizi deneme sonucunda

⁹ http://wiki.ros.org/orb_slam2_ros

¹⁰ https://github.com/uzh-rpg/rpg_open_remode

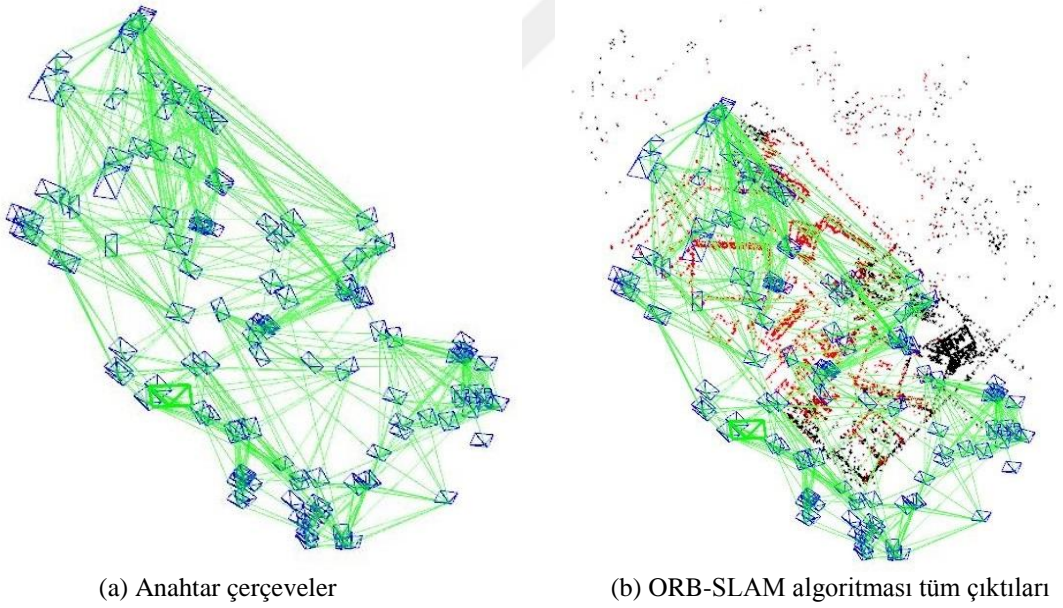
sırasıyla 10 fps, bir resim için 2000 özellik ve başlangıçlandırma için en az 3 özelliğin bulunması şeklinde seçilmiştir. Diğer ORB-SLAM parametrelerinin değişiminin uygulama sonuçlarına fazla etkisi olmadığı görülmüş ve algoritma varsayılan değerleri kullanılmıştır. ORB-SLAM algoritması *ORB/DenseInput* başlığından nokta bulutu verileri yayınlamaktadır. Bu başlık çıktısı REMODE yöntemine uygulanarak yoğunluk tabanlı haritalar elde edilmiştir. REMODE yönteminde tanımlanan farklı parametrelerin çalışma sonuçlarına çok fazla bir etkisi olmadığı gözlemlendiği için varsayılan değerler kullanılmıştır.

Benzer uygulamaların farklı ortamlarda yapılabilmesi için ilk olarak kullanılacak kameranın kalibrasyonu yapılmalı ve ilgili parametreler elde edilmelidir. Daha sonra bu parametreler ve kameradan *usb_cam* paketi ile alınan görüntüler ORB-SLAM algoritmasına uygulanmalıdır. Eğer stereo veya RGB-D kameralar kullanılmak istenirse *usb_cam* paketi yerine farklı türde paketler kullanılarak görüntüler elde edilebilir. Bununla birlikte çalışma ortamına veya gereksinim duyulan algoritma hızına göre saniyede alınan kare sayısı belirlenmelidir. Üçüncü olarak tek gözlü, stereo veya RGB-D kamera türüne göre ORB-SLAM yönteminden ilgili ROS çalıştırma düğümü seçilmelidir. Ayrıca kamera ve uygulamaya göre en uygun ORB-SLAM parametreleri saptanmalıdır. En son olarak ORB-SLAM çıktısı REMODE yöntemine girdi olarak verilmelidir.

Algoritma sonucunda ortamdan elde edilen özellik tabanlı harita, anahtar kareler ve bunlar arasında oluşan bağlantılar izleyen şekillerdeki gibidir (Şekil 4.26 ve 4.27).



Şekil 4.26. ORB-SLAM yöntemi kullanılarak elde edilen özellik tabanlı nihai harita



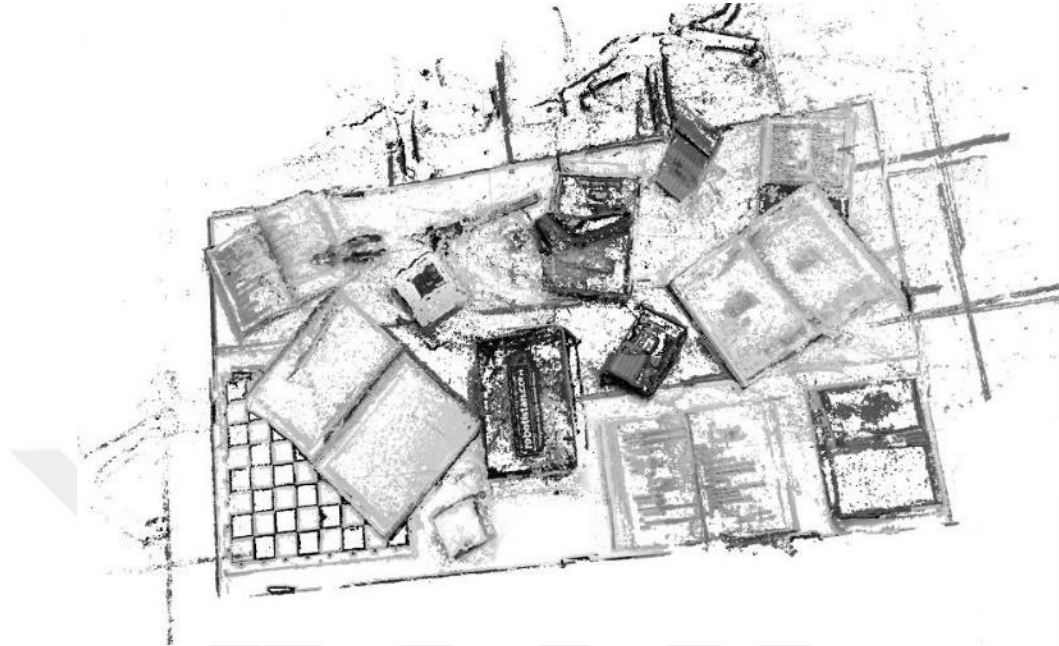
(a) Anahtar çerçeveler

(b) ORB-SLAM algoritması tüm çıktıları

Şekil 4.27. Görsel SLAM tekniklerinin kullanıldığı çalışmaya dair yalnızca ORB-SLAM yöntemi çıktıları; (a) Anahtar çerçeveler ve tüm graf içerisindeki ara bağlantılar, (b) ORB özellikleri, anahtar çerçeveler ve bağlantılarının hepsinin gösterildiği nihai harita

ORB-SLAM metodu ile yukarıda görüleceği üzere ortamdan özellik tabanlı görüntüler elde edilebilmektedir. Geliştirilen yöntem çerçevesinde ortamın derinlik

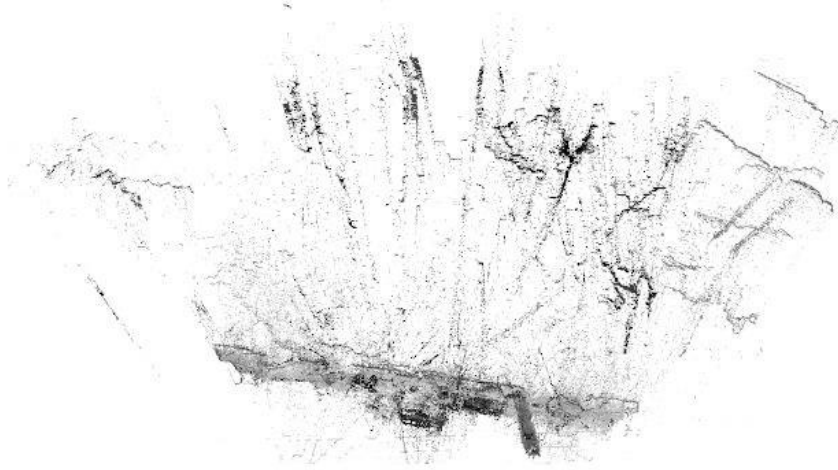
yoğunluk tabanlı haritaları izleyen şekillerdeki gibidir (Şekil 4.28'den Şekil 4.30'a kadar).



Şekil 4.28. Görsel SLAM tekniklerinin kullanıldığı çalışmaya dair ORB-SLAM yöntemi temel alınarak geliştirilen yöntem sonucu elde edilen derinlik haritası, yakın görünüm



Şekil 4.29. Görsel SLAM tekniklerinin kullanıldığı çalışmaya dair ORB-SLAM yöntemi temel alınarak geliştirilen yöntem sonucu elde edilen derinlik haritası, uzak görünüm



Şekil 4.30. Görsel SLAM tekniklerinin kullanıldığı çalışmaya dair ORB-SLAM yöntemi temel alınarak geliştirilen yöntem sonucu elde edilen derinlik haritası, yan görünüm

Özellik tabanlı haritalar kullanımı açısından derinlik/yoğunluk haritalarına göre daha az görsel bilgi içermektedir. Bu alt başlıkta bu sorun üzerinde durularak ORB-SLAM yönteminde karşılaşılan bu problemin farklı bir bakış açısıyla ele alınması hedeflenmiştir. Bu sonuçlara göre elde edilen veriler Korkmaz ve Durdu (2017) çalışmasında yayınlanmıştır.

4.7. Yaya Navigasyonu Başlangıç Problemi için Geliştirilen Yöntem

Konumlama, haritalama veya SLAM konuları robotlar üzerinde incelendiği gibi insanlar için de ele alınmıştır. Bu konu ile ilgili Bölüm 1 ve 2’de literatürde var olan bazı çalışmalardan bahsedilmiştir. Tez kapsamında geliştirilen yaklaşımda başlangıç konumu bilinmeyen yaya konumlanması problemi üzerinde durulmuştur. Bu kapsamda yeni bir metot olan harita eşleme tabanlı yinelemesiz Bayesian filtresi kullanılmıştır (Koroglu ve Yılmaz, 2017).

Yaya konumlama çözümlerine GPS verisinden yoksun bölgelerdeki konumlama durumlarında ihtiyaç duyulmaktadır. Bunun için en sık kullanılan sensör IMU’lardır. IMU’ların doğrusal/açısal hızlanma ve manyetik alan bilgileri ile yaya konumlama problemi çözülebilmektedir. Bununla birlikte IMU’nun vücutta yerleştirildiği bölgeye göre de farklı algoritmaların kullanılması söz konusudur. Bu çalışmada incelenen SparkFun firmasına ait 9 eksenli IMU¹¹, bir yayanın ayakkabısı üzerine yerleştirilmiş ve

¹¹ <https://www.sparkfun.com/products/14001>

gerçek zamanlı veriler bu test düzeneği ile elde edilmiştir. Elde edilen veriler ile MATLAB programı üzerinde denemeler gerçekleştirilmiştir.

Bir insanın yürüyüş modeli incelendiğinde adımlar arası geçişlerde durağan olan kısmın ayak bölgesi olduğu görülmektedir (Şekil 4.31).



Şekil 4.31. İnsan yürüme modeli ve farklı basamakların gösterimi

Ayağın durağan olduğu zaman dilimlerinde ayak hızının kısa bir süreliğine de olsa sıfır olacağı aşikardır. Algoritmada ayak hızının sıfırdan farklı olarak algılandığı durumlar hatayı ifade eder ve farklı filtrelere (sıfır geçiş hızı - ZUPT) girdi olarak verilerek tahmin ve durum güncellemesi olarak kullanılabilir. Durum hataları ile birleştirilmesi durumu hata durumlu KF olarak bilinir (ESKF).

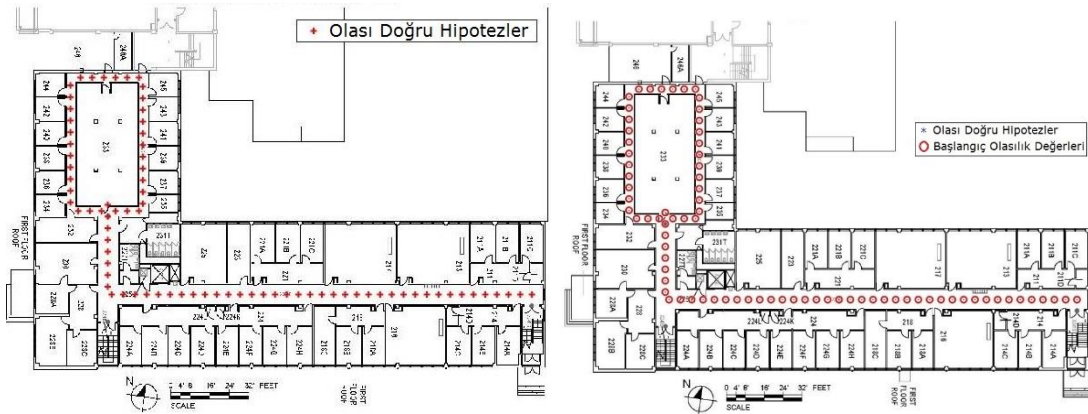
ZUPT tabanlı girdiler açısız yönelimi düzeltmede doğrusal mesafeyi tahmin etmedeki gibi başarılı değildir. Bu yüzden genellikle harita eşleme yöntemleri kullanılarak iyileştirmeler yapılır. Bu çalışma içeriğinde de konumlama verilerinin düzeltilmesi için harita eşleme ve bina bilgi modellerinden (BIM) yararlanılmıştır. Klasik parçacık filtresi harita eşleme yönteminde, yayanın olası bir sonraki adımı etrafında parçacıklar oluşturulurken tahmin adımında durum yayılma modeli kullanılır. Bunun yanı sıra, güncelleme adımında ise parçacık ağırlıkları oluşturulurken duvar sınırları vb. değişkenler dikkate alınır. Bu da klasik yinelemeli döngüyü oluşturur. Öte yandan, Bayesian harita eşleme yönteminde ise tarama haritalarından (raster haritası) yararlanılır. Tarama haritaları konumla ilgili önsel olasılıkları içermektedir. Bu sayede konum bilgilerine dair olasılıklar, duvar konumları, harita dışındaki bölgeler benzerlik olasılığına sahip olmaktadır. Böylece parçacık filtresi yaklaşımından farklı olarak bir sonraki durum tahmini hesaplama yöntemi ile elde edilmez, bunun aksine önceden BIM modeline göre oluşturulmuş tarama haritalarından bulunur. Bu da yöntemi kestirim sonuçları açısından yinelemesiz yapmaktadır.

IMU ile yapılan çalışmalarda IMU'nun kendi koordinat düzlemi ve çalışılan ortamın koordinat düzlemleri arasındaki ilişki bilinmelidir. Bu bilgi örneğin başlangıç

konumu ve hareket yönü olabilir ve algoritmaya girdi olarak verildiğinde veri eşleştirilmesi yapılarak konumlama problemi çalışılan ortam için çözülebilir. Bu bilginin olmadığı durumlar da başlangıç konumu ve hareket yönü probleminin çözümü araştırılmalıdır. Bunun için tez kapsamında çoklu hipotez testi yaklaşımı geliştirilmiştir. Geliştirilen yöntem, yürünebilir yol ve yol üzerinde oluşturulan olası başlangıç hipotezlerini esas almaktadır. Bunun için ilk olarak harita üzerinde tanımlanan yürünebilir yollar farklı uzaklıklarda örneklenmiştir. Buna bağlı olarak farklı sayıda hipotezler oluşturulmuş ve bu hipotezlerin değerleri yaya adımları sonrasında güncellenmiştir. Böylece en iyi değeri veren hipotez başlangıç noktası olarak bulunmuştur. Örnek bir uygulama için bir koridor ortamı seçilmiştir (Şekil 4.32a). Bu ortamda yaya ayağı üzerine bağlanan IMU ile veriler toplanmıştır. Başlangıç konumu bilinmediği için ortamda tanımlanan yürünebilir yol üzerinde başlangıç hipotezleri oluşturulmuştur. Bu hipotezlerin arasındaki mesafe 30 pixel (~1.46 metre) olacak şekilde ayarlanmıştır.

Çalışmanın başında yayanın örneklenen bütün olası hipotezlerden başladığı varsayımı yapılmıştır. Bu yüzden ilk durumda bütün hipotezin ($H_1 \dots H_n$) değerleri birbirine eşit olmaktadır (Eşitlik (4.8) ve Şekil 4.32b, kırmızı daireler).

$$P(H_1 = x) = \dots = P(H_n = x) \quad (4.8)$$



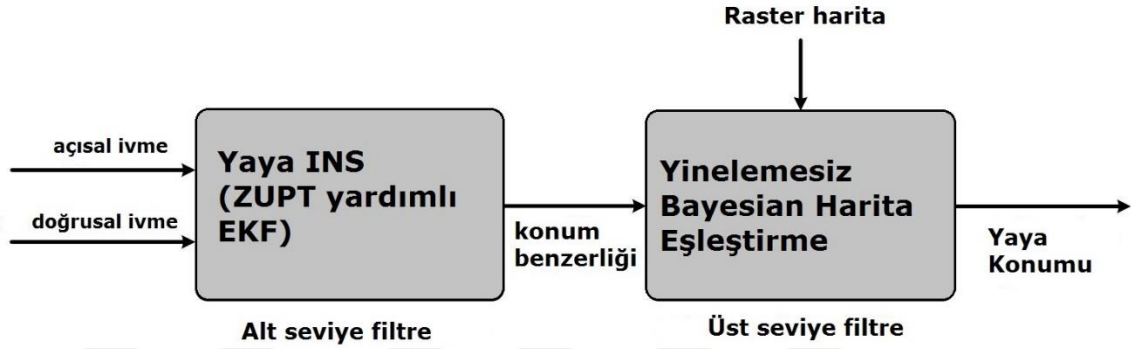
(a) Olası başlangıç hipotez noktaları

(b) Başlangıç hipotezleri normalize olasılık değerleri

Şekil 4.32. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotezler; (a) Örneklem yoluyla yürünebilir yol üzerinde oluşturulmuş başlangıç hipotezleri ve (b) başlangıç hipotezlerinin olasılık değerleri (daireler, hipotezlerin normalize edilmiş olasılık değerlerini ifade eder ve dairelerin çapı ile hipotezlerin olasılık değerleri doğru orantılıdır) (birinci deneme ortamı)

Başlangıç durumu olasılığını gösteren x' in değeri herhangi bir rassal değişken veya $1/n$ olarak seçilebilir. n kuantalama değeri sonucu elde edilen hipotez sayısını göstermektedir.

Yaya ortamda hareket ettikçe yanlış hipotezlere dair olasılık değerleri giderek azalacak ve bunun sonucunda en iyi değeri veren doğru hipotez elde edilecektir. Çalışmada yaya konumunun bulunmasına dair blok diyagram Şekil 4.33'teki gibidir.



Şekil 4.33. Yaya navigasyonu problemi için kullanılan yinelemesiz Bayesian harita eşleştirme metodu tabanlı yöntem için genel blok diyagram

Hareket devam ettikçe her bir adım sonucunda hipotezlerin olasılık değerleri güncellenmektedir. Herhangi bir hipotezin herhangi bir zamandaki olasılık değeri önceki durumlarla ilişkili olduğu için zincir kuralına göre hipotezlerin olasılık değerleri ifade edilebilir (Eşitlik (4.9)).

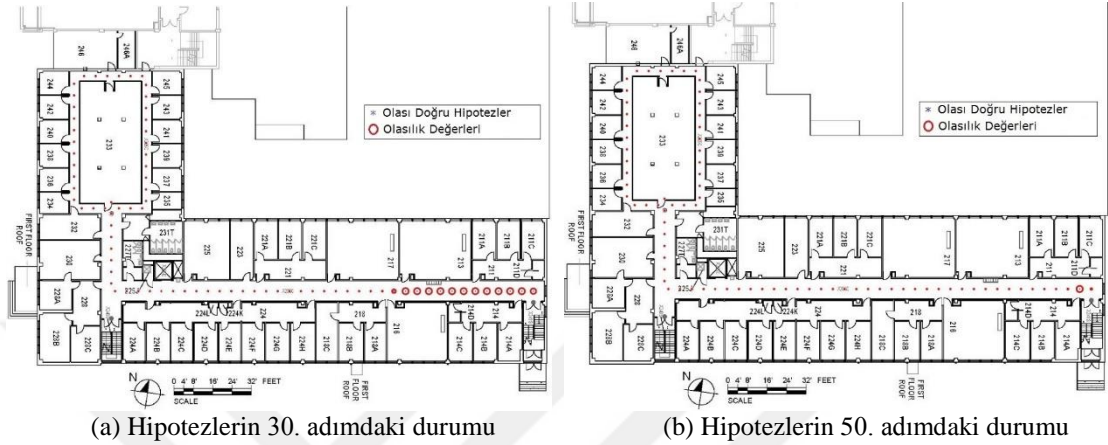
$$P(H) = P(H_s \cap \dots \cap H_1) = \prod_{k=1}^s P(H_k | \cap_{j=1}^{k-1} H_j) \quad (4.9)$$

burada s adım sayısını göstermektedir.

Eşitlik (4.9)'a göre çalışma sonucunda en yüksek olasılık değerine sahip olan hipotez doğru başlangıç noktası olarak seçilebilir. Ancak IMU ve harita koordinat eksenleri arasındaki yön eşleşmesi bilinmiyorsa bu parametre de bilinmeyen duruma dahil edilmelidir. Bu çalışmada bu iki koordinat eksenini arasındaki yön bilgisinin bilinmediği ve yayanın kardinal hareket yönleri olan doğu, batı, kuzey, güney eksenleri doğrultusunda hareket ettiği varsayılmıştır. Böylece en iyi olasılık değerini veren hipotez Eşitlik (4.10)'a göre seçilebilir.

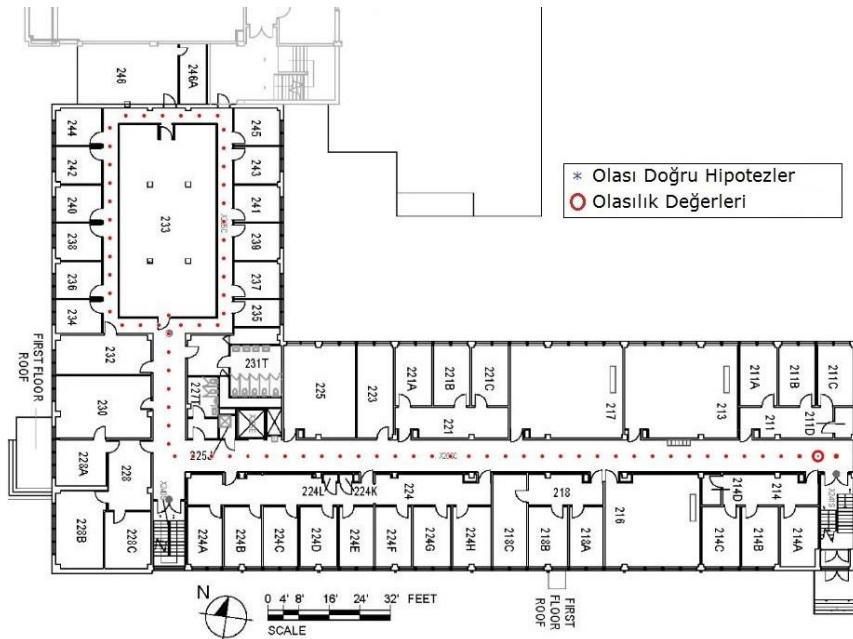
$$\arg \max_{i \in [D], j \in [E]} (i, j | P(H_i^j) \geq P(H_k^l), P(H_k^l) \in A, \forall k \in D \wedge l \in E) \quad (4.10)$$

burada, $[D] = \{1, \dots, n\}$, $[E] = \{1, \dots, m\}$, $[A] = \{P(H_1^1), \dots, P(H_1^m), \dots, P(H_n^1), \dots, P(H_n^m)\}$ şeklinde ifade edilir ve n toplam hipotez sayısını, m yön bilgisini, A ise toplam hipotez kümesini temsil etmektedir. Bu bilgiler ışığında çalışma sırasında farklı adım değerlerindeki hipotezlerin normalize edilmiş olasılık değerleri Şekil 4.34'teki gibidir.



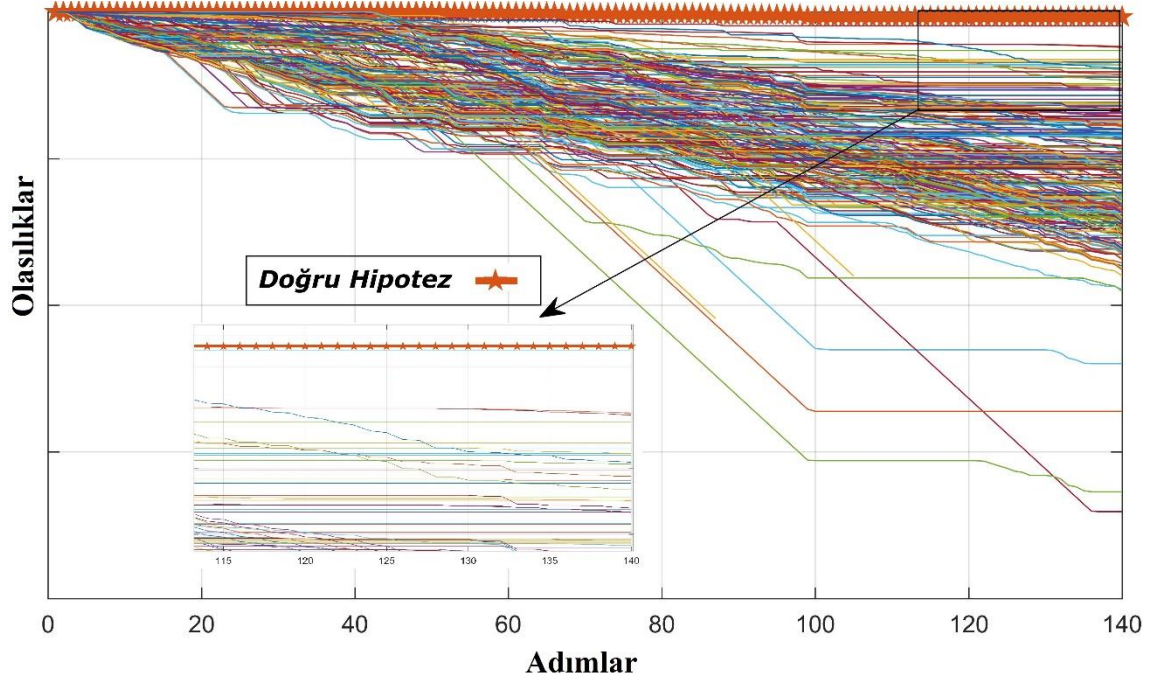
Şekil 4.34. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait farklı adımlardaki olasılık değerleri; (a) Hipotezlerin 30. adımdaki değerleri, (b) Hipotezlerin 50. adımdaki değerleri (daireler, hipotezlerin normalize edilmiş olasılık değerlerini ifade eder ve dairelerin çapı ile hipotezlerin olasılık değerleri doğru orantılıdır) (birinci deneme ortamı)

Şekil 4.35'te ise çalışma sonunda hipotezlerin normalize edilmiş olasılık değerlerine göre temsilleri gösterilmiştir.



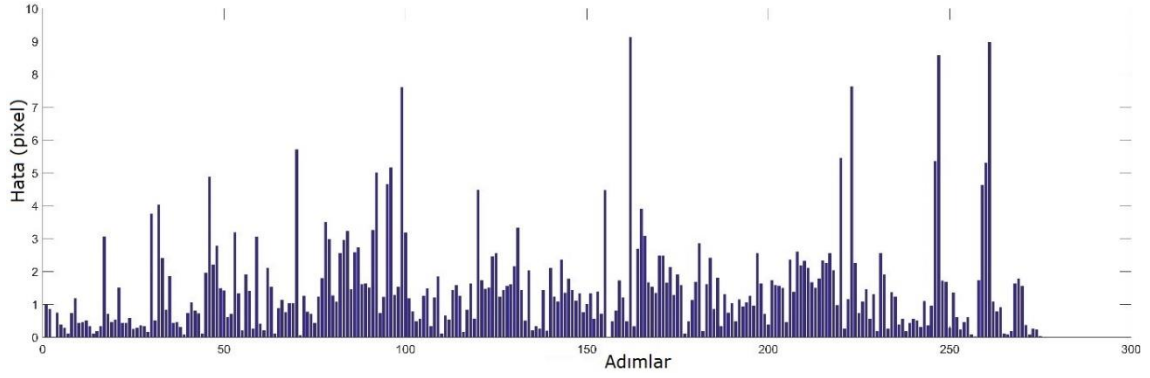
Şekil 4.35. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait deney sonunda elde edilen hipotezlerin normalize edilmiş olasılık değerleri (doğru başlangıç noktası en büyük çaplı daireye sahip olan 2. başlangıç hipotezidir) (birinci deneme ortamı)

Burada da görüleceği üzere 50. ve son adım arasında hipotezlerin normalize edilmiş olasılıkları arasında bir benzerlik bulunmaktadır. Bu da doğru hipotezin çalışma bitmeden çok daha önce bulunabileceği ve bu sayede çalışma hızının artırılabilceğini göstermektedir. Bununla birlikte çalışma boyunca tüm hipotezlerin olasılık değerleri kayıt edilmiştir. Şekil 4.36'dan da görüleceği üzere doğru hipotezin olasılığı, çalışma boyunca korunmuştur. Öte yandan yanlış hipotezlerin değerleri adım sayısı arttıkça azalmıştır.

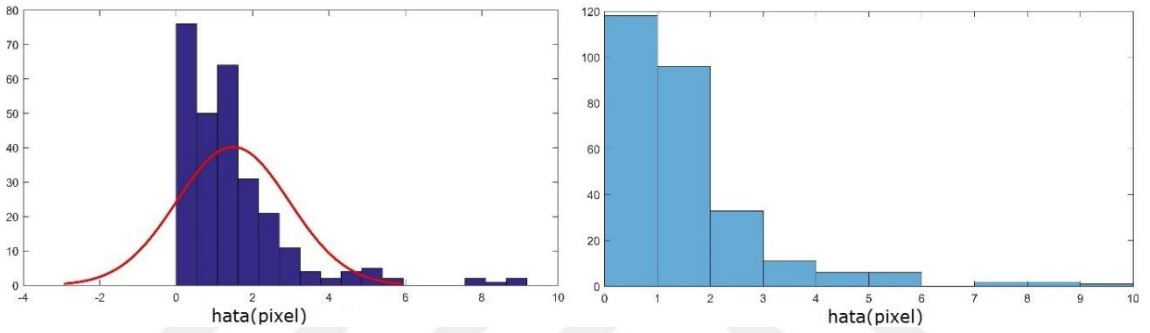


Şekil 4.36. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait deney süresince kayıt edilmiş tüm hipotezlerin olasılık değerleri (birinci deneme ortamı)

Bununla birlikte uygulama sırasında gerçekte ayak basılan yerler ve yinelemesiz Bayesian filtresi sonucunda ortaya çıkan adımlara ait konum noktaları arasındaki fark karşılaştırılmıştır. Bunun için algoritma sonucu ortaya çıkan konumlar nitel, çalışma ortamında tanımlanan yol üzerindeki değerler ise nicel değerler olarak alınmıştır. Buna göre karşılaştırma Şekil 4.37 ve 4.38'deki gibi olmaktadır.



Şekil 4.37. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait adımların kantitatif ve kalitatif sonuçlarının karşılaştırılması (birinci deneme ortamı)

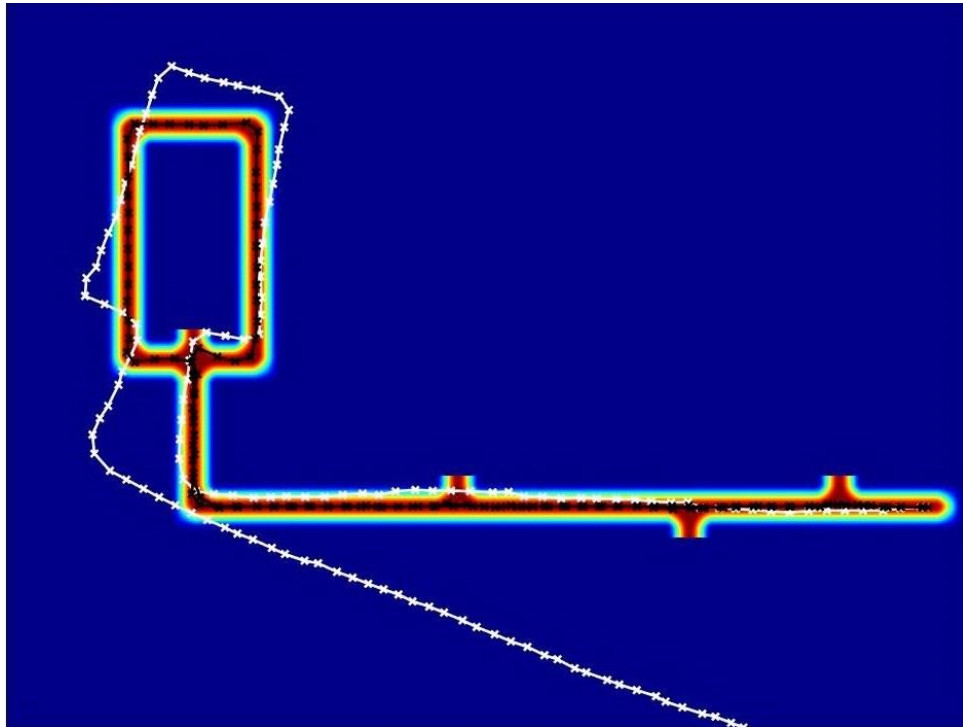


Şekil 4.38. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait adımların kantitatif ve kalitatif sonuçlarının karşılaştırılması; hata dağılımı ve bar temsili (birinci deneme ortamı)

Bu bilgiler ışığında elde edilen doğru başlangıç noktasına göre ZUPT çıktısı yaya yol eğrisi ve yinelemesiz harita eşleme yöntemi ile düzeltilmiş doğru yaya eğrisi Şekil 4.39'daki gibidir. Çalışma boyunca kullanılan tarama haritası ise Şekil 4.40'teki gibidir.

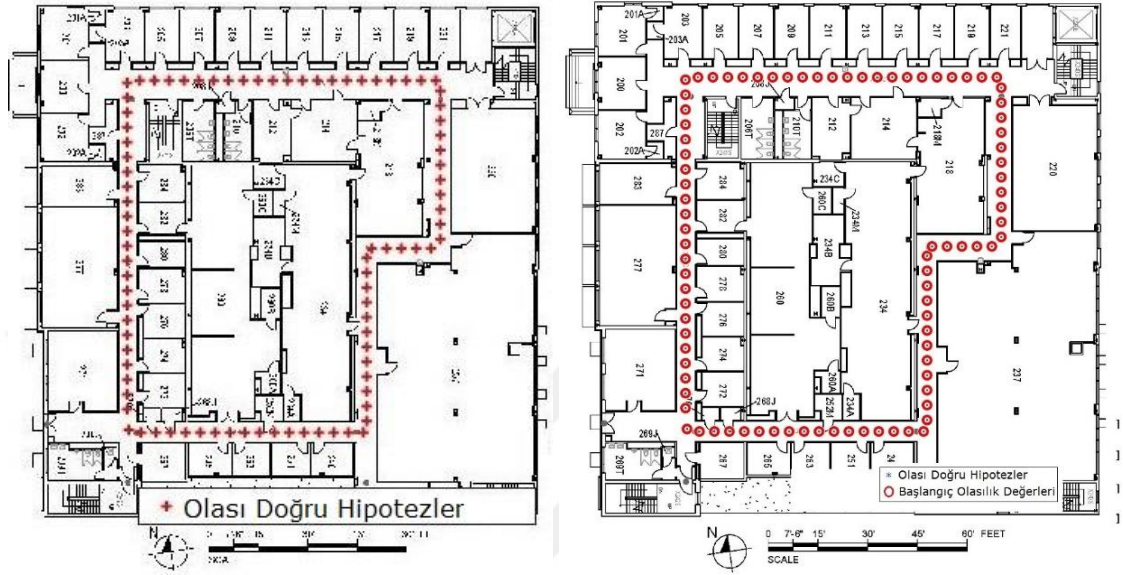


Şekil 4.39. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait doğru hipotez noktasından başlanması durumunda elde edilen eğriler: Kesikli çizgiler, ZUPT yardımcı yaya eğrisini; Düz çizgiler, yinelemesiz Bayesian harita eşleştirme yöntemine göre düzeltilen eğrileri temsil eder (birinci deneme ortamı)



Şekil 4.40. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait kullanılan tarama haritası; mavi bölgeler düşük olasılıklı bölgeler, kırmızı bölgeler en yüksek olasılıklı bölgeler (birinci deneme ortamı)

Sonuçları doğrulamak adına aynı algoritma farklı bir ortamda da denenmiştir. Bir önceki deneye benzer şekilde bu ortam koridoru üzerinde yürünebilir yol tanımı yapılmıştır. Koridor üzerinde tanımlanan yürünebilir yol, belirli bir uzaklıkta olacak şekilde örneklendirilerek başlangıç hipotezleri oluşturulmuştur (Şekil 4.41a).

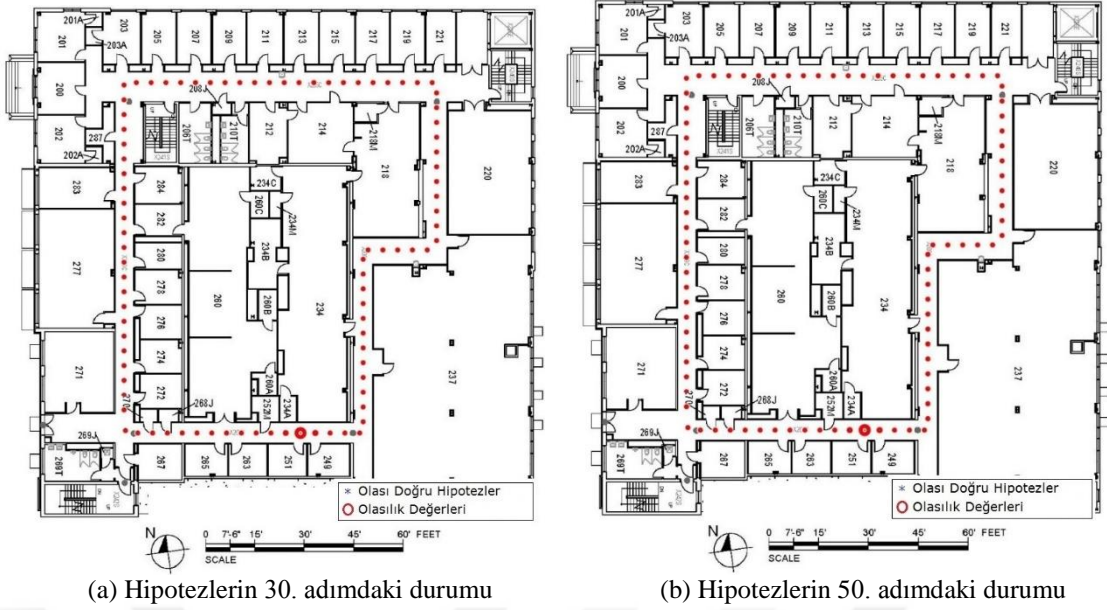


(a) Olası başlangıç hipotez noktaları

(b) Başlangıç hipotezleri normalize olasılık değerleri

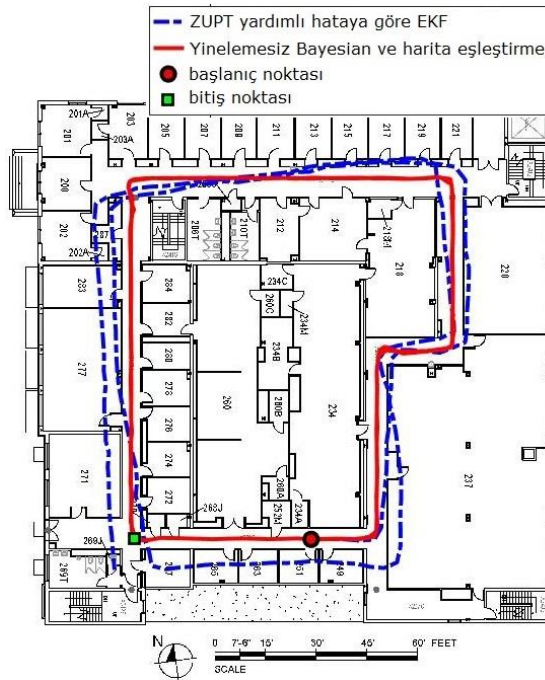
Şekil 4.41. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotezler; (a) Örnekleme yoluyla yürünebilir yol üzerinde oluşturulmuş başlangıç hipotezleri ve (b) Başlangıç hipotezlerinin olasılık değerleri (daireler, hipotezlerin normalize edilmiş olasılık değerlerini ifade eder ve dairelerin çapı ile hipotezlerin olasılık değerleri doğru orantılıdır) (ikinci deneme ortamı)

Bahsedildiği üzere yaya ortamda yürüdükçe yanlış hipotezler kaybolmakta ve buna bağlı olarak doğru başlangıç noktası bulunmaktadır (Şekil 4.42).



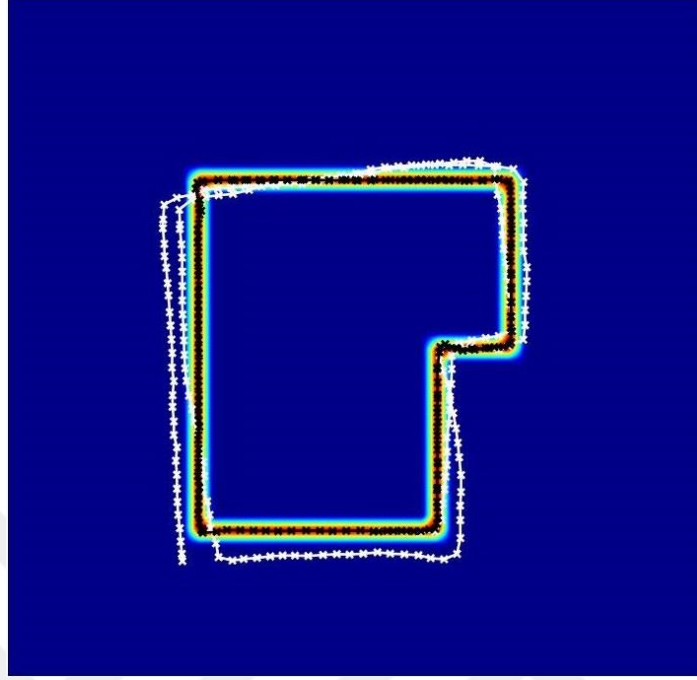
Şekil 4.42. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait farklı adımlardaki olasılık değerleri; (a) Hipotezlerin 30. adımdaki değerleri, (b) Hipotezlerin 50. adımdaki değerleri (daireler, hipotezlerin normalize edilmiş olasılık değerlerini ifade eder ve dairelerin çapı ile hipotezlerin olasılık değerleri doğru orantılıdır) (ikinci deneme ortamı)

Bu çalışmanın sonuçlarında da görüleceği üzere doğru hipotez önceki çalışmaya göre çok daha erken bulunabilmektedir. Bu bilgiler ışığında bulunan doğru hipotez başlangıç noktasına göre yayanın izlediği yol ve düzeltilmiş eğrisi Şekil 4.43'teki gibidir.



Şekil 4.43. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait doğru hipotez noktasından başlanması durumunda elde edilen eğriler: Kesikli çizgiler, ZUPT yardımcı yaya eğrisini; Düz çizgiler yinelemesiz Bayesian harita eşleştirme yöntemine göre düzeltilen eğrileri temsil eder (ikinci deneme ortamı)

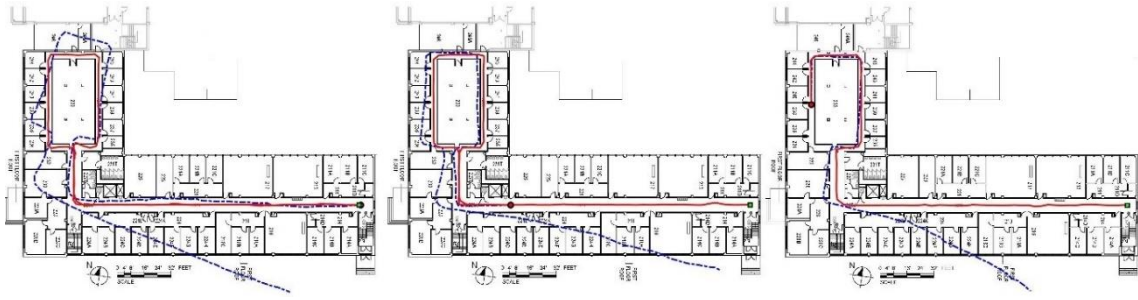
Bu çalışmada da kullanılan tarama haritası Şekil 4.44'teki gibidir.



Şekil 4.44. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait kullanılan tarama haritası; mavi bölgeler düşük olasılıklı bölgeler, kırmızı bölgeler en yüksek olasılıklı bölgeler (ikinci deneme ortamı)

IMU uygulamalarında çalışma süresi arttıkça doğru yol eğrisinden sapma miktarı artmakta ve konumlama probleminin çözümü zorlaşmaktadır. Bu düşünceyi test etmek adına bir önceki çalışmadan farklı olarak ortamda tek bir tur yerine birden fazla tur atılmıştır. Bununla birlikte kapalı bir çevrim içermesi açısından farklı bir karaktere sahip ortamdaki sonuçlarda irdelenmek istenmiştir. Bu özelliklere sahip bir ortamda yapılan çalışma sonuçlarına göre başlangıç noktasına dair hipotez bir önceki uygulamada olduğu gibi doğru bir şekilde bulunmuştur.

İki farklı ortamda toplamda beş farklı başlangıç noktası seçilerek veriler toplanmıştır. Ortamlar 20 , 30 , 40 ve 60 pixel uzaklıklarında kuantize edilerek varsayılan olası başlangıç hipotezleri oluşturulmuştur. Birinci uygulama ortamına dair üç farklı başlangıç noktasından başlanması durumu ve ilgili sonuçları Şekil 4.45'teki gibidir.



Şekil 4.45. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait birinci ortam için üç farklı başlangıç noktasından başlanması durumu ve buna bağlı düzeltilmiş yaya navigasyon eğrileri (birinci deneme ortamı)

Toplamda 20 farklı deneme için sonuçlar elde edilmiş ve buna bağlı olarak doğru başlangıç noktaları bulunmuştur. Çizelge 4.9’da farklı kuantize noktalarına göre her bir ortam için gerçek ve bulunan başlangıç noktaları gösterilmiştir.

Çizelge 4.9. Yaya navigasyonu problemi için başlangıç noktasının bulunmasına dair geliştirilen çoklu hipotez çalışmasına ait birinci ve ikinci uygulama ortamları için farklı başlangıç noktaları ve farklı örneklem mesafeleri dikkate alınarak gerçekleştirilen tüm denemelerden elde edilen sonuçlar

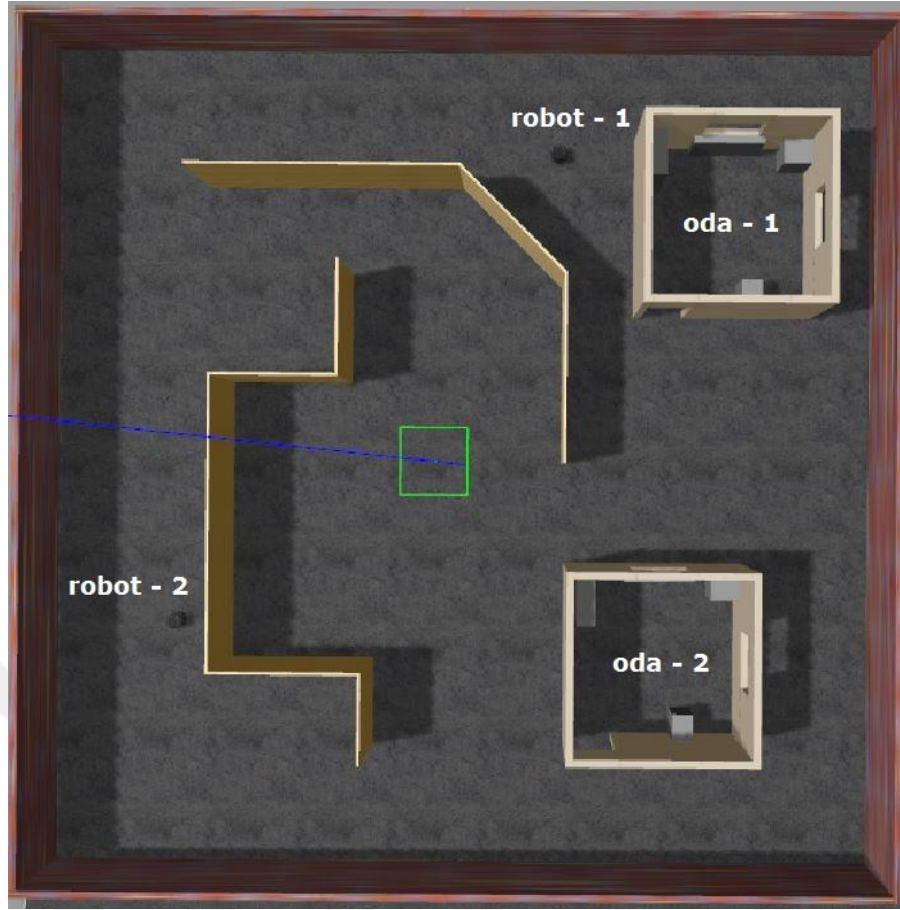
Test Ortamı	Denemeler	Gerçek pozisyon [x,y]	Kuantize Mesafesi	Başlangıç konumuna dair bulunan hipotez [x,y]	Toplam hipotez sayısı
Birinci deneme ortamı	1	1345-736	20 px	1345-736	117
			30 px	1335-736	79
			40 px	1325-736	60
			60 px	1365-736	40
	2	179-360	20 px	179-360	117
			30 px	179-350	79
			40 px	179-360	60
			60 px	179-380	40
	3	474-736	20 px	465-736	117
			30 px	465-736	79
			40 px	485-736	60
			60 px	465-736	40
İkinci deneme ortamı	1	386-1082	20 px	390-1077	131
			30 px	390-1077	89
			40 px	390-1077	66
			60 px	390-1077	45
	2	740-1082	20 px	746-1082	131
			30 px	745-1082	89
			40 px	752-1082	66
			60 px	745-1082	45

Robotlar veya farklı ajanlar üzerinde gereklenen konumlama, haritalama uygulamaları yayalar üzerinde sensör yapısının uygun olmamasından dolayı klasik algoritmalarla gereklenememektedir. Bu noktada bu alt başlıkta, yayalar üzerinde konumlama ve başlangı konumu bulunması üzerinde durulmuştur. Özellikle başlangı noktasının bilinmediđi durumların ele alınması ve konumlama sürecinde manyetik alan bilgisi olmaksızın yaya güzergah eğrisinin düzeltilmesi ile farklı bir bakış açısı sunulmuştur. Geliştirilen yöntem ilerleyen yıllarda yapılması planlanan haritalama ve çok katlı bina uygulamaları için de bir ön hazırlık olarak düşünölmektedir.



5. YENİ BİR ÇOKLU ROBOT HARİTA BİRLEŞTİRME YÖNTEMİNİN GELİŞTİRİLMESİ VE UYGULANMASI

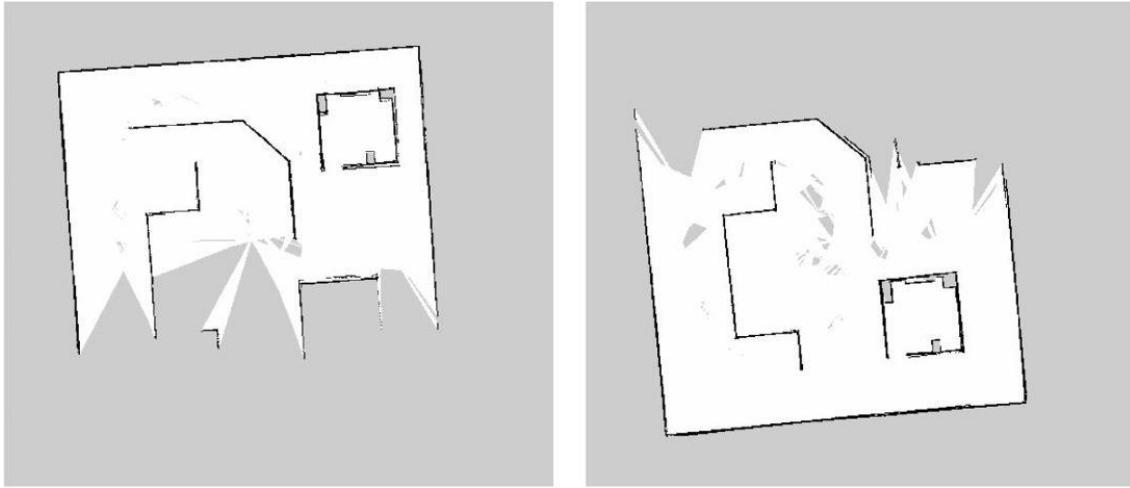
Robotlar farklı görevleri yerine getirerek insan hayatı için birçok kolaylık sağlamaktadır. Bu sayede gelecek yıllarda robotların gündelik yaşamımızda yer almaya başlaması kaçınılmaz bir olgudur. Tek bir robot veya bir robot takımı kullanılarak farklı görev atamaları ile robotlardan yararlanılabilir. Literatürdeki birçok çalışma tek bir robotun kullanıldığı uygulamaları baz almakta ve bu yönde oluşan sorunlara dair geliştirmeler yapılmaktadır. Bununla birlikte pek çok durumda robot takımlarına ihtiyaç duyulmaktadır. Örneğin, arama-kurtarma alanlarında tek bir robot yerine bir robot takımının yapacağı müdahale daha hızlı sonuçlar verecektir. Benzer şekilde endüstriyel uygulamaların birçoğunda tek bir robot yerine farklı veya benzer işleri yapan robot takımlarının kullanılması ile görevlerin daha etkin bir şekilde gerçekleştirilmesi sağlanabilir. Bu gibi durumlar göz önüne alındığında çoklu robot uygulamalarının sağladığı önemli üstünlükler bulunmaktadır (Topal ve ark., 2013). Ancak her mühendislik probleminde olduğu gibi bu tür üstünlüklerin kazanımı da fazladan bir maliyet oluşturmaktadır. Bu maliyet, robotların koordinasyonunun nasıl yapılacağı, lider odaklı veya eşit dağılımlı görev atamalarının yapılması, takım içerisindeki bir robotta karşılaşılabilecek olası bir problem ve bu durumda yapılması gereken adımlar gibi birçok konudan oluşmaktadır. Buna paralel şekilde daha da detaylı bir inceleme yapılırsa örneğin arama-kurtarma bölgesinde robot takımına verilen bir haritalama görevi için robotların oluşturduğu yerel haritaların nasıl birleştirileceği, yerel haritalar üzerinde çakışmayan bölgelerin bulunması veya kısmi haritalarda eşleşen bölgelerin algoritmaya bağlı küçük farklılıklar içermesi gibi problemlerde bulunmaktadır. Bütün bu problemler göz önüne alındığında çoklu robot uygulamaları üstünlüklerine rağmen çözülmesi gereken farklı problemlere de sebep olmaktadır. Bu düşünceyle tez kapsamında çoklu robotlara dair bir uygulama yapılmış ve çoklu robotların kullanılmasında karşılaşılan kısmi haritaların birleştirilmesi problemi üzerinde durulmuştur. Çalışma kapsamında benzetim ve gerçek zamanlı ortamlarda farklı iki deneme yapılmıştır. İlk olarak denemelerin gerçekleştirilmesi için Şekil 5.1'deki gibi bir ortam Gazebo platformunda tanımlanmıştır.



Şekil 5.1. Çoklu robot problemi ve yerel haritaların oluşturulması uygulamasına dair denemelerin yapıldığı birinci çalışma ortamı (uygulama-1, benzetim modeli denemesi)

Oluşturulan bu ortam içerisinde iki robot ve bunlara ilave olarak duvarların, koridorların ve birbirine benzer kısımların olduğu parçalar bulunmaktadır.

Çalışmanın temel amacı, homojen veya heterojen birden fazla robotun bir ortamda haritalama görevlerini yerine getirmesi ve global haritanın oluşturulmasıdır. Bu noktada ilk olarak, literatürde var olan metotların incelemesi yapılmıştır. İki robot içeren bu durum için ROS *gmapping* paketi kullanılarak robotlardan elde edilen meşguliyet haritaları Şekil 5.2'deki gibidir. Şekilde de görüleceği üzere robotlar, ortamla ilgili kısmi haritaları elde etmiştir.

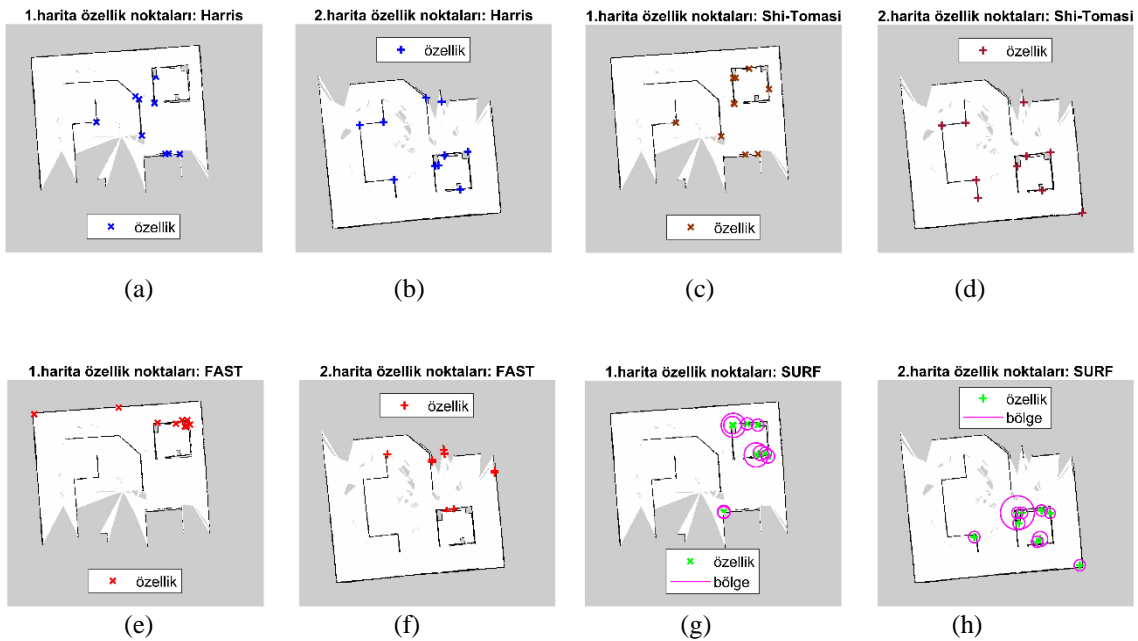


(a) 1. Robot yerel haritası

(b) 2. Robot yerel haritası

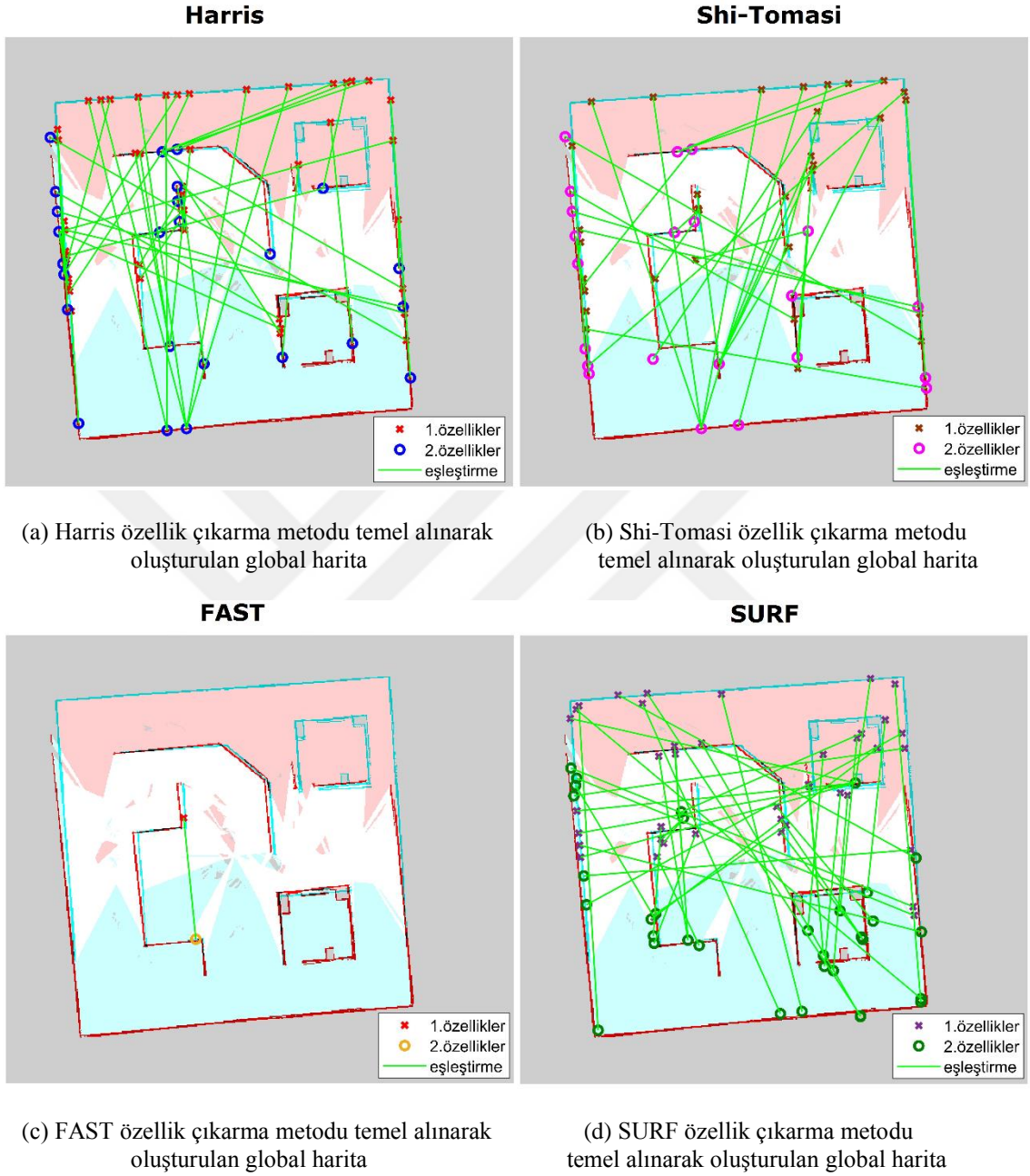
Şekil 5.2. Çoklu robot problemi ve yerel haritalardan global haritaların oluşturulması uygulamasına dair robotlardan elde edilen sonuçlar; (a) Birinci robotun oluşturduğu yerel harita, (b) İkinci robotun oluşturduğu yerel harita (uygulama – 1, benzetim modeli denemesi)

Literatürde bulunan yöntemler genel olarak özellik tabanlı görüntü işleme teknikleri üzerinden haritaların birleştirilmesi üzerinedir. Bu kapsamda Şekil 5.3'teki gibi kısmi haritalar ile ilgili özellikler elde edilmektedir. Elde edilen bu özellikler üzerinden kısmi haritaların birleştirilmesi amaçlanmaktadır. Özellik tanıma metodu olarak Harris, Shi-Tomasi, FAST ve SURF olmak üzere dört farklı metot kullanılmıştır.



Şekil 5.3. Benzetim ortamını esas alan birinci uygulama ile ilgili yerel haritalarda farklı yöntemlere göre bulunan özellikler; (a), (b) Harris (c), (d) Shi-Tomasi (e), (f) FAST (g), (h) SURF

Elde edilen özellikler üzerinden kısmi haritaların birleştirilmesine dair birleştirilmiş global harita Şekil 5.4’te görülmektedir.



Şekil 5.4. Çoklu robot çalışması benzetim ortamı uygulaması için robotların ürettiği yerel haritaların farklı özellik çıkartma yöntemleri girdisi ve Topal ve ark. (2013) çalışması esas alınarak birleştirilmesi sonucunda elde edilen global haritalar; (a) Harris, (b) Shi-Tomasi, (c) FAST, (d) SURF özelliklerinin kullanıldığı durumlar

Topal ve ark. (2013) çalışması incelendiğinde, yerel haritalarda bulunan ortak özellikler ve anahtar noktaların tanımı üzerinden global haritanın oluşturulması amaçlanmıştır. Ancak Şekil 5.4’te de görüleceği üzere literatürde yer alan özellik tabanlı

metotlar kullanılarak global haritanın üretilmesi, yönteme bağlı olmaksızın yanlış sonuçlar vermektedir. Bunun en önemli sebeplerinden birisi bahsedilen yöntemlerin ortamın asimetrik özelliklerden oluştuğunu düşünmesinden kaynaklanmaktadır. Böylece yerel haritalar birbirinden farklı gibi değil birbirinin aynısı gibi düşünülmekte buna bağlı olarak yanlış eşleştirmeler yapılmaktadır. Özellikle birbirinin aynısı olarak tasarlanmış iki odanın birbirinin aynısı gibi algılandığı göze çarpmaktadır. Bu yüzden çoklu robotların kullanımında karşılaşılan bu problemle ilgili yeni çözümlere ihtiyaç duyulmuştur. Topal ve ark. (2013) çalışmasını bir adım ileri götürecek bir çalışma tez kapsamında yapmıştır. Bu kapsamda yerel haritalarda bulunan özellik sayısından ziyade “güvenilir özelliklerin” bulunmasının gerekliliği görülmüştür. Buna göre güvenilir özellik tanımı yapılmış ve anlamlı bir harita birleştirme yöntemi geliştirilmiştir.

Tanım:

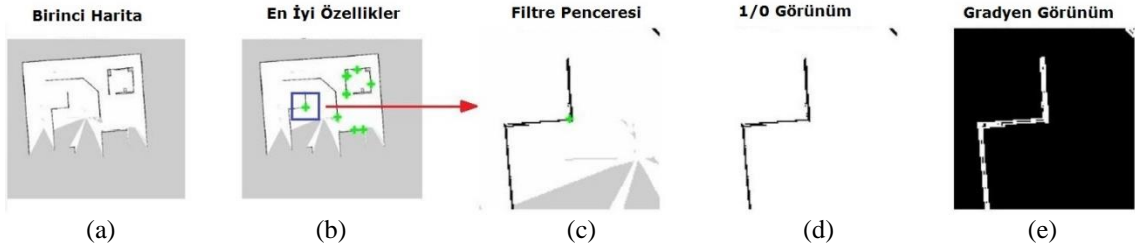
Güvenilir Özellik: N, M pozitif reel sayı ve $I, N \times M$ boyutunda bir harita fonksiyonu olsun. $I(x, y)$ boş, meşgul veya bilinmeyen hücrelerden oluşan bir haritayı temsil etmektedir. Bu hücreler eğer pozitifse engel olmayan, negatifse engelli veya sıfır değerine sahipse bilinmeyen bölgeler anlamına gelmektedir. P , bir özellik çıkarma metoduna göre elde edilmiş özellik kümesi ve p bu haritadaki (I) her bir özelliği temsil etmektedir (Eşitlik (5.1) ve Şekil 5.5b).

$$p = \{p(n, m) \mid p(n, m) \in P, n \leq N, m \leq M \wedge n, m \in \mathbb{Z}^+\} \forall P(u, v) \subseteq I_{N \times M}(x, y) \quad (5.1)$$

L ise özellik etrafında ve ilgili özelliğin güvenilir olup olmadığını belirlemek için oluşturulan bir alt penceredir (Eşitlik (5.2) ve Şekil 5.5c).

$$L = \{L_{f \times f} \mid L(x_c, y_c) = p(n, m) \wedge p(n, m) \in P\} \forall L_{f \times f} \subseteq I_{N \times M} \quad (5.2)$$

Burada x_c ve y_c , L boyutundaki bir pencere için merkez noktalarıdır ve filtre boyutu f 'dir.



Şekil 5.5. Özellik noktası, etrafında oluşturulan filtre ve varyasyonları; (a) Herhangi bir robottan alınan yerel harita, (b) Yerel harita üzerinde bulunan bazı özellik noktaları, (c) Seçilen özellikle ilgili f boyutunda seçilmiş L alt penceresi, (d) L alt penceresinin siyah-beyaz görünümü, (e) L penceresinin gradyeni alınarak oluşturulan görüntü

Buna göre Eşitlik (5.3)'te bahsedilen şartları sağlayan p özellik noktaları aynı zamanda t "güvenilir noktaları" olarak adlandırılabilir.

$$t \leftarrow \begin{cases} p \text{ güvenilir,} & \frac{\# L(x,y)|_{\text{bilinmeyen hücreler}}}{f} \leq f \\ p \text{ güvenilir değil,} & \frac{\# L(x,y)|_{\text{bilinmeyen hücreler}}}{f} > f \end{cases}, t \in P \wedge \forall L \subseteq I_{N \times M} \quad (5.3)$$

Bu denklem dikkatle incelendiğinde bir p özelliğinin güvenilirliği etrafındaki hücrelerin bilinip bilinmemesi ile ilgili olduğu görülmektedir. Bu, robotlar tarafından bu noktanın ne kadar sıklıkla görüldüğü ile ilgili bir test olarak düşünülebilir. Bu sayede haritada daha çok görülen bölgelerin bulunması amaçlanmaktadır.

Tez kapsamında geliştirilen güvenilir nokta belirlenmesine dair adımlar Algoritma 5.1'deki gibidir.

Güvenilir özelliğın belirlenmesinde önemli olgulardan birisi de filtre boyutu f 'nin değeridir. Eğer filtre boyutu gereğinden yüksek bir değerde seçilirse hesaplama zamanı artar. Buna paralel bir şekilde, bilinmeyen hücrelerin fazla olduğu durumlarda güvenilirlik testi sağlıklı bir şekilde yapılamayabilir. Bunun zıttına küçük seçilen filtre değerlerinde ise özelliğın etrafındaki bilinmeyen hücreler tam anlamıyla incelenemeyebilir. Bunun en önemli sebeplerinden birisi farklı robotlar tarafından elde edilen haritalarda aynı bölgelerin küçük farkla da olsa farklı temsillerinin olabilmesidir. Benzer şekilde diğer bir sebep ise kolon, duvar, büyük engeller gibi iç kısmına erişimin mümkün olmadığı kapalı bölgelerin iç kısımlarının harita çözünürlüğüne bağlı olarak tam temsil edilmemesidir. Bu yüzden filtre boyutunun seçilmesi önemli bir etken olup çalışma kapsamında ortam özelliklerinin dikkate alındığı bir formülasyon geliştirilmiştir. Bu kapsamda ortamda özellikler olarak düşünülebilen koridorların yaklaşık olarak

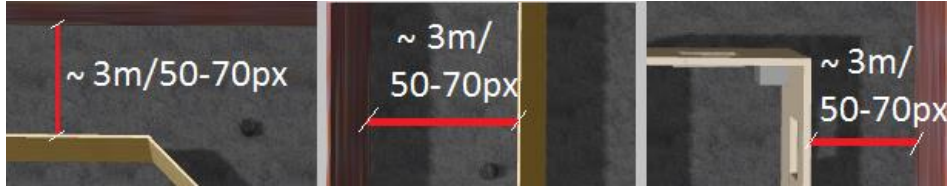
uzaklığının ortam özelliği olarak alınması fikri ortaya atılmıştır. Benzetim çalışması için oluşturulan ortama ait tipik bir koridor görüntüsü Şekil 5.6'daki gibidir.

Algoritma 5.1: Güvenilir Nokta Belirleme

```

Giriş :  $I$  //  $i$ . robot tarafından üretilen kısmi harita
Çıkış :  $T_i$  // her bir yerel harita  $I$  için güvenilir nokta kümesi
1: for bütün  $I$  yap
2:    $P_i \leftarrow$  fonksiyon ÖZELLİKÇIKART( $I$ ) // özellikleri belirle
3:   for bütün  $p_i \in P_i$  yap
4:      $T_i \leftarrow$  If (fonksiyon GÜVENİLİRNOKTABELİRLEYİCİ( $p_i$ )) //
                                                güvenilirlik testi yap
5:   end if
6: end for
7: fonksiyon GÜVENİLİRNOKTABELİRLEYİCİ ( $p_i$ )
8:    $L_i \leftarrow$   $I$  yi  $p_i$  çevresinde  $f$  filtre boyutunda böl
9:    $w_i \leftarrow$   $L_i$  için güvenilirlik ölçütüne göre hesapla
10:  If ( $w_i >$  eşikdeğer)  $\rightarrow$   $p_i$  güvenli nokta
11:  else if  $\rightarrow$   $p_i$  güvenli değil
12:  end if
13: end fonksiyon

```



Şekil 5.6. Çoklu robot çalışması benzetim ortamı uygulamasından elde edilen bazı koridor görüntüleri ve mesafeleri

Koridor görüntülerinin filtre boyutunu belirlemede özellik olarak düşünülmesi fikri ışığında Eşitlik (5.4)'te görülen bir formülasyon oluşturulmuştur.

$$f[px] = c_w [m] \times r[m/px]^{-1} + a[px] \quad (5.4)$$

Burada f filtre boyutu r harita çözünürlüğü, c_w en sık görülen koridor genişliği ve a güvenlik aralığı olarak tanımlanmıştır.

Yapılan benzetim çalışması için en sık görülen koridor genişliği 3 metre olarak tespit edilmiştir. Harita çözünürlüğünün 0.05 m/px olduğu göz önüne alınırsa bu değerler ışığında filtre boyutu 60 olarak bulunmuştur. α güvenlik aralığı değeri ise ± 10 px aralığında seçilmesinin deneysel çalışma sonuçlarına göre uygun olduğu görülmüştür. Buna göre nihai filtre değeri 50 – 70 aralığında olacak şekilde elde edilmiştir. Deneysel çalışmalarda bu değerlerin uygun olduğu ve birbirine yakın sonuçlar alınarak güvenilir nokta belirlenmesinin yapılabildiği görülmüştür.

Elde edilen bu bilgilere göre anlamlı harita birleştirilmesi yöntemi için Algoritma 5.2’de bahsedilen ve geliştirilen bu metot ile robotlardan elde edilen kısmi haritalar birleştirilmiş ve global harita oluşturulmuştur.

Algoritma 5.2: Anlamlı harita birleştirme

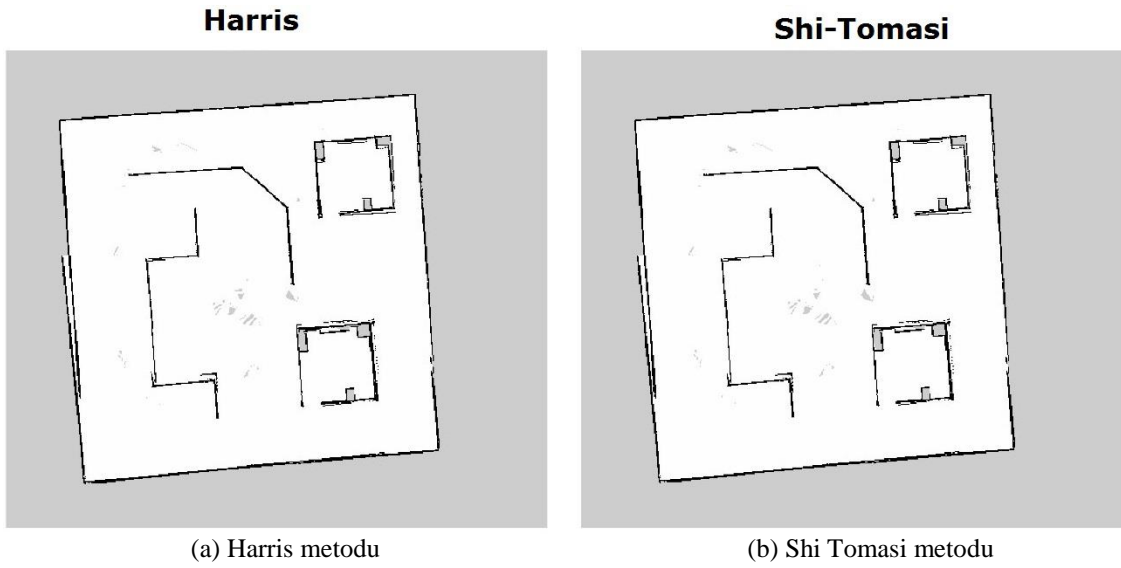
Giriş : I^i, T_j // Herhangi bir robot tarafından üretilen kısmi haritalar ve kısmi haritalara ait güvenilir nokta kümesi
Çıkış : F // Algoritma sonucu birleştirilen harita

- 1: **for** bütün I^i **yap**
- 2: **for** bütün t_j **yap** $\forall t_j \in T_j$
- 3: $L_{i,j} \leftarrow I^i$ ’yi t_j etrafında f boyutunda böl
- 4: $\nabla f_{i,j} \leftarrow \left(\frac{\partial L_{i,j}}{\partial x}, \frac{\partial L_{i,j}}{\partial y} \right)$ // gradyeni hesapla
- 5: **end for**
- 6: **end for**
- 7: **for** bütün I^i ’deki $i=1: \# \{ T_j \}$ **yap** // iki yerel harita için
- 8: $X = f_{1,i}$
- 9: **for** bütün I^i ’deki $j=1: \# \{ T_j \}$ **yap**
- 10: $Y = f_{2,j}$
- 11: $C(i,j) \leftarrow \frac{cov(X,Y)}{s_x s_y}$
- 12: **end for**
- 13: **end for**
- 14: $\underset{i,j}{\operatorname{argmax}} C = \{i,j \mid C(i,j) \geq C(k,l), \forall C(k,l) \in C\}$
- 15: $m_p \leftarrow C(i,j)$
- 16: **If**(m_p pozitifse) \rightarrow kısmi haritaları m_p güvenilir noktasına göre eşleştir
- 17: **elseif** \rightarrow Eşleşme yok
- 18: **end if**

Algoritma kısmi haritaları (I) ve her bir kısmi harita için güvenilir nokta kümesini (T) girdi olarak alır. Güvenilir noktaların etrafındaki alt pencelerin gradyen değerini elde

ederek bu değerlerinin kovaryans hesabını yapar ve pozitif olarak yüksek bir ilinti değerinin bulunduğu noktalara göre haritaları birleştirir.

Buna göre dört farklı metot için Algoritma 5.1 ve 5.2’de bahsedilen adımlar uygulanmış ve eşleşme durumları ile toplam süreler elde edilmiştir. Harris ve Shi-Tomasi yöntemleri ile elde edilen özelliklere göre haritalar eşleştirilebilmiş ve global harita bulunmuştur (Şekil 5.7). Diğer iki yöntemle herhangi bir güvenilir nokta bulunamadığı veya güvenilir noktalar arası ilinti bulunamadığı için eşleştirme yapılamamıştır (Çizelge 5.1).



Şekil 5.7. Çoklu robot çalışması benzetim ortamı uygulaması için farklı özelliklerin girdi olarak sunulması ve geliştirilen anlamlı harita birleştirilmesi metodu ile yerel haritaların birleştirilmesi durumuna dair elde edilen nihai global haritalar; (a) Harris özelliklerinin girdi olarak verildiği duruma dair global harita, (b) Shi-Tomasi özelliklerinin girdi olarak verildiği duruma dair global harita (uygulama – 1, benzetim modeli denemesi)

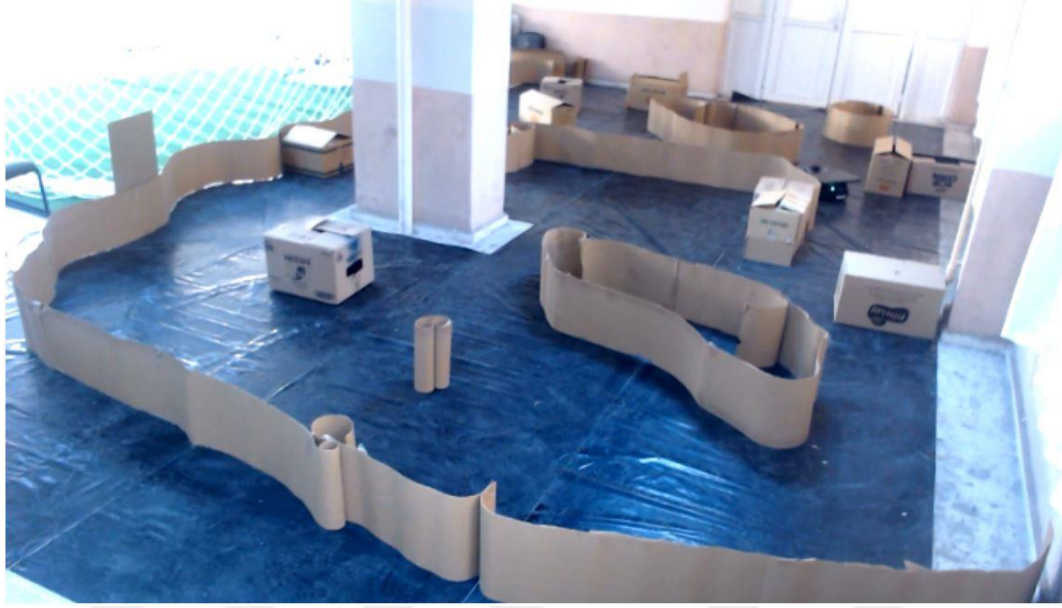
Çizelge 5.1. Çoklu robot çalışması benzetim ortamı uygulaması için global haritaların bulunmasına dair geçen süre ve birleştirme durumu (uygulama – 1, benzetim modeli)

Algoritma	Birleştirme Zamanı (s)	Birleştirme Durumu
Harris	1.713244	Evet
Shi-Tomasi	1.712669	Evet
FAST	1.418407	Hayır
SURF	1.544664	Hayır

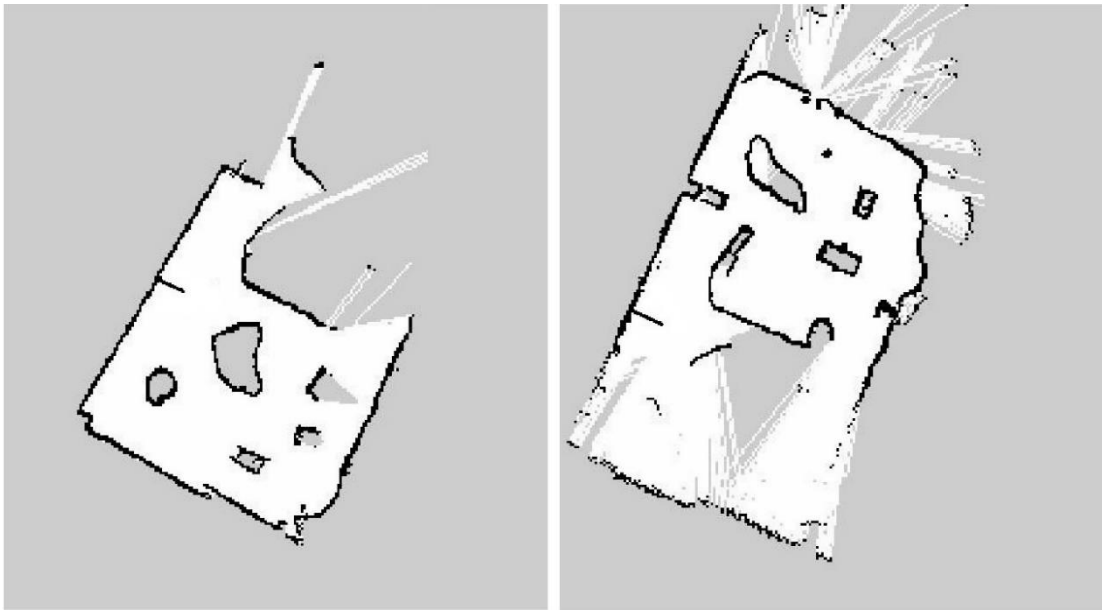
Aynı ortamda tek bir robot ile haritalama işlemi 105 saniyede yapılırken geliştirilen metot sayesinde 65 saniye gibi bir sürede ortam haritalaması yapılabilmektedir. Bunlara ilave olarak haritalama süresinin düşürülmesinin nasıl bir öneme sahip olduğunu

vurgulamak için herhangi bir robota görev ataması yapılması ile ilgili testler de yapılmıştır.

Benzetim çalışması sonuçlarını doğrulamak adına ikinci bir çalışma gerçek zamanlı bir ortamda yapılmıştır. İlk olarak Şekil 5.8’de görülen bir ortam için farklı iki kısmi harita elde edilmiştir (Şekil 5.9).



Şekil 5.8. Çoklu robot problemi ve yerel haritaların oluşturulması uygulamasına dair denemelerin yapıldığı ikinci çalışma ortamı (uygulama-2, gerçek zamanlı ortam denemesi)

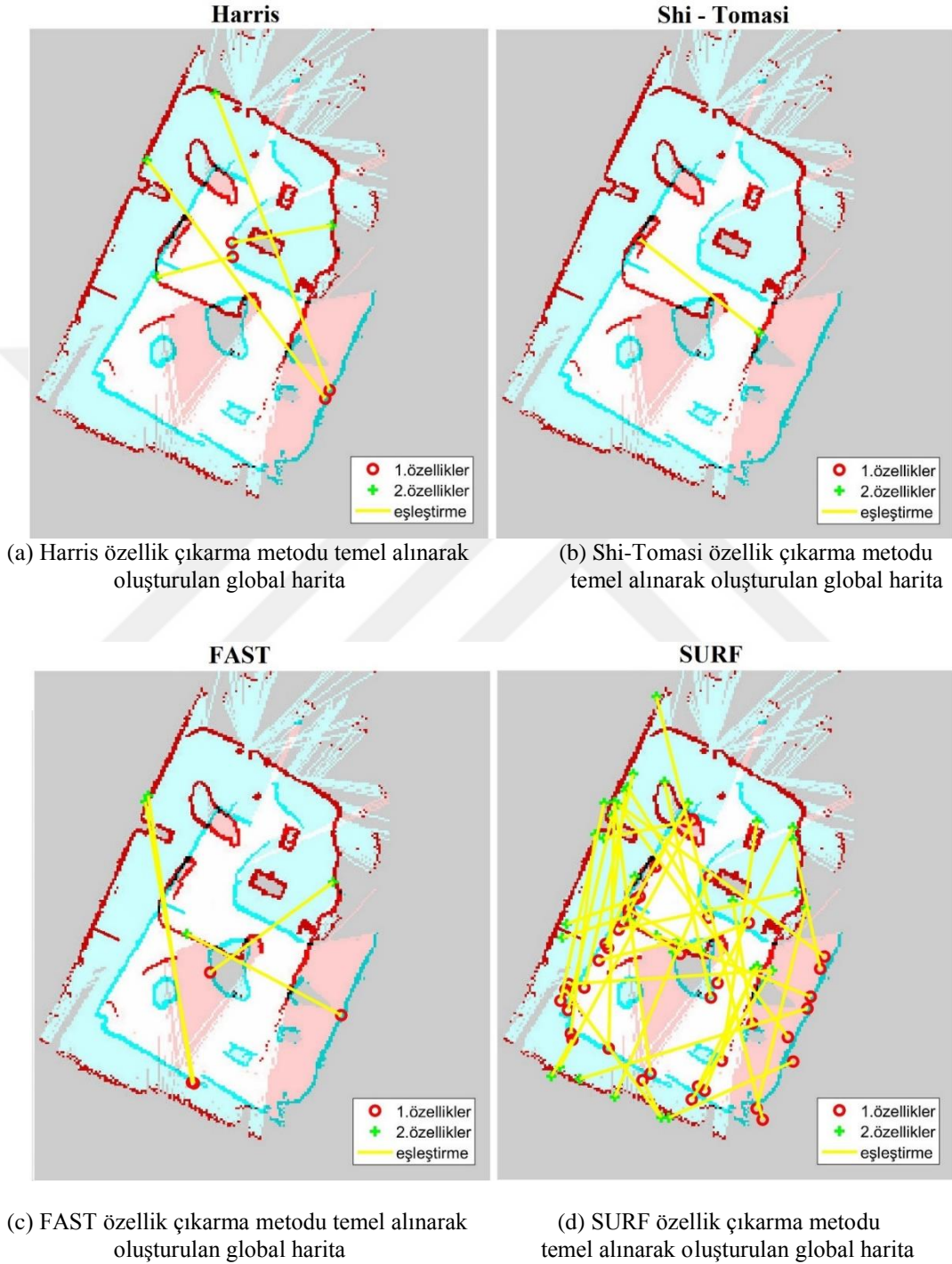


(a) 1. Robot yerel haritası

(b) 2. Robot yerel haritası

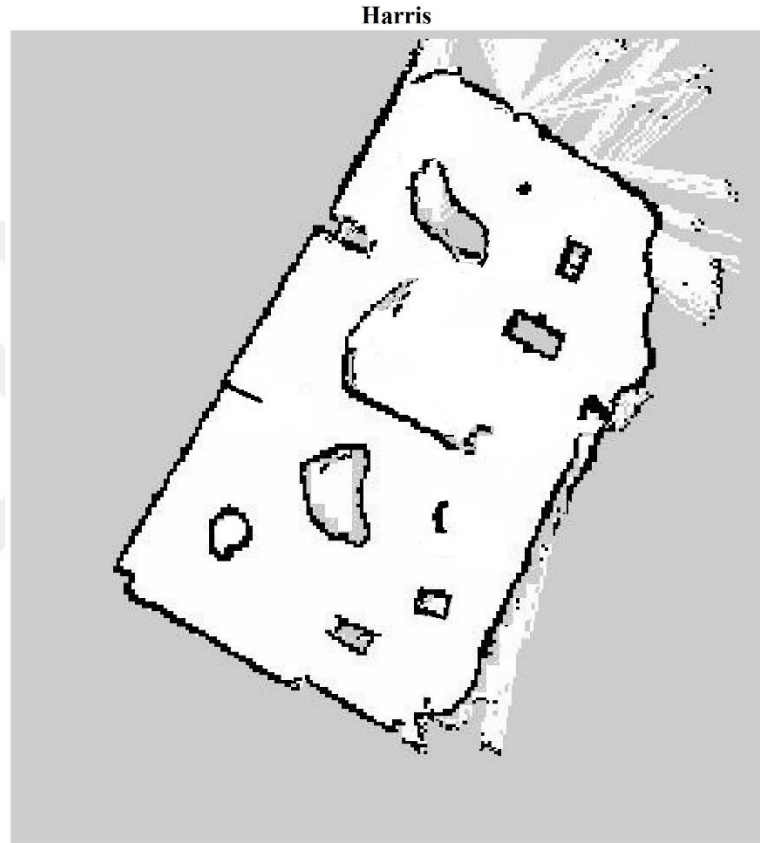
Şekil 5.9. Çoklu robot problemi ve yerel haritalardan global haritaların oluşturulması uygulamasına dair robotlardan elde edilen sonuçlar; (a) Birinci robotun oluşturduğu yerel harita, (b) İkinci robotun oluşturduğu yerel harita (uygulama – 2, gerçek zamanlı ortam denemesi)

Daha sonra elde edilen kısmi haritalardan özelliklerin çıkarılması ve buna göre anahtar özellikler üzerinden birleştirilme yapılması hedeflenmiştir (Şekil 5.10).



Şekil 5.10. Çoklu robot çalışması gerçek zamanlı ortam uygulaması için robotların ürettiği yerel haritaların farklı özellik çıkartma yöntemleri girdisi ve Topal ve ark. (2013) çalışması esas alınarak birleştirilmesi sonucunda elde edilen global haritalar; (a) Harris, (b) Shi-Tomasi, (c) FAST, (d) SURF özelliklerinin kullanıldığı durumlar

Ancak şekilden de görüleceği üzere hiçbir özellik çıkartma ve buna bağlı eşleştirme yöntemi ile global harita elde edilememiştir. Benzetim modelinde ortaya çıkan farklı yerleri aynı yer gibi değerlendirme problemi gerçek zamanlı denemelerde de ortaya çıkmıştır. Bu bilgiler ışığında tez kapsamında geliştirilen güvenilir özellik tanımlama ve anlamlı harita birleştirme algoritmaları yerel haritalara uygulanmış ve ortam global haritası Şekil 5.11’deki gibi elde edilmiştir.



Şekil 5.11. Çoklu robot çalışması gerçek zamanlı ortam uygulaması için Harris özelliklerinin girdi olarak sunulması ve geliştirilen anlamlı harita birleştirilme metodu ile yerel haritaların birleştirilmesi durumuna dair elde edilen nihai global harita (uygulama – 2, gerçek zamanlı ortam denemesi)

Benzetim modelindeki çalışmaya göre gerçek zamanlı çalışmada sadece Harris modeli başarılı sonuçlar vermiştir (Çizelge 5.2). Bunun farklı nedenleri olmakla birlikte ilgili özellik tanıma yönteminin ızgara tabanlı haritaya uygun olmaması veya lokal haritalardaki nüansların sebep olduğu ortak özellik noktalarının bulunamaması başlıca sebepleridir.

Çizelge 5.2. Çoklu robot çalışması gerçek zamanlı ortam uygulaması için global haritaların bulunmasına dair geçen süre ve birleştirme durumu (uygulama – 2, gerçek zamanlı ortam denemesi)

Algoritma	Birleştirme Zamanı	Birleştirme Durumu
Harris	2.610178	Evet
Shi-Tomasi	2.393639	Hayır
FAST	2.438914	Hayır
SURF	2.306784	Hayır

Bununla birlikte algoritmaların global haritayı elde etme süresi değerlendirildiğinde benzetim modelindeki yapılan çalışmaya paralel sonuçlar elde edilmiş ve birbirleri arasında önemli bir fark görülmemiştir. Çoklu modelin avantajınının görülmesi adına robotlara bir noktadan diğer noktaya gitme görevleri verilmiş ve zaman değerleri kaydedilmiştir. Buna göre tek robot ortam haritasını 260 saniyede, çoklu robotlar ise 140 saniyede elde ederken, robot tekli durum için bir noktadan başka bir noktaya gitme görevini toplamda 284 saniyede yaparken çoklu robot durumu için 164 saniyede gerçekleşmiştir. Bölüm 2’de de bahsedildiği üzere çoklu robotların kullanıldığı durumlarda klasik tek robotlu durumlara göre daha farklı problemlerinde ele alınması gerekmektedir. Bu alt başlıkta incelenen yöntem, çoklu robotlarda karşılaşılan harita birleştirme durumunu simetrik ortamlar için ele alarak literatürde değinilmeyen bir konu üzerinde durmuştur.

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

Bu tez çalışması SLAM algoritmaları üzerine yapılan incelemeleri ve buna bağlı olarak eksik görülen noktalardaki gelişmeleri incelemektedir. SLAM problemi mobil robotlar için önemli bir basamaktır. SLAM sayesinde mobil bir robotun kendi kendine bilinmeyen ortamlarda gezinimi mümkün olmaktadır. Bu sebeple algoritma çözümlerinin günümüzde hayatımızın her alanına etki eden/edecek sonuçları ortaya çıkmıştır. Bölüm 1 ve 2’de de bahsedildiği üzere sağlıktan araç teknolojisine kadar birçok alanda farklı teknolojik gelişmeler SLAM sayesinde elde edilebilmiştir. SLAM problemi ile ilgili tarihçe ve kaynak araştırması yapılmış ve buna göre literatürde eksik noktalar belirtilmiştir. Bu eksik noktalarla ilgili farklı çalışmalar yapılarak sonuçlar elde edilmiştir.

SLAM problemi genel itibariyle klasik ve modern olmak üzere iki farklı dönemde sınıflandırılmıştır. Klasik dönem algoritmaları temelinde Bayes filtreleri ve bunun tahmin/güncelleme adımlarından yararlanmaktadır. Bu düşünceyle ilk olarak, literatürde sıklıkla karşılaşılan EKF ve UKF algoritmaları ile ilgili bir çalışma yapılmış ve sonuçları sunulmuştur. Buna ilave olarak klasik dönem SLAM çalışmalarında kullanılan farklı sensörlerin etkileri üzerine bir araştırma yapılmıştır. Son yıllarda birçok alanda farklı amaçlarla kullanılan RGB-D kameraların 2B lazer sensör gibi kullanılması durumu incelenmiştir. Bunun için gerçek zamanlı bir çalışmada 2B veri verebilen bir lazer sensörle RGB-D sensörler karşılaştırılmış ve elde edilen haritalar incelenerek ilgili sonuçlar analiz edilmiştir. Modern dönem SLAM yöntemleri olarak görülen kameraların ve görüntü işleme tekniklerinin kullanıldığı vSLAM son yıllarda araştırmacılar tarafından yoğun bir şekilde çalışılmaktadır. Bu düşünceyle vSLAM teknikleri üzerine de incelemeler yapılmış ve ilk olarak, var olan iki yöntemin gerçek zamanlı bir uygulamada karşılaştırmalı analizi yapılmıştır. İkinci olarak son yıllarda özellik tabanlı yöntemler içerisinde en iyi algoritmalarından birisi olarak görülen ORB-SLAM yöntemi ele alınmıştır. Bu yöntem hızlı çalışması açısından özellik tabanlı vSLAM algoritmaları için önemli bir gelişme sağlamıştır. Ancak elde edilen haritalar özellik tabanlıdır. Buna bir çözüm olarak REMODE olarak bilinen başka bir yöntemle ORB-SLAM algoritması birleştirilmiş ve gerçek zamanlı bir ortamın yoğunluk tabanlı derinlik haritaları çıkarılmıştır. Buna dair algoritma basamakları Bölüm 4’te detaylıca açıklanmıştır.

Bölüm 3'te de bahsedildiği üzere SLAM bir üst küme olacak şekilde bünyesinde farklı çözümlenmeleri de barındırmaktadır. Aktif SLAM ve harita birleştirme konuları bunların en bilinenlerindedir. Aktif SLAM mobil bir robota tam otonomi kazandıran bir çözümlenmedir. Bu konu dahilinde robot, harita ve konum bilgilerine ilave olarak nereye gitmesi gerektiğine de karar vermektedir. Bu kapsamda farklı iki çalışma yapılmıştır. İlkinde, küçük ölçekli fabrika benzeri bir ortam tasarlanarak zeminde bulunan iş çizgilerinin takibi amaçlanmıştır. Salt bir çizgi takibi kararsız sonuçlar üreteceği için sola yönelimli ismiyle bir algoritma geliştirilmiş ve robotun izleyeceği yola algoritma çerçevesinde karar vermesi beklenmiştir. Robot, ortamda otonom hareket ederken aynı zamanda konum ve harita bilgilerine de erişmiştir. Çalışma sonucunda kullanıcı ve geliştirilen algoritma ile yapılan yönlendirmeye bağlı olarak elde edilen haritalar karşılaştırılmış ve geliştirilen yöntemin etkinliği görülmüştür. Bir diğer çalışmada ise son yıllarda popüler olan CNN tabanlı bir metot kullanılmıştır. Gerçek zamanlı olarak ortam görüntüleri ve robotun teker manevra değerleri CNN mimarisine eğitim amaçlı olarak verilmiştir. Eğitilen ağı farklı bir test ortamında denenmesi amaçlanmıştır. Robot elde ettiği ağ mimarisi sayesinde kullanıcı girdisi olmaksızın otonom bir şekilde dönüş, ileri gitme gibi aksiyonları gerçeklemiştir. Bu hareket mekanizması ile SLAM yapısı birleştirilmiş ve aktif SLAM algoritmaları için farklı bir çözümlenme sunulmuştur.

Bir diğer SLAM alt alanı ise haritaların birleştirilmesi problemidir. Tez kapsamında çoklu robot çalışmaları veya bir robottan elde edilen yerel haritaların birleştirilmesi problemi ele alınmıştır. Literatürdeki yöntemlerin birbirine benzemeyen lokal haritaları birleştirmede başarılı iken simetrik kısımların olduğu kısmi haritalarda kararsız sonuçlar ürettiği görülmüştür. Bu kapsamda güvenilir özellik ismiyle yeni bir tanımlama yapılmış ve bu tanım temel alınarak anlamlı harita birleştirme yöntemi geliştirilip kısmi haritaların birleştirilmesi amaçlanmıştır. İlgili yöntemle dair tanımlamalar, adımlar ve sonuçlar Bölüm 5 içerisinde detaylı bir şekilde ele alınmıştır.

Bununla birlikte robotların dışında yayalar üzerinde de bir konumlama çalışması yapılmıştır. Çalışmada başlangıç noktası bilinmeyen durumlar için örnekleme fikri temeline dayanan bir yaklaşım geliştirilmiş ve başlangıç nokta bulma problemi çözülmüştür. Yinelemesiz Bayes tabanlı ve harita eşleştirme yöntemleri ile yaya takip eğrisi düzeltilerek manyetik alan bilgisine ihtiyaç duymadan IMU ile yaya konumlaması yapılmıştır.

Bütün bunların yanı sıra elde edilen haritaların nasıl kullanılacağı ile ilgili bir çalışma yapılmıştır. Robotlardan elde ettiği haritalara göre kendilerine verilecek görevleri

yerine getirmeleri beklenmektedir. Bununla ilgili temel problem, robotun bulunduğu noktadan hedef noktasına nasıl gideceği üzerinedir. Bu kapsamda beş farklı optimal yol planlama algoritması denenmiştir. Algoritmalar yol uzunluğu ve yolun bulunması için gereken zaman açısından incelenerek karşılaştırmalı bir analiz yapılmış ve sonuçlar istatistiki açıdan değerlendirilmiştir. Ayrıca geliştirilen pratik bir metotla optimal mesafe/zaman ölçütü tanımlanmış ve sonuçların istatistiki metotlarla tutarlı olduğu görülmüştür.

6.2. Öneriler

Klasik SLAM ile ilgili problem tanımı tam manasıyla yapılmış ve birçok problem çözülmüş olmasına karşın hala birçok çalışma dinamik çevre değişkeni ve çoklu robot durumlarını göz ardı etmektedir. Bu konuda yapılacak geliştirmeler klasik SLAM algoritmalarını bir adım öteye taşıyacaktır. Öte yandan vSLAM algoritmaları henüz yolun başı olarak tarif edilebilecek bir noktada olup araştırmacılar tarafından yoğun bir şekilde incelenmektedir.

Bir diğer araştırma konusu ise mikro araçlar ve bunlarla yapılacak olan SLAM çözümleridir. Bu türde araçların kullanımı günden güne yaygınlaşmaktadır. Ancak bu araçların kontrolü ile ilgili yaşanan problemler SLAM çözümlerini de etkilemektedir. Bu yüzden gürbüz bir SLAM algoritmasının tanımlanmasına ihtiyaç duyulmaktadır.

Son yıllarda üzerinde durulan derin öğrenme ve varyasyonlarının SLAM çözümlerine entegre edilmesi de farklı gelişmelere yol açacaktır.

Benzer şekilde harita birleştirme konusu genelde meşguliyet haritaları için ele alınmaktadır. Bunun diğer harita türlerine uygulanması konusunda yapılmış pek az çalışma vardır. Bununla birlikte anlamlı haritaların elde edilmesi başlı başına önemli ve ucu açık bir konu olarak ele alınabilir. Her ne kadar SLAM'den elde edilen haritalar doğruluğu yüksek çözümler içerse de haritaların anlamlandırılması gelecek SLAM çalışmalarına yön verecektir. Bu konuda da derin ağlardan yararlanılabileceği düşünülmektedir.

SLAM haritaları 2B veya 3B olarak elde edilmektedir ve özellikle 2B haritalarda optimal yol tanımı kolaylıkla yapılabilen ve entegrasyon sağlanmaktadır. Ancak 3B haritalarla ilgili bu konularda yapılan çalışmalar nadirdir. Bu konudaki yapılacak çalışmaların ufuk açıcı olacağı aşikardır.

SLAM çözümlerinin navigasyon ve keşif algoritmaları ile birleştirilmesi de hala ucu açık birçok problemi barındırmaktadır. Özellikle vSLAM algoritmaları ile keşif/navigasyon algoritmalarının birleştirilmesi önemli sorunların çözümüne sebep olacaktır.

Bütün bunlara ilave olarak zaman iyileştirme problemi SLAM ve vSLAM algoritmaları için her zaman önemli bir gelişme olacaktır. Benzer şekilde çoklu robot durumları için birçok konu ele alınmayı beklemektedir. Ayrıca akademik anlamda birçok yönüyle SLAM konusu ele alınsa da endüstriyel bazdaki çalışmalar daha başlangıç aşamasında olup bu alanda katedilecek önemli mesafeler vardır.



KAYNAKLAR

- Achtelik, M. W., Lynen, S., Weiss, S., Kneip, L., Chli, M. ve Siegwart, R., 2012, Visual-inertial SLAM for a small helicopter in large outdoor environments, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2651-2652.
- Ali, S. S., Hammad, A. ve Tag Eldien, A. S., 2018, FastSLAM 2.0 tracking and mapping as a Cloud Robotics service, *Computers & Electrical Engineering*, 69, 412-421.
- Angermann, M. ve Robertson, P., 2012, FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors; Hitchhiking on Human Perception and Cognition, *Proceedings of the IEEE*, 100 (Special Centennial Issue), 1840-1848.
- Aulinas, J., Petillot, Y., Salvi, J., Llad, X. ve #243, 2008, The SLAM problem: a survey. Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, IOS Press: 363-371.
- Bailey, T., 2002, Mobile robot localisation and mapping in extensive outdoor environments, *University of Sydney*.
- Bailey, T. ve Durrant-Whyte, H., 2006, Simultaneous localization and mapping (SLAM): Part II, *IEEE Robotics & Automation Magazine*, 13 (3), 108-117.
- Bailey, T., Nieto, J., Guivant, J., Stevens, M. ve Nebot, E., 2006, Consistency of the EKF-SLAM algorithm, *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3562-3568.
- Bruno, L. ve Robertson, P., 2011, WiSLAM: Improving FootSLAM with WiFi, *2011 International Conference on Indoor Positioning and Indoor Navigation*, 1-10.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. ve Leonard, J. J., 2016, Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age, *IEEE Transactions on Robotics*, 32 (6), 1309-1332.
- Carlone, L., Du, J., Kaouk Ng, M., Bona, B. ve Indri, M., 2013, Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence, *Journal of Intelligent & Robotic Systems*, 75 (2), 291-311.
- Carpin, S., Birk, A. ve Jucikas, V., 2005, On map merging, *Robotics and Autonomous Systems*, 53 (1), 1-14.
- Carpin, S., 2008, Fast and accurate map merging for multi-robot systems, *Autonomous Robots*, 25 (3), 305-316.
- Carrillo, H., Reid, I. ve Castellanos, J. A., 2012, On the comparison of uncertainty criteria for active SLAM, *2012 IEEE International Conference on Robotics and Automation*, 2080-2087.
- Chandra, K. P. B., Da-Wei, G. ve Postlethwaite, I., 2010, SLAM using EKF, EH^∞ and mixed EH_2/H^∞ filter, *2010 IEEE International Symposium on Intelligent Control*, 818-823.
- Chekhlov, D., Pupilli, M., Mayol-Cuevas, W. ve Calway, A., 2006, Real-time and robust monocular SLAM using predictive multi-resolution descriptors, *International symposium on visual computing*, 276-285.
- Cheong, J. Y. ve Park, I. K., 2017, Deep CNN-Based Super-Resolution Using External and Internal Examples, *IEEE Signal Processing Letters*, 24 (8), 1252-1256.

- Chien, S.-Y., Wang, H. ve Lewis, M., 2010, Human vs. Algorithmic Path Planning for Search and Rescue by Robot Teams, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 54 (4), 379-383.
- Csorba, M., 1997, Simultaneous localisation and map building, *University of Oxford Oxford*.
- Davison, 2003, Real-time simultaneous localisation and mapping with a single camera, *Proceedings Ninth IEEE International Conference on Computer Vision*, 1403-1410 vol.1402.
- Davison, A. J., Reid, I. D., Molton, N. D. ve Stasse, O., 2007, MonoSLAM: real-time single camera SLAM, *IEEE Trans Pattern Anal Mach Intell*, 29 (6), 1052-1067.
- Dine, A., Elouardi, A., Vincke, B. ve Bouaziz, S., 2015, Graph-based SLAM embedded implementation on low-cost architectures: A practical approach, *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 4612-4619.
- Dissanayake, G., Huang, S., Wang, Z. ve Ranasinghe, R., 2011, A review of recent developments in Simultaneous Localization and Mapping, *2011 6th International Conference on Industrial and Information Systems*, 477-482.
- Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F. ve Csorba, M., 2001, A solution to the simultaneous localization and map building (SLAM) problem, *IEEE Transactions on Robotics and Automation*, 17 (3), 229-241.
- Durdu, A. ve Korkmaz, M., 2019, Autonomously Simultaneous Localization and Mapping Based on Line Tracking in a Factory-like Environment, *Advances in Electrical and Electronic Engineering*, 17 (1), 45-53.
- Durrant-Whyte, H. ve Bailey, T., 2006, Simultaneous localization and mapping: Part I, *IEEE Robotics & Automation Magazine*, 13 (2), 99-108.
- Durrantwhyte, H. F., 1988, Uncertain Geometry in Robotics, *IEEE Journal of Robotics and Automation*, 4 (1), 23-31.
- Elibol, A., Kim, J., Gracias, N. ve Garcia, R., 2014, Efficient image mosaicing for multi-robot visual underwater mapping, *Pattern Recognition Letters*, 46, 20-26.
- Engel, J., Schöps, T. ve Cremers, D., 2014, LSD-SLAM: Large-Scale Direct Monocular SLAM, Cham, 834-849.
- Erden, A., 2003, TÜBİTAK Bilim ve Teknik, Robotik-1.
- Erinc, G. ve Carpin, S., 2013, Anytime merging of appearance-based maps, *Autonomous Robots*, 36 (3), 241-256.
- Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D. ve Stewart, B., 2006, Distributed Multirobot Exploration and Mapping, *Proceedings of the IEEE*, 94 (7), 1325-1339.
- Fu, C., Olivares-Mendez, M. A., Suarez-Fernandez, R. ve Campoy, P., 2013, Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers, *Journal of Intelligent & Robotic Systems*, 73 (1-4), 513-533.
- Fuentes-Pacheco, J., Ruiz-Ascencio, J. ve Rendon-Mancha, J. M., 2015, Visual simultaneous localization and mapping: a survey, *Artificial Intelligence Review*, 43 (1), 55-81.
- Garcia Puyol, M., Bobkov, D., Robertson, P. ve Jost, T., 2014, Pedestrian Simultaneous Localization and Mapping in Multistory Buildings Using Inertial Sensors, *IEEE Transactions on Intelligent Transportation Systems*, 15 (4), 1714-1727.
- Gentner, C. ve Ulmschneider, M., 2017, Simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope, *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1-8.

- Grasa, O. G., Bernal, E., Casado, S., Gil, I. ve Montiel, J. M., 2014, Visual SLAM for Handheld Monocular Endoscope, *IEEE Trans Med Imaging*, 33 (1), 135-146.
- Grisetti, G., Kummerle, R., Stachniss, C. ve Burgard, W., 2010, A Tutorial on Graph-Based SLAM, *IEEE Intelligent Transportation Systems Magazine*, 2 (4), 31-43.
- Guivant, J. E. ve Nebot, E. M., 2001, Optimization of the simultaneous localization and map-building algorithm for real-time implementation, *IEEE Transactions on Robotics and Automation*, 17 (3), 242-257.
- Havangi, R., Taghirad, H. D., Nekoui, M. A. ve Teshnehlab, M., 2014, A Square Root Unscented FastSLAM With Improved Proposal Distribution and Resampling, *IEEE Transactions on Industrial Electronics*, 61 (5), 2334-2345.
- Henry, P., Krainin, M., Herbst, E., Ren, X. ve Fox, D., 2014, RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments, In: *Experimental Robotics*, Eds, p. 477-491.
- Holmes, S., Klein, G. ve Murray, D. W., 2008, A Square Root Unscented Kalman Filter for visual monoSLAM, *2008 IEEE International Conference on Robotics and Automation*, 3710-3716.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. ve Burgard, W., 2013, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots*, 34 (3), 189-206.
- Huang, W. H. ve Beevers, K. R., 2005, Topological map merging, *The International Journal of Robotics Research*, 24 (8), 601-613.
- Khairuddin, A. R., Talib, M. S. ve Haron, H., 2015, Review on simultaneous localization and mapping (SLAM), *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 85-90.
- Kim, C., Sakthivel, R. ve Chung, W. K., 2008, Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem, *IEEE Transactions on Robotics*, 24 (4), 808-820.
- Klein, G. ve Murray, D., 2007, Parallel Tracking and Mapping for Small AR Workspaces, *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 225-234.
- Klein, G. ve Murray, D., 2009, Parallel Tracking and Mapping on a camera phone, *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, 83-86.
- Konolige, K., Fox, D., Limketkai, B., Ko, J. ve Stewart, B., 2003, Map merging for distributed robot navigation, *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 212-217 vol.211.
- Korkmaz, M. ve Durdu, A., 2017, Real-Time Implementation of ORB-Based Probabilistic 3D Mapping, *ELECTRONICS WORLD*, 123 (1975), 38-40.
- Korkmaz, M. ve Durdu, A., 2018, Comparison of optimal path planning algorithms, *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 255-258.
- Korkmaz, M., Durdu, A. ve Tusun, Y. E., 2018, Sensor Comparison for a Real-Time SLAM Application, *International Journal of Information and Electronics Engineering*, 8 (1), 1-4.
- Korkmaz, M., Demircan, O. ve Durdu, A., 2019, A Real-Time Study And Comparison On VSLAM Techniques. 1st International Symposium On Implementations Of Digital Industry And Management Of Digital Transformation 2019. Konya, TURKEY: 180-189.

- Koroglu, M. T. ve Yilmaz, A., 2017, Pedestrian inertial navigation with building floor plans for indoor environments via non-recursive Bayesian filtering, *2017 IEEE SENSORS*, 1-3.
- Kurt-Yavuz, Z. ve Yavuz, S., 2012, A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms, *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, 37-43.
- Labbé, M. ve Michaud, F., 2019, RTAB - Map as an open - source lidar and visual simultaneous localization and mapping library for large - scale and long - term online operation, *Journal of Field Robotics*, 36 (2), 416-446.
- Li, S. ve Ni, P., 2010, Square-root unscented Kalman filter based simultaneous localization and mapping, *The 2010 IEEE International Conference on Information and Automation*, 2384-2388.
- Lv, Q., Lin, H., Wang, G., Wei, H. ve Wang, Y., 2017, ORB-SLAM-based tracing and 3D reconstruction for robot using Kinect 2.0, *2017 29th Chinese Control And Decision Conference (CCDC)*, 3319-3324.
- Makarenko, A. A., Williams, S. B., Bourgault, F. ve Durrant-Whyte, H. F., 2002, An experiment in integrated exploration, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 534-539 vol.531.
- Martinez-Cantin, R. ve Castellanos, J. A., 2005, Unscented SLAM for large-scale outdoor environments, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3427-3432.
- Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K. ve De Schutter, J., 2003, A Comparison of Decision Making Criteria and Optimization Methods for Active Robotic Sensing, Berlin, Heidelberg, 316-324.
- Montemerlo, M., Thrun, S., Koller, D. ve Wegbreit, B., 2002, FastSLAM: A factored solution to the simultaneous localization and mapping problem, *In Proceedings of the AAAI National Conference on Artificial Intelligence*, 593-598.
- Mu, B., Giamou, M., Paull, L., Agha-mohammadi, A., Leonard, J. ve How, J., 2016, Information-based Active SLAM via topological feature graphs, *2016 IEEE 55th Conference on Decision and Control (CDC)*, 5583-5590.
- Mur-Artal, R., Montiel, J. M. M. ve Tardos, J. D., 2015, ORB-SLAM: A Versatile and Accurate Monocular SLAM System, *IEEE Transactions on Robotics*, 31 (5), 1147-1163.
- Mur-Artal, R. ve Tardos, J. D., 2017a, ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, *IEEE Transactions on Robotics*, 33 (5), 1255-1262.
- Mur-Artal, R. ve Tardos, J. D., 2017b, Visual-Inertial Monocular SLAM With Map Reuse, *IEEE Robotics and Automation Letters*, 2 (2), 796-803.
- Newcombe, R. A., Lovegrove, S. J. ve Davison, A. J., 2011, DTAM: Dense tracking and mapping in real-time, *2011 International Conference on Computer Vision*, 2320-2327.
- Nieto, J., Bailey, T. ve Nebot, E., 2006, Scan-SLAM: Combining EKF-SLAM and scan correlation, *Field and service robotics*, 167-178.
- Nourbakhsh, I. R. ve Siegwart, R., 2004, Introduction to autonomous mobile robots, *The MIT Press, Cambridge, Massachusetts, England, ISBN 0, 262 (19502)*, 142-150.
- Nüchter, A., 2009, 3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom, Springer-Verlag Berlin Heidelberg, p.
- Oriolo, G., Paolillo, A., Rosa, L. ve Vendittelli, M., 2015, Humanoid odometric localization integrating kinematic, inertial and visual information, *Autonomous Robots*, 40 (5), 867-879.

- Paz, L. M., Tardos, J. D. ve Neira, J., 2008, Divide and Conquer: EKF SLAM in $\mathcal{O}(n)$, *IEEE Transactions on Robotics*, 24 (5), 1107-1120.
- Pizzoli, M., Forster, C. ve Scaramuzza, D., 2014, REMODE: Probabilistic, monocular dense reconstruction in real time, *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2609-2616.
- Polvi, J., Taketomi, T., Yamamoto, G., Dey, A., Sandor, C. ve Kato, H., 2016, SlidAR: A 3D positioning method for SLAM-based handheld augmented reality, *Computers & Graphics*, 55, 33-43.
- Quigley, M., Gerkey, B. ve Smart, W. D., 2015, Programming Robots with ROS: a practical introduction to the Robot Operating System, " O'Reilly Media, Inc.", p.
- Ran, L., Zhang, Y., Zhang, Q. ve Yang, T., 2017, Convolutional Neural Network-Based Robot Navigation Using Uncalibrated Spherical Images, *Sensors*, 17 (6), 1341.
- Saeedi, S., Trentini, M., Seto, M. ve Li, H., 2016, Multiple-Robot Simultaneous Localization and Mapping: A Review, *Journal of Field Robotics*, 33 (1), 3-46.
- Saian, P. O. N., Suyoto ve Pranowo, 2016, Optimized A-Star algorithm in hexagon-based environment using parallel bidirectional search, *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1-5.
- Sheng, J., Tano, S. ve Jia, S., 2011, Mobile robot localization and map building based on laser ranging and PTAM, *2011 IEEE International Conference on Mechatronics and Automation*, 1015-1020.
- Smith, P., Reid, I. D. ve Davison, A. J., 2006, Real-time monocular SLAM with straight lines, 17-26.
- Smith, R., Self, M. ve Cheeseman, P., 1987, Estimating uncertain spatial relationships in robotics, *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 850-850.
- Smith, R. C. ve Cheeseman, P., 1986, On the Representation and Estimation of Spatial Uncertainty, *International Journal of Robotics Research*, 5 (4), 56-68.
- Sola, J., 2007, Multi-camera VSLAM: from former information losses to self-calibration, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA*.
- Sonmez, A., Kocyigit, E. ve Kugu, E., 2015, Optimal path planning for UAVs using Genetic Algorithm, *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 50-55.
- Stachniss, C., Hahnel, D. ve Burgard, W., 2004, Exploration with active loop-closing for FastSLAM, *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 1505-1510 vol.1502.
- Stachniss, C., 2009, Robotic Mapping and Exploration, Springer-Verlag Berlin Heidelberg, p.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W. ve Cremers, D., 2012, A benchmark for the evaluation of RGB-D SLAM systems, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 573-580.
- Sudhakara, P., Ganapathy, V. ve Sundaran, K., 2017, Probabilistic roadmaps-spline based trajectory planning for wheeled mobile robot, *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 3579-3583.
- Sun, Q., Li, M., Wang, T. ve Zhao, C., 2018, UAV path planning based on improved rapidly-exploring random tree, *2018 Chinese Control And Decision Conference (CCDC)*, 6420-6424.

- Taketomi, T., Uchiyama, H. ve Ikeda, S., 2017, Visual SLAM algorithms: a survey from 2010 to 2016, *IPSJ Transactions on Computer Vision and Applications*, 9 (1).
- Thrun, S., Burgard, W. ve Fox, D., 1998, A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learning*, 31 (1-3), 29-53.
- Thrun, S., Burgard, W. ve Fox, D., 2005, Probabilistic robotics, MIT Press, p.
- Thrun, S. ve Leonard, J. J., 2008, Simultaneous localization and mapping, *Springer handbook of robotics*, 871-889.
- Topal, S., Erkmen, I. ve Erkmen, A. M., 2013, A novel multirobot map fusion strategy for occupancy grid maps, *Turkish Journal of Electrical Engineering and Computer Sciences*, 21 (1), 107-119.
- Tuna, G., Gulez, K., Gungor, V. C. ve Mumcu, T. V., 2012, Evaluations of different Simultaneous Localization and Mapping (SLAM) algorithms, *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2693-2698.
- Wang, J. ve Cho, H., 2008, Micropeg and Hole Alignment Using Image Moments Based Visual Servoing Method, *IEEE Transactions on Industrial Electronics*, 55 (3), 1286-1294.
- Weingarten, J. ve Siegwart, R., 2005, EKF-based 3D SLAM for structured environment reconstruction, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3834-3839.
- Yamauchi, B., 1997, A frontier-based approach for autonomous exploration, *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 146-151.
- Zamora, E. ve Yu, W., 2013, Recent advances on simultaneous localization and mapping for mobile robots, *Iete Technical Review*, 30 (6), 490-496.
- Zhou, X. S. ve Roumeliotis, S. I., 2006, Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case, *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1785-1792.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mehmet KORKMAZ
Uyruğu : T.C.
Doğum Yeri ve Tarihi : KONYA - 19 / 08 / 1987
Telefon : +90 382 288 3668
Faks : +90 382 288 3525
E-Posta : mkorkmazeem@gmail.com

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: Cemil Keleşoğlu Lisesi / Selçuklu / Konya			2004
Üniversite	: SÜ. Elektrik-Elektronik Mühendisliği / Selçuklu / Konya			2009
Yüksek Lisans	: SÜ. Elektrik-Elektronik Mühendisliği / Selçuklu / Konya			2013
Doktora	: KTÜN Elektrik-Elektronik Mühendisliği / Selçuklu / Konya			2019

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2018-2019	The Ohio State University	Ziyaretçi Araştırmacı
2011-	Selçuk Üniversitesi	Araştırma Görevlisi
2010-	Aksaray Üniversitesi	Araştırma Görevlisi
2010-2010	RMA Müşavirlik-Mühendislik	Elektrik Mühendisi
2009-2010	CMS CNC Makine	Elektrik Mühendisi

UZMANLIK ALANI

Kesirli dereceden denetleyiciler, SLAM, Robotik, ROS

YABANCI DİLLER

İngilizce (YDS: 83.75; YÖKDİL: 90)

YAYINLAR

Aydogdu O, Korkmaz M, editors. A simple approach to design of variable parameter nonlinear pid controller. International Conference on Electrical Engineering and Applications; 2011. (*Yüksek Lisans*)

Aydogdu O, Korkmaz M. Optimal Design of a Variable Coefficient Fractional Order PID Controller by using Heuristic Optimization Algorithms. International Journal of Advanced Computer Science and Applications. 2019;10(3):314-21. (*Yüksek Lisans*)

Durdu A, Korkmaz M. Autonomously Simultaneous Localization and Mapping Based on Line Tracking in a Factory-like Environment. *Advances in Electrical and Electronic Engineering*. 2019;17(1):45-53. (*Doktora*)

Erdoğan K, Korkmaz M, Durdu A, Yılmaz N, Topal S,. Formation Morphing of Multi-Robots Using Graph Theory: Fugitive Chasing. *Proceedings of the World Congress on Engineering and Computer Science*; 2015. (*Diğer*)

Kasgoz A, Korkmaz M, Alanalp MB, Durmus A. Effect of processing method on microstructure, electrical conductivity and electromagnetic wave interference (EMI) shielding performance of carbon nanofiber filled thermoplastic polyurethane composites. *Journal of Polymer Research*. 2017;24(9):148. (*Diğer*)

Korkmaz M. Fizyoterapist Kontrollü, Supinasyon ve Pronasyon El Bilek Hareketleri İçin Fizik Tedavi ve Rehabilitasyon Cihazı. 2017. (*Diğer*)

Korkmaz M, Aydogdu O, editors. Fractional Order Controller Design for Ball and Beam System. *Applied Mechanics and Materials*; 2013: Trans Tech Publ. (*Yüksek Lisans*)

Korkmaz M, Aydogdu O. Performance Comparison of Fractional Controllers and Artificial Intelligence Techniques. *International Journal of Machine Learning and Computing*. 2013;3(2):229. (*Yüksek Lisans*)

Korkmaz M, Aydogdu O. Designing Fractional Order Pid Controllers With Artificial Intelligent Optimization Techniques. *Electronics World*. 2017;123(1969):40-3. (*Yüksek Lisans*)

Korkmaz M, Aydoğdu Ö. Değişken Parametrelili Kesirli PID Tasarımı. (*Yüksek Lisans*)

Korkmaz M, Aydoğdu Ö. Kesir Dereceli bir PID Denetleyicinin Genetik Algoritma Optimizasyonlu ANFIS Modeli. Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı. 2012:11-3. (*Yüksek Lisans*)

Korkmaz M, Aydoğdu Ö, Doğan H, editors. Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller. 2012 International Symposium on Innovations in Intelligent Systems and Applications; 2012: IEEE. (*Yüksek Lisans*)

Korkmaz M, Durdu A. Real-Time Implementation Of Orb Based Probabilistic 3d Mapping. *Electronics World*. 2017;123(1975):38-40. (*Doktora*)

Korkmaz M, Durdu A, editors. Comparison of optimal path planning algorithms. 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET); 2018: IEEE. (*Doktora*)

Korkmaz M, Yılmaz N. Face Recognition by Using Back Propagation Artificial Neural Network and Windowing Method. Journal of Image and Graphics. 2016;4(1):15-9.(Diğer)

Korkmaz M, Yılmaz N, Durdu A, editors. Comparison of the SLAM algorithms: Hangar experiments. MATEC Web of Conferences; 2016: EDP Sciences. (Doktora)

Korkmaz M, Durdu A, Tusun YE. Sensor Comparison for a Real-Time SLAM Application. International Journal of Information and Electronics Engineering. 2018;8(1):1-4. (Doktora)

PROJELER

TÜBİTAK 1005 - Yeni Fikirler ve Ürünler, Araştırmacı/Uzman, 214M562, Ses Ve Elektromanyetik Dalgaları Perdeleyerek Kapalı Ortam Dinlemelerini Engellereyebilen Çok Katmanlı Polimer Kompozitlerin Geliştirilmesi, Sonuçlandı, K:15/04/2015, A: 15/10/2016, Türkiye

T.C. Bilim, Sanayi ve Teknoloji Bakanlığı – Teknogirişim, Yürütücü, 2015 Fizyoterapist Kontrollü, Supination ve Pronation El Bilek Hareketleri İçin Fizik Tedavi ve Rehabilitasyon Cihazı, 08/06/2015 – 19/10/2016, Türkiye

BURSLAR

TÜBİTAK 2214-A Yurt Dışı Araştırma Burs Programı (Doktora öğrencileri için), 1059B141601279, 27/04/2018 – 23/04/2019, The Ohio State University, ABD (1 yıl)

TÜBİTAK 2211-C Yurt İçi Öncelikli Alanlar Doktora Burs Programı, 1649B031603019, 2016/1, Türkiye (1 yıl)

PATENTLER

Fizyoterapist Kontrollü Pronasyon ve Supinasyon Fizik Tedavi Cihazı, TR 2016/11675 B, Belge Tarihi: 2018/09/21