



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



OTONOM MOBİL ROBOTLARDA GÖRSEL
SLAM UYGULAMALARI

Mehmet Fatih KOÇ

YÜKSEK LİSANS TEZİ

Mekatronik Mühendisliği Anabilim Dalı

MAYIS-2024
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Mehmet Fatih KOÇ tarafından hazırlanan “Otonom Mobil Robotlarda Görsel SLAM Uygulamaları” adlı tez çalışması .../.../... tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Hasan Erdiñ KOÇER

Danışman

Doç. Dr. Barış GÖKÇE

Üye

Doç. Dr. Abdülsamed TABAK

İmza

.....

.....

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../20.. gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Şerife Yurdağül KUMCU
FBE Müdürü

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 Fatih KOÇ

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

Mobil Robotlarda Görsel SLAM Uygulamaları

Mehmet Fatih KOÇ

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Barış GÖKÇE

2024, 57 Sayfa

Jüri

Doç. Dr. Barış GÖKÇE
Prof. Dr. Hasan Erdiç KOÇER
Doç. Dr. Abdülsamed TABAK

Mobil robotlar geçtiğimiz on yıllarda ciddi ilerlemeler kaydetmiş olup, üretimden lojistiğe, havacılıktan askeri alanlara, hizmet sektöründen evlere kadar geniş bir yelpazede kullanım alanı yaygınlaşmıştır. Mobil robotların bilmediği bir ortamda verilen görevleri yapabilmesi için pozisyon tahmini ve aynı anda harita çıkarabilmesi, SLAM teknikleri kullanılarak yapılmaktadır. Bu kapsamda kullanılan geleneksel SLAM yöntemlerinin yerini kameraların kullanılması fikri ile görsel SLAM tekniği kullanılmaya başlanmıştır.

Görsel SLAM tekniğinde düşük maliyetli sensörler, yoğun çevresel bilgilerin olması ayrı bir avantaj sağlamaktadır. Son yıllarda araştırmacılar bu alana yoğun ilgi duymaktadır ve bu alanda çeşitli algoritmalar ve teknikler üzerinde geliştirmeler yapılmıştır. Geliştirilen yöntemler ile ciddi başarılar elde edilmiş olmasına rağmen hala çözülmesi gereken sorunlar bulunmaktadır. Bu sorunlar arasında Görsel SLAM algoritmalarının performansını etkileyen birçok dış kaynaklı parametreler bulunmaktadır. Ortamın ışıklandırılması, çevredeki nesnelerin dokusu ve yapısal özellikler, sistemin hızı ve ani sert manevralar, kalabalık ve dinamik ortamlar gibi birçok dış kaynaklı parametreler bulunmaktadır.

Bu tez çalışmasında V-SLAM performansının artırılması adına, ortam ışıklandırılması üzerinde durulmuş ve düşük ışıklandırılmalı ortamlar için görüntü iyileştirme algoritmaları ile performans artırılması çalışması yapılmıştır. Literatürde iyi sonuçlar elde edilmiş ORB-SLAM3 algoritması ile tasarlanan bir platform üzerinden toplanan görüntüler histogram eşitleme ve gama düzeltmesi gibi görüntü iyileştirme algoritmalarına verilerek SLAM performansları ve verimlilikleri karşılaştırılmıştır. Bu kapsamda kullanılan algoritma ve kamera sensörlerin performansları, mutlak pozisyon hatası ve göreceli pozisyon hatası gibi hata metrikleri ile değerlendirilmiştir. Sistem sonuçları görüntü üzerinde yapılan iyileştirmelerin daha kararlı ve iyi sonuçlar verdiği gözlemlenmiştir. İlerleyen dönemde V-SLAM için derin öğrenme algoritmalarının entegre edilmesi ve semantik tabanlı yaklaşımlara yönelik çalışmalar yapılması planlanmaktadır.

Anahtar Kelimeler: Görsel SLAM, Lokalizasyon, Mobil Robotlar, Navigasyon, Haritalama.

ABSTRACT

MS THESIS

Visual SLAM Applications in Autonomous Mobile Robots

Mehmet Fatih KOÇ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE / DOCTOR OF PHILOSOPHY
IN MECHANICAL ENGINEERING**

Advisor: Assoc. Prof. Dr. Barış GÖKÇE

2024, 57 Pages

Jury

**Assoc. Prof. Dr. Barış GÖKÇE
Prof. Dr. Hasan Erdiñ KOÇER
Assoc. Prof. Dr. Abdülsamed TABAK**

Mobile robots have made significant progress in the past decades, and their use has become widespread in a wide range of areas, from production to logistics, from aviation to military areas, from the service sector to homes. In order for mobile robots to perform tasks in an unfamiliar environment, position estimation and mapping at the same time are done using SLAM techniques. In this context, visual SLAM technique has started to be used with the idea of using cameras instead of traditional SLAM methods.

In the visual SLAM technique, low-cost sensors and intense environmental information provide a distinct advantage. In recent years, researchers have shown great interest in this field, and developments have been made on various algorithms and techniques in this field. Although serious successes have been achieved with the developed methods, there are still problems that need to be solved. Among these problems, there are many external parameters that affect the performance of Visual SLAM algorithms. There are many external parameters such as the lighting of the environment, the texture and structural features of the surrounding objects, the speed of the system and sudden harsh maneuvers, crowded and dynamic environments.

In this thesis study, in order to increase V-SLAM performance, ambient lighting was emphasized and a study was carried out to increase performance with image enhancement algorithms for low-light environments. Images collected on a platform designed with the ORB-SLAM3 algorithm, which has good results in the literature, were given to image improvement algorithms such as histogram equalization and gamma correction, and SLAM performances and efficiencies were compared. In this context, the performances of the algorithms and camera sensors used were evaluated with error metrics such as absolute position error and relative position error. It has been observed that the improvements made on the system results give more stable and better results. In the future, it is planned to integrate deep learning algorithms for V-SLAM and work on semantic-based approaches.

Keywords: Visual SLAM, Localization, Mobile Robots, Navigation, Mapping.

ÖNSÖZ

Tez çalışmamda her türlü desteklerini sağlayan kıymetli danışmanım Doç. Dr. Barış GÖKÇE'ye ve Arş. Gör. Mümine YILDIZ'a;

Eğitim hayatı boyunca sabır, anlayış ve destekleriyle yanımda olan anneme, babama ve kardeşlerime;

Teşekkür ederim.

Mehmet Fatih KOÇ
KONYA-2024



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	iv
1. GİRİŞ	1
1.1. Tarihçe	1
1.2. Tez Organizasyonu	2
2. KAYNAK ARAŞTIRMASI	3
2.1. SLAM Çalışmaları.....	3
2.2. Görsel SLAM Çalışmaları.....	3
2.1.1. Stereo / Monoküler vSLAM	3
2.1.2. Visual-Inertial SLAM	5
2.1.3. RGB-D SLAM.....	5
3. MATERYAL VE YÖNTEM.....	8
3.1. SLAM.....	8
3.1.1. Lokalizasyon	8
3.1.2. Haritalama.....	8
3.2. Görsel SLAM.....	12
3.2.1. Özellik tabanlı.....	12
3.2.2. Doğrudan yöntem	12
3.3. Kamera Modelleme ve Kalibrasyon.....	13
3.3.1. Monoküler kamera	14
3.3.2. Stereo kameralar	14
3.3.3. RGB-D kameralar	14
3.3.4. Kamera kalibrasyonu	14
3.4. Özellik Çıkarımı ve Eşleme.....	16
3.4.1. Harris köşe detektörü	16
3.4.2. Shi-Tomasi köşe dedektörü	17
3.4.3. FAST corners	17
3.4.4. SIFT-SURF features	18
3.4.5. BRIEF descriptor	18

3.4.6. ORB	19
3.5. RANSAC	19
3.6. Döngü Kapama	20
3.7. Görsel Odometre	21
3.8. Yazılım ve Donanım Bileşenleri.....	27
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	31
4.1. Donanım ve Yazılım Bileşenleri.....	31
4.2. Görüntü Ön İşleme	32
4.3. Kamera Kalibrasyonu	33
4.4. Çalışma Platformu	37
4.5. Deney ve Gözlemler	38
5. SONUÇLAR VE ÖNERİLER	51
6. KAYNAKLAR	53

ŞEKİLLER DİZİNİ

Şekil 2.1. RGB-D SLAM işleyişi (Yousif vd., 2015).....	6
Şekil 3.1 (a) Metrik Harita (b) Topolojik Harita (Anderson, 2019)(Sundram vd., 2018)	9
Şekil 3.2 SLAM'ın işleyişi(Durrant-Whyte ve Bailey, 2006).....	9
Şekil 3.3 Özellik ve Doğrudan tabanlı yöntem (Macario Barros, 2022).....	13
Şekil 3.4 Kamera çeşitleri (Lu vd., 2018).....	13
Şekil 3.5 Kalibrasyon için içsel ve dışsal parametreler	15
Şekil 3.6 Kalibrasyon için kullanılan dama şekli	15
Şekil 3.7. Bozunma şekiller	16
Şekil 3.8 Harris köşe dedektörü algılama yapısı (Collins, 2007)	17
Şekil 3.9 FAST köşe algılayıcısı (Rosten ve Drummond, 2006)	18
Şekil 3.10 ORB özellik tespiti (Rublee vd.,2011)	19
Şekil 3.11 RANSAC algoritması (Toshihiko Watanabe, 2013).....	20
Şekil 3.12 ICP algoritması (Molero vd., 2011)	21
Şekil 3.13 Görsel odometri temsili (a) yerel 3 boyutlu harita ve (b) tespit edilen 2 boyutlu görüntü özellikleri. Her bir 3B harita noktası ile tespit edilen 2B özellik arasındaki uyumu bulmaktır. (Aladem ve Rawashdeh, 2018).....	22
Şekil 3.14 Görüntüden hareket çıkarımı aşamaları	23
Şekil 3.15 Epipolar geometri gösterimi (Fraundorfer ve Scaramuzza, 2011).....	24
Şekil 3.16 Üçgenleme metodu (Gao ve Zhang, 2021)	26
Şekil 3.17 PR1 Robot(Wyrobek, 2008) ve PR2 Robot (Benier vd., 2016).....	28
Şekil 3.18 ROS çalışma yapısı	29
Şekil 3.19 ROS iletişimi	29
Şekil 4.1 Mobil platform	31
Şekil 4.2 RealSense d435i derinlik kamerası	32
Şekil 4.3 Görüntü ön işleme sistem adımları (a) ham görüntü (b) görüntü iyileştirme uygulanan görüntü (c) iyileştirme uygulanmış görüntü üzerinden ORB-SLAM3 özellik çıkarımı (d) ORB-SLAM3 yörünge çıktısı.....	33
Şekil 4.4 Stereo kamera kalibrasyon işlemi.....	34
Şekil 4.5 AprilGrid	35
Şekil 4.6 AprilGrid ile kalibrasyon.....	36
Şekil 4.7 Realsense-Viewer kalibrasyon paneli	37
Şekil 4.8 Farklı ortamlardan toplanan görüntüler.....	37
Şekil 4.9 Düşük aydınlatmalı ortamlar	38
Şekil 4.10 Işık patlamalarının olduğu ortamlar	38
Şekil 4.11 Histogram eşitleme sonucu görüntü çıktıları.....	40
Şekil 4.12 Görüntü iyileştirme uygulandıktan sonraki ORB özellikleri (a) orijinal, (b) histogram eşitleme, (c) kontrast sınırlamalı uyarlamalı histogram eşitleme, (d) gama düzeltme.....	41
Şekil 4.13 Mühendislik fakültesi zemin kat (a) ORB-SLAM3 3B nokta bulutu haritası ve (b) 2B Cartographer haritası	42
Şekil 4.14 Mühendislik fakültesi kütüphane yörünge çıktıları.....	43
Şekil 4.15 Mühendislik fakültesi kütüphane yörünge çıktıları.....	44
Şekil 4.16 Mühendislik fakültesi zemin kat yörünge çıktıları.....	45
Şekil 4.17 Euroc MH_04_difficult verisi yörüngeleri (a) monoküler kamera (b) stereo kamera.....	46
Şekil 4.18 Histogram eşitleme ile çıkarılan mono-SLAM APE sonuçları.....	47
Şekil 4.19 Histogram eşitleme ile çıkarılan mono-SLAM yörüngesi	47

Şekil 4.20 Fakülte C-Blok kat 4 SLAM performansı (a) ham veri, (b) HE uygulanmış, (c)kontrast kısıtlamalı HE uygulanmış veriler	47
Şekil 4.21 Fakülte C-Blok kat 4 lux değerleri	48
Şekil 4.22 Fakülte kat 2 SLAM performansı (a) ham veri, (b) HE uygulanmış, (c) kontrast kısıtlamalı HE uygulanmış veriler	48
Şekil 4.23 Fakülte C-Blok kat 4 lux değerleri	48
Şekil 4.24 ORB-SLAM özellik çıkarımı (a) ham veri, (b) HE, (c) kontrast kısıtlamalı uyarlanabilir HE.....	49
Şekil 4.25 Karanlık ortamda ORB-SLAM özellik çıkarımı (a) ham veri, (b) HE, (c) kontrast kısıtlamalı uyarlanabilir HE	50



SİMGELER VE KISALTMALAR

Simgeler

m	: Yer işaretleri.
m_i	: i 'nci dönüm noktasının konumunu tanımlayan vektör.
u_k	: Kontrol vektörü.
U_k	: Kontrol girdi geçmişleri.
x_k	: Robotun konum ve yönelimini veren durum vektörü.
z_{ik}	: k zamanında i 'nci dönüm noktasının konumuna ilişkin bir gözlem.
X_k	: Robot konumlarının geçmişi.
$Z_{0:k}$: Dönüm noktası gözlemlerinin kümesi.

Kısaltmalar

2D	: İki boyutlu
3D	: Üç boyutlu
AMR	: Otonom Mobil Robot(Autonomous Mobile Robot)
AGV	: Otonom Yönlendirmeli Araç(Automated Guided Vehicle)
BOW	: Kelime Çantası (Bag Of Word)
BRIEF	: Binary Robust Independent Elementary Features
EKF	: Genişletilmiş Kalman Filtresi (Extended Kalman Filter)
FAST	: Features from Accelerated Segment Test
FPS	: Saniyedeki Kare Sayısı (Frame Per Second)
GPS	: Küresel Konumlandırma Sistemi
HE	: Histogram Eşitleme
ICP	: Yinelemeli En Yakın Nokta
IMU	: Ataletsel Ölçüm Birimi
KF	: Kalman Filtresi
KKUHE	: Kontrast Kısıtlamalı Uyarlanabilir Histogram Eşitleme
LiPo	: Lityum Polimer Batarya
LIDAR	: Işık Algılama ve Mesafe Tayini (Light Detection and Ranging)
ORB	: Oriented FAST and Rotated BRIEF
PF	: Parçacık Filtresi
RANSAC	: Rastgele örnekleme konsensüsü
ROS	: Robotik İşletim Sistemi
SLAM	: Eşzamanlı Yerelleştirme ve Haritalama
SIFT	: Scale Invariant Feature Transform
SURF	: Speeded-Up Robust Features
V-SLAM	: Görsel Eş Zamanlı Yerelleştirme ve Haritalama (Visual SLAM)

1. GİRİŞ

Geçtiğimiz on yılda mobil robotik sistemler alanında ciddi mesafeler katedildiği gözlemlenmiştir. Özellikle imalat, askeriye, lojistik, tarım ve tıp gibi alanlarda verimliliği arttırmak ve sistemleri otonom hale getirmek için önemli çalışmalar yapılmıştır. Otonom robotlar endüstride tekrarlanan görevler, riskli ortamlar, hız ve verimlilik gerektiren işler için önemli bir araç haline gelmiştir. Konvansiyonel üretim tekniklerindeki sabit ve belirli işleri yapan makinelerin yerini, işbirlikçi, kendi kendine karar verebilen ve esneklik sağlayabilen mobil robotik sistemler almıştır. Özellikle yapay zekanın ve çevre algılama sistemlerinin gelişimi ile robotlar sürü halinde hareket ederek işleri oldukça verimli ve hızlı yapabilme kabiliyeti kazanmışlardır.

1.1. Tarihçe

Mobil robotların ilk gelişimi 1940'lı yıllara dayanmaktadır. William Gray tarafından tasarlanan Elmer ve Elsie adlı robotlar yavaş hareketleri ve şekilleri itibariyle kaplumbağa robot olarak anıldılar. 1950'li yıllara gelindiğinde dünyanın ilk endüstriyel robotu olarak tanımlanan Unimate George Devol tarafından icat edilmiştir. 1960 ve 1970'li yıllarda artan otomasyona yönelik işler için çalışmalar gerçekleştirildi. 1990'lı yıllara gelindiğinde William Gray tarafından tasarlanan HelpMate robotu hastaneler ve bakımevleri için malzeme taşıma görevini üstlendi (Evans, 1989). Üzerinde barındırdığı ultrasonik ve kızılötesi sensörleri, hafızasında saklı olan harita üzerinde otonom şekilde hareket edebilme kabiliyetine sahipti. 2000'li yıllardan itibaren mobil robotik sistemler ivmelenerek hızlı bir şekilde ilerleme sağlandı. Sensörler ve işlemcilerin gelişmesi, ucuzlaması, yeni algoritmalar, yazılım sistemlerinin gelişmesi gibi etkiler ticari anlamda robot kullanımını yüksek seviyelere çıkardı. Depolama sistemleri, üretim, lojistik gibi alanlarda AMR ve AGV robotlar, hizmet sektörüne yönelik servis robotları ve evlere giren temizlik robotları birçok alanda hizmet vermeye devam etmektedir.

Mobil robotların otonom bir şekilde hareket kabiliyeti kazanması açısından bulunduğu ortamı algılaması ve tanınması gerekmektedir. Robotlar, üzerinde barındırdığı sensörler sayesinde yerleştirme problemine çözüm sağlaması ve ben neredeyim sorusuna cevap araması beklenir. SLAM problemi tam olarak bu konuda devreye girmektedir. SLAM problemi robotun bilmediği ortamı haritalandırması ve bu harita üzerinde giderken eş zamanlı bir şekilde kendini konumlandırması olarak ele alınır. Bu fikrin temelleri ilk olarak 1986 yılında IEEE Robotik ve Otomasyon Konferansında Smith

tarafından atılmıştır (Smith, 1986). Problemin çözümü için ilk yıllarında Genişletilmiş Kalman Filtresi tabanlı çözümler, olasılıksal yaklaşımlardan Rao-Blackwellized parçacık filtresi gibi yöntemler üzerine durulmuştur (Grisetti vd.,2007).

Problemin çözümüne yönelik ilerleyen aşamalarında kamera kullanılması fikri ile Görsel SLAM fikri doğmuştur. Kameralar ucuz maliyetleri, kolay erişilebilir olmaları, zengin veriye sahip olmaları gibi avantajları bulunmaktadır. Bu gibi avantajlarından dolayı araştırmacıların dikkatini çekmiş ve bu yönde çeşitli algoritmaların çıkmasına öncülük etmişlerdir. Görsel SLAM de kameralardan alınan görsel verilerin geometrik özellikleri ele alınıp bunun üzerinden kendini konumlandırması esasına dayanır. Konsept genel anlamda ön uç ve arka uç olmak üzere iki kısımdan oluşmaktadır. Ön uç adımında görüntü sensöründen gelen veriler ile görüntüden özellik çıkarımı ve takibi ile görsel odometre verileri elde edilir. Arka uçta ise konumlandırma esnasında biriken hataları optimizasyon teknikleri ile düzeltme sağlanmaktadır. Görsel SLAM de sensör girişleri için çeşitli tiplerde kameralar ve bu kameralara göre farklı algoritmalar kullanılmaktadır. Bunlar arasında monoküler ve stereo kameralar kullanılarak geliştirilen görsel SLAM uygulamalarıdır. Bu kameraların yanında IMU sensörü kullanılarak görüntü sensörünün ve çevresel gürültülerin önüne geçilmeye çalışılan Görsel-Ataletsel SLAM yaklaşımları ve RGB-D kameralar kullanılarak ham derinlik bilgilerini ele alan RGB-D SLAM yaklaşımları bulunmaktadır. Görsel SLAM de Durrant-White ve Bailey ile başlayan öncü görsel SLAM yaklaşımları günümüze kadarki ilerlemeler ve gelişmeler ile geleneksel yaklaşımlardan derin öğrenme temelli semantik V-SLAM sistemlerine doğru gelişim sağlamıştır (Chen vd.,2022).

1.2. Tez Organizasyonu

Bu tez çalışması kapsamında; İlk bölümde mobil robotların kullanımı ve SLAM probleminin tanımları yapılmıştır. İkinci bölümde geleneksel ve görsel SLAM algoritmalarının literatür taraması verilmiştir. Üçüncü bölümde SLAM tanımı, görsel SLAM de kullanılan teknikler, V-SLAM de kullanılan kamera çeşitleri ve kalibrasyon, görüntüden özellik çıkarımı ve eşleştirme algoritmaları, V-SLAM de kullanılan teknikler, yazılım ve donanım bileşenlerinden bahsedilmiştir. Dördüncü bölümde kullanılan algoritmaların performans analizleri ve verimlilikleri üzerine durulmuştur. Son bölümde ise değerlendirmeler yapılarak öneriler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

SLAM problemi önerildiği ilk yıllardan beri birçok araştırmacının dikkatini çekmiş ve günümüze kadar çeşitli yaklaşımlarla hızla yol katetmiştir. Problem literatürde genel olarak SLAM ve VSLAM olmak üzere iki ana başlık altında incelenmektedir.

2.1. SLAM Çalışmaları

SLAM probleminin tanımı ilk olarak 1986'da IEEE Robotik ve Otomasyon Konferansında Smith tarafından ortaya koyuldu (Smith, 1986). Çalışmada harita üzerindeki farklı yer işaretleri ile konumların arasında bir korelasyon olması ve bu korelasyonun zamanla büyüyeceği üzerine durulmuştur (Durrant-Whyte ve Bailey, 2006). Böylelikle tekniğin gelişimi için ilk adım atılmış olundu. 1991'de ilk SLAM algoritmalarından EKF-SLAM (Genişletilmiş Kalman Filtresi) yaklaşımı tanıtıldı (Leonard ve Durrant-Whyte, 1991). Sonrasında yapılan çalışmalarda problemlere olasılıksal yaklaşım tekniği üzerinden ilerlerdi.

2002 yılına gelindiğinde parçacık filtresi ve genişletilmiş kalman filtresi tabanlı hibrit bir sistem olan FAST-SLAM tanıtıldı (Montemerlo, Thrun vd. ,2002). Algoritma robotun pozisyonu için parçacık filtresini yer işaretlerinin konumu için ise EKF'den yararlanır. FAST-SLAM tanıtıldıktan bir yıl sonra daha etkili versiyonu FAST-SLAM2 yayınlanmıştır (Montemerlo, Thrun vd. , 2003). FAST-SLAM algoritmalarının önemli bir sorunu karmaşık çevrenin olduğu ortamlarda işlem yükünün artması ve hesaplama yapmada yetersiz kalmasıdır. İlerleyen dönemlerde algoritmayı geliştirmek için UKF-FAST-SLAM gibi algoritmalar ve çeşitli optimizasyon teknikleri ile algoritma geliştirilmeye çalışılmıştır (Zhu vd. , 2022).

2.2. Görsel SLAM Çalışmaları

2.1.1. Stereo / Monoküler vSLAM

Görsel SLAM çalışmalarındaki ilk konsept Davison ve arkadaşları tarafından önerildi (Davison, 2003). Çalışmada monoküler kamera ile Shi-Tomasi köşe detektörü kullanarak ortamın 3B haritası çıkarılmıştır. Durum tahmini için Genişletilmiş Kalman Filtresinden yararlanılmış ve GKF'nin hesaplama maliyeti için sınırlı örnekler seçilip kullanılmıştır. 2007 yılına gelindiğinde çalışmalarının devamı niteliğinde Davidson ve arkadaşları

tarafından monoSLAM önerilmiştir (Davison vd. , 2007). Çalışma ilk gerçek zamanlı vSLAM algoritması olarak kabul edilmektedir. Önceki çalışmalardaki gibi görüntüden özellik çıkarımı için Shi-Tomasi köşe dedektörü kullanılır. Durum tahmini için ise genişletilmiş kalman filtresinden yararlanır. Algoritmada küresel konumlandırma için optimizasyon teknikleri ve döngü tespiti kullanılmadığı için hassas doğruluk gerektiren uygulamalar için dezavantajlı olmaktadır. Aynı yıllarda ki bir diğer önemli çalışmada Paralel İzleme ve Haritalandırma (PTAM) algoritmasıdır (Klein ve Murray, 2007). PTAM temelde SLAM problemine iki farklı iş parçacığı şeklinde ele alarak işlem yeteneğini arttırabilen bir yaklaşımdır. İlk iş parçacığı olan ön uçta FAST detektörü kullanarak özellik çıkarma ve kamera hareket tahmini çıkarmaktadır. Diğer iş parçacığı olan arka uçta ise doğrusal olmayan optimizasyon kullanarak hatanın büyümesinin önüne geçmektedir.

2011 yılında yapılan bir diğer çalışmada literatürdeki ilk doğrudan yöntem örneği olan DTAM (Yoğun İzleme ve Haritalama) Newcombe ve arkadaşları tarafından önerilmiştir. DTAM mevcut görüntüdeki özellik çıkarımı ile değil her piksel için yoğun yöntem tabanlı sistem ile hesaplama yapar. Algoritma etkili ve detaylı bir şekilde veriler üretebilmekte fakat verilerin yoğunluğu hesaplama maliyetlerini arttırmaktadır (Newcombe vd. , 2011).

Engel J. ve arkadaşları 2014 yılında LSD-SLAM i önerdiler (Engel vd. , 2014). Algoritma üç aşamada gerçekleşir. İzleme, derinlik haritası tahmini ve optimizasyon. İlk aşamada sensor pozisyon tahmini için fotometrik hata en aza indirilir. Sonrasında derinlik tahmini için bir anahtar kare seçilir. Son olarak seçilen anahtar kareler haritaya eklenir ve optimizasyon yapılır. Aynı yıl yapılan bir diğer çalışmada Forster ve arkadaşları yarı doğrudan görsel SLAM algoritması SVO'yu önerdiler (Forster vd. , 2014). Algoritmada her bir görüntü için özellik çıkartmadan yüksek FPS'lerde ve düşük bütçeli donanımlarda çalışabilme kabiliyeti kazandırmıştır. Algoritma özellik tabanlı ve doğrudan yöntemin avantajlarını birleştirerek hareket tahmini ve haritalama olarak iki ana bölümde probleme yaklaşmıştır. Haritalama aşaması doğrudan yöntemle gerçekleştirilirken, hareket tahmini özellik eşleşmesi üzerinden gerçekleştirilmiştir.

2015 yılında Mur-Artal ve arkadaşları tarafından özellik tabanlı önemli bir algoritma olan ORB-SLAM önerildi (Mur-Artal vd. , 2015). Algoritma PTAM'e dayalı olarak yeni geliştirmeler eklenerek oluşturulmuştur. Sistem genel olarak izleme, yerel haritalama ve döngü kapanışı olarak üç aşamada çalışmaktadır. Özellik çıkarımı için ORB özellik eşleştirmesinden faydalanılmıştır. Yerel haritalama aşamasında haritaya yeni noktalar

eklenip tekrar edenler kaldırılır. Döngü kapama aşmasında DBoW (kelime çantası) denilen yöntem kullanılarak hızlı bir şekilde döngü tespit edilip harita optimize edilmektedir. Aynı ekip devam eden yıllarda geliştirmeler devam ederek ORB-SLAM2 (Mur-Artal vd. , 2017) ve ORB-SLAM3'ü (Campos vd. , 2021) yayınladılar.

ORB-SLAM'deki gelişmeler doğrultusunda 2017 yılında ORB-SLAM2 ve 2021 yılında ORB-SLAM3 yayınlanmıştır. Monoküler, Stereo ve RGB-D kameralar için uygulanabilen ORB-SLAM3 kolay entegrasyonu, yüksek hassasiyeti ve sağlam yapısıyla öne çıkmaktadır. Sistem gerçek zamanlı olarak yörünge hesaplama ve 3 boyutlu sahne yeniden oluşturma yapabilmektedir. Sistem hem iğne deliği (pinhole) hem de balıkgözü (fisheye) lens modellerini desteklemekte ve monoküler, stereo ve görsel-ataletsel sistemleri desteklemektedir.

2.1.2. Visual-Inertial SLAM

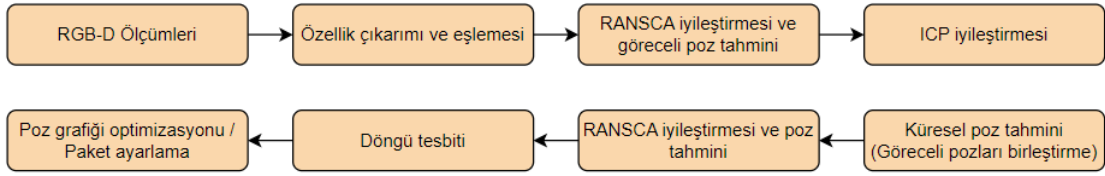
Görsel SLAM algoritmalarının düşük ışık şiddetinde, hızlı ve ani hareket durumlarındaki görüntü bozukluğu gibi nedenlerden ötürü IMU sensörleri ile birlikte kameraların kullanılması etkili SLAM çözümü için araştırma konusu olmuştur.

2007 yılında Mourikis ve arkadaşları tarafından MSCKF önerilmiştir (Mourikis ve Roumeliotis, 2007). Algoritma temelinde IMU ve Kamera verilerini genişletilmiş kalman filtresi ile birleştirmektedir. Algoritma özellikle İHA'larda başarılı bir şekilde kullanılmıştır. Sistem monoküler ve stereo kameralar için uygundur. Ayrıca düşük işlem gereksinimleri ile gömülü uygulamaları için uygundur. Algoritmanın yüksek titreşimli ortamlarda kararsızlığı artabilmektedir.

2015 yılında Leutenegger ve arkadaşları tarafından OKVIS önerildi (Leutenegger vd. , 2015). Optimizasyon temelli olan algoritma re-projeksiyon hatası ve IMU hatalarını optimize ederek bir amaç fonksiyonunda birleştirir. 2018 yılına gelindiğinde Hong Kong Bilim ve Teknoloji Üniversitesinden bir ekip Monoküler Görsel-Ataletsel Sistem (VINS-Mono) önerildi (Qin ve Shen, 2018). Algoritma hesaplamaya özellik çıkarımı ve takibi, IMU verilerinin ön işlenmesi ile başlar. Sistem yeniden konumlandırma için döngü tespiti ve pose-graph optimizasyonu modülü ile IMU ve özellik çıkarımını birleştirmektedir. Algoritma monoküler kamera ile IMU, stereo kamera ile IMU ve yalnızca stereo gibi çoklu sensör tiplerini desteklemektedir. İlerleyen süreçteki çalışmada GPS destekli sürüm olan VINS-Fusion da yayınlanmıştır.

2.1.3. RGB-D SLAM

RGB-D SLAM in temelinde görüntü ve derinlik bilgilerini paylaşan RGB-D kameralar yer almaktadır. Üzerlerinde bulunan projeksiyon sayesinde etrafa yapılandırılmış ışık yayarak derinlik verilerini elde edebilmektedirler. Şekil 2.1’de RGB-D SLAM işleyişi gösterilmiştir.



Şekil 2.1. RGB-D SLAM işleyişi (Yousif vd., 2015)

RGB-D kamerayı temel alan ilk görsel SLAM algoritmalarından biri KinectFusion algoritmasıdır (Newcombe vd., 2011). Sistem Yinelemeli En Yakın Nokta (ICP) algoritması ile kamera pozisyon tahmini için kullanılmaktadır. Algoritma orta düzeyde ortamlarda düzgün haritalar çıkarabilse de ortam büyüdükçe döngü eşlemesi olmadığı için biriken hata zamanla artmaktadır. 2013 te Kerl ve arkadaşları DVO-SLAM’i önerdiler (Kerl vd. , 2015). Sistem kamera hareket tahmini için fotometrik ve geometrik hatayı minimize etmektedir. Aynı zamanda döngü tespiti yaparak haritayı optimize etmektedir. Algoritma düz yapılı ortamlara karşı iyi performans göstermektedir.

RGBD-SLAM 2013 yılında ortaya çıktı (Enders vd. , 2013). Sistem özellik çıkarımından RANSAC algoritmasını kullanarak eşleşen görüntüler arasındaki dönüşümü hesaplamaktadır. Pozisyon tahmini için ise ICP algoritmasını kullanır. Ayrıca optimizasyon tekniklerini kullanarak biriken hatayı düzeltmektedir. RGBD-SLAM genel olarak yoğun ve ayrıntılı bir şekilde haritalar oluştursa bile kullanılan algoritmaların yoğun işlem gücü gereksinimleri bulunmaktadır.

Görsel SLAM probleminde, çevresel etkenler, SLAM performansına yönelik olumsuz etkileri bulunmaktadır. Düşük aydınlatma koşulları, karmaşık, dinamik, yapılandırılmamış ve büyük ölçekli ortamlara karşı daha gelişmiş çözümler aranmaktadır. Görüntü işleme tekniklerinin gelişimi SLAM probleminin ilerlemesine ve bu alanda birçok çalışma yapılmasına olanak sağlamıştır. Gelişen yeni teknolojiler ve derin öğrenme tabanlı yeni teknikler ile bu alanda ilgi giderek artmaktadır. Karmaşık otonom görevlerin yapılabilmesi ve insan-robot işbirliği sağlanabilmesi adına robotlarda çevresel algı kapasitesinin artırılması, giderek gelişen donanım ve yazılım mimarilerinin

entegrasyonu ile gelişmiş bir robot sisteminin inşasına yönelik çalışmalar giderek hız kazanmaktadır. Bu kapsamda son yıllara da gelişen derin öğrenme tabanlı sistemler ve anlamsal V-SLAM yaklaşımları ile birlikte gelecekte, SLAM bu alana kayarak geliştirilmesi ve uygulanması beklenmektedir.



3. MATERYAL VE YÖNTEM

Bu bölümde SLAM probleminin çözümünde kullanılan materyal ve yöntemler ele alınmıştır. Bu kapsamda SLAM ve V-SLAM kavramaları, haritalama ve lokalizasyon, V-SLAM için kullanılan yöntemler, kamera modelleme ve kalibrasyon, görüntü üzerinde özellik tespiti için kullanılan algoritmalar, V-SLAM sürecinde kullanılan algoritmalar ve görsel odometre çıkarımı, kullanılan yazılım ve donanım bileşenlerine yer verilmiştir.

3.1. SLAM

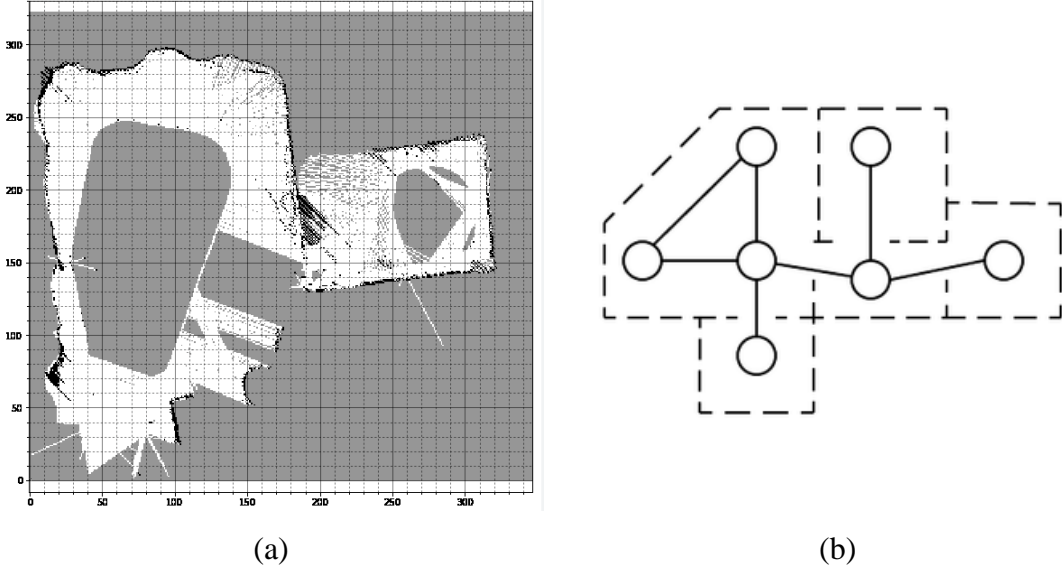
Eş Zamanlı Haritalama ve Yerelleştirme (SLAM) mobil bir robotun bilinmeyen bir ortamda kendini konumlandırması ve çevrenin haritasını çıkarması için kullanılan tekniğe verilen isimdir. Teknik kullanılarak çıkartılan harita ile robot yol planlama, yer tayini ve navigasyon yapabilme kabiliyeti kazanmaktadır. SLAM'in ana hedefi, robotun göreceli olarak çevrenin tutarlı bir haritasını çıkartmak ve aynı anda oluşturulan harita üzerindeki yer işaretlerini robotun konumunu tahmin etmek için kullanmaktır.

3.1.1. Lokalizasyon

Otonom mobil robotların en önemli kabiliyetlerinden biri yerelleştirme (lokalizasyon) özelliğinin olmasıdır. Yerelleştirme, robotun üzerinde bulunan sensörler vasıtasıyla (örneğin enkoder, LIDAR, kamera vs. gibi sensörler ile) harita üzerindeki konum ve yönelim bilgilerini tahmin edebilme kabiliyetidir (Durrant-Whyte, 2006). Sensörlerden gelen bilgiler ile robot kendi hareketini izleyerek yön tayinini sağlamaya çalışır. Bu bilgiler ile yol planlama ve hareket kontrolünü sağlayabilmektedir.

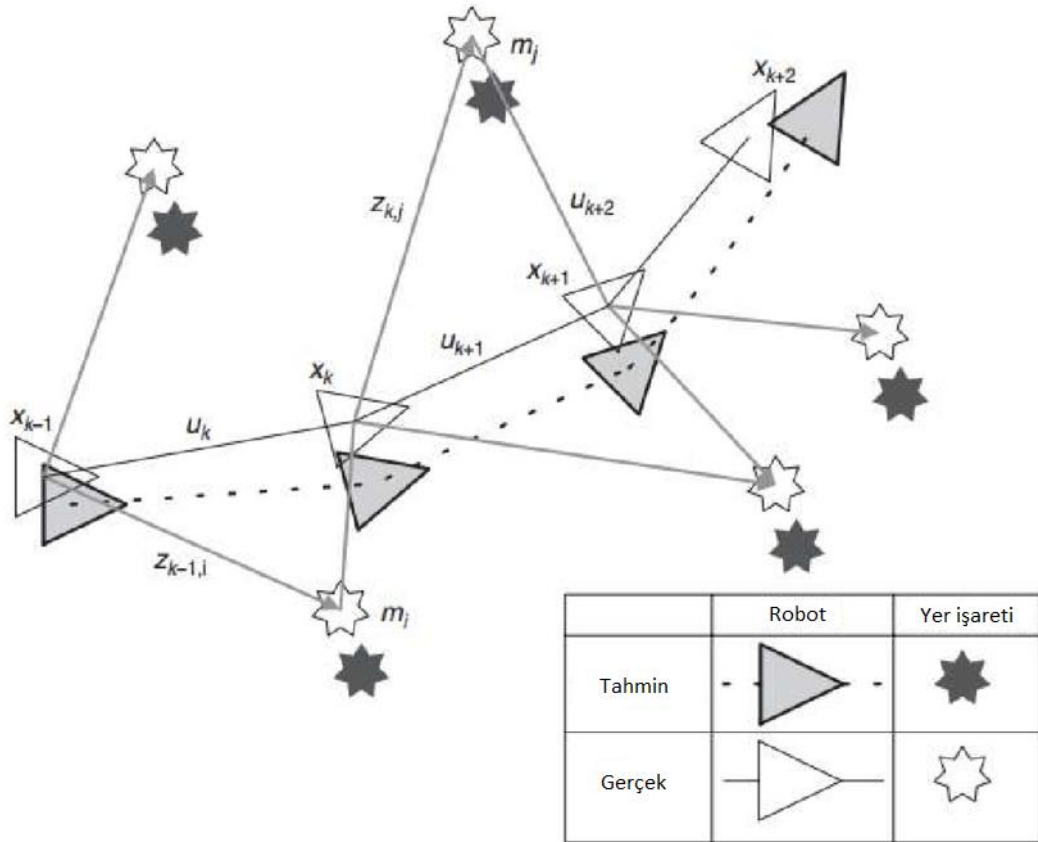
3.1.2. Haritalama

Haritalama otonom robotlarda konumlandırma ve navigasyon yapılabilmesi için önemli bir gerekliliktir. Görsel SLAM de öne çıkan iki ayrı harita tipi olan metrik harita ve topolojik harita vardır. Metrik haritalarda daha hassas haritalar üretilebilse de büyük ölçekli ortamlarda hesaplama yapmak zorlaşır. Topolojik haritada ise noktalar arası ilişkileri soyutlaştırarak verimli hesaplamaya yapmayı sağlar (Thrun, 1998). Şekil 3.1'de metrik ve topolojik harita örneği verilmiştir.



Şekil 3.1 (a) Metrik Harita (b) Topolojik Harita (Anderson, 2019)(Sundram vd., 2018)

SLAM'in önemli bir özelliği, yer işaretlerinin konumu anlık olarak tahmin edilip hesaplandığından, mobil robot ve çevrenin konumu hakkında önceden belirlenmiş bir bilgi gerektirmemesidir. Şekil 3.2 SLAM'in işleyişini göstermektedir.



Şekil 3.2 SLAM'in işleyişini (Durrant-Whyte ve Bailey, 2006)

Şekilde bir robot üzerindeki sensör verilerinin yardımı ile bilinmeyen yer işaretlerinin göreceli gözlemlerini alarak hareket eden robotumuz bulunmaktadır. Büyüklükler aşağıdaki gibi bir k anında tanımlanır:

- x_k : Robotun konum ve yönelimini veren durum vektörü.
- u_k : Aracı k zamanında x_k durumuna sürmek için $k-1$ zamanında uygulanan kontrol vektörü.
- m_i : Gerçek konumunun zamanla değişmediği varsayılan i 'nci dönüm noktasının konumunu tanımlayan vektör.
- z_{ik} : k zamanında i 'nci dönüm noktasının konumuna ilişkin araçtan alınan bir gözlem.
- X_k : Robot konumlarının geçmişi.
- U_k : Kontrol girdi geçmişleri.
- m : Yer işaretleri.
- $Z_{0:k}$: Dönüm noktası gözlemlerinin kümesi.

x_k , düzlemdeki konumu ve yönelimi ifade eder. X_k robot yolu için $k=0$ zamanından alınır ve denklem 3.1'de verilmiştir:

$$X_k = \{x_0, x_1, x_2, x_3, \dots, x_k\} \quad (3.1)$$

İki zaman adımı $k-1$ ve k arasındaki bağıl hareket U_k için denklem 3.2'de verilmiştir:

$$U_k = \{u_0, u_1, u_2, u_3, \dots, u_k\} \quad (3.2)$$

Düzlemde hareket eden bir robotun konumunu belirlemek için yalnızca robot odometrisine bağlı kalmak yeterince doğru olmaz çünkü gerçek uygulamada, çevresel durumlardan kaynaklı örneğin yüzey yapısı veya tekerlerin patinaj atması gibi durumlardan dolayı biriken odometri hataları doğru konumlandırma için gereken hassasiyeti sunmamaktadır. Robot gerekli hassasiyeti yakalayabilmesi için sürekli duyusal ölçüm Z_k 'ye de güvenilir. Zaman adımı başına Z_k duyusal ölçüm sırası denklem 3.3'de verilmiştir:

$$Z_k = \{z_0, z_1, z_2, z_3, \dots, z_k\} \quad (3.3)$$

SLAM tekniđi, robotun ve yer iřaretlerinin konumunu tahmin etmek iin oluřturulan haritada olasılıksal bir yaklařım ile probleme yaklařır. Olasılık dađılımı P denklem 3.4’de řu řekilde tanımlanır:

$$P(x_k, m | Z_k, U_k) \quad (3.4)$$

Buradan, lm gemiři ve odometri verileri zerinden, k zamanındaki konumun ve haritanın olasılıđı olarak ele alınabilir. Ayrıca gzlem modeli olan ek iliřkiyi de gerektirir. Gzlem modeli, robot konumu x_k ile odometri u_k arasındaki iliřkiyi belirtir ve denklem 3.5’de verilmiřtir:

$$P(x_k | x_{k-1}, u_k) \quad (3.5)$$

Gerekli olan diđer bir model ise hareket modelidir. Hareket modeli, sensr lmleri z_k , harita ortamı m ve robot konumu x_k arasındaki iliřkiyi belirtir ve denklem 3.6’da verilmiřtir:

$$P(z_k | x_k, m) \quad (3.6)$$

Genel uygulamada robot, tipik ortamdaki yer iřareti aralıđını, greceli yn ve benzersiz kimliđi tespit edebilir. Bu yntemle, bir lm modelinin ıkarılmasında temel olarak ele alınabilir. Olasılık dađılımını lm modeline uygulayarak denklem 3.7’de ki gibi tanımlanabilir:

$$P(z_k | x_k, m) \sim N(h(x_k, m), Q_k) \quad (3.7)$$

Burada $h(x_k, m)$, duysal ekipmanın alıřmasını temsil eden isteđe bađlı bir fonksiyondur, N , iki boyutlu normal dađılımdır ve Q_k , iki boyutlu grlt kovaryansdır. h fonksiyonu, konum ve evre haritasını girdi olarak kullanarak hesaplanmıř bir lm dndrr (Durrant-Whyte ve Bailey, 2006).

Hareket modeli iin kinematik hareket modeli kovaryansına odaklanan normal bir dađılım uygulanır. Hareket modelinin tretilmesi denklem 3.8’de řu řekildedir:

$$P(x_k | x_{k-1}, u_k) = N(g(x_{k-1}, u_k), R_k) \quad (3.8)$$

Burada $g(x_{k-1}, u_k)$ standart kinematik fonksiyondur ve R_k ,  boyutlu bir grlt kovaryansdır. g fonksiyonu nceki x_{k-1} konumunu ve odometri u_k cinsinden konum deđiřikliklerini birleřtirerek yeni x_k konumunu dndrr.

Robot, nceki zaman adımı $k - 1$ ’deki konumunu ve yer iřareti lmn, k zaman adımıdaki diđer yer iřaretinden ve nceki robot konumundan topladıđı verilerden geliřtirilmiř hassasiyetle gncelleyecektir. Robotun evredeki keřfini tamamlayana kadar robot lmnn gncellenmesi iin iřlem srekli olarak yapılır.

3.2. Görsel SLAM

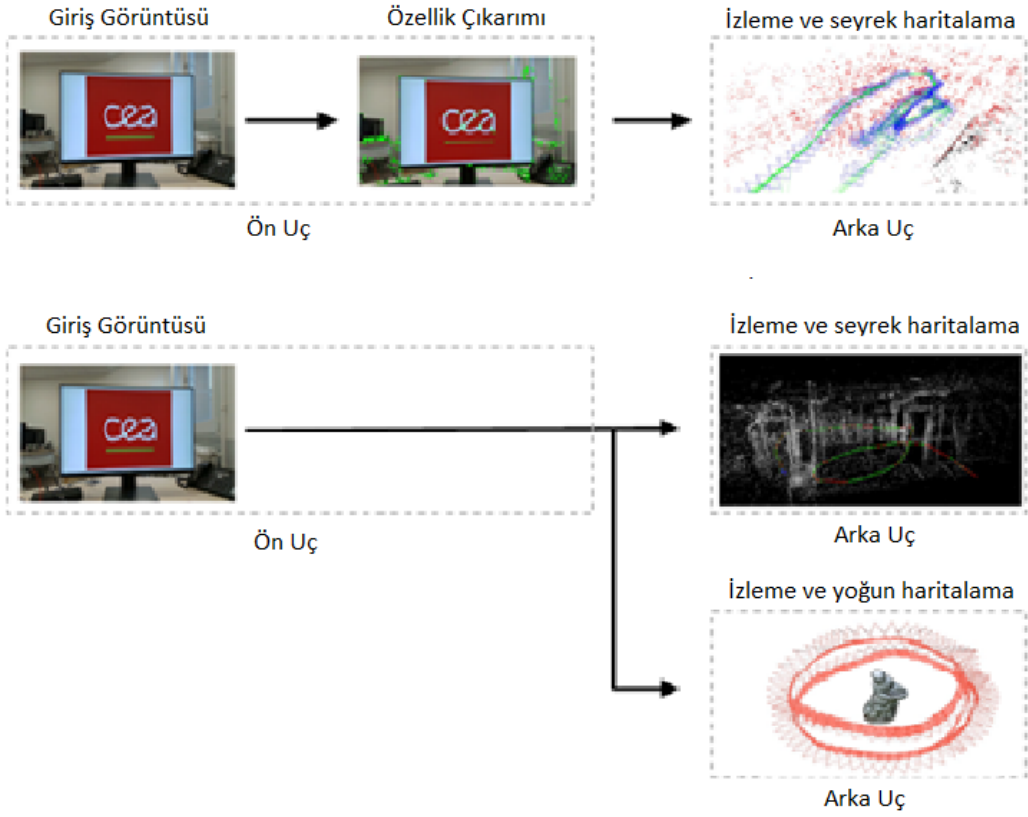
Görsel SLAM kavramı SLAM probleminin çözümü için kullanılan sensörlerden birincil olarak kamerayı kullanan yaklaşımdır. Sistem kullanılan sensörün pozisyonunu ve yönelimini belirlerken aynı zamanda çevrenin haritasını da çıkarabilmektedir. Probleme temelinde iki önemli yaklaşım olan özellik tabanlı ve doğrudan yöntem ile yaklaşmaktadır. Şekil 3.3’de özellik ve doğrudan tabanlı yönetime ait bir örnek görsel verilmiştir.

3.2.1. Özellik tabanlı

Özellik tabanlı yöntem, anahtar noktalar adı verilen çevrenin belirli geometrik şekillerine göre (köşeler, düzlemler, çizgiler) özellik çıkarımı ve eşleştirme yaparak kamera konumunu, yönelimini hesaplayan ve bu bilgiler ile ortamın haritasını çıkarabilen bir yöntemdir. Sistemin ilk zamanlarında özellik çıkarımı için FAST, Harris, GFTT gibi köşe bulma algoritmaları kullanılsa da bu algoritmaların basit, yapılandırılmamış ortamlardaki karasızlıklarından dolayı farklı özellik çıkarımı yapan algoritmalara yönelmiştir. Bu yöntemlerin başında SIFT, SURF ve ORB gibi özellik çıkartma yöntemleri gelmektedir.

3.2.2. Doğrudan yöntem

Doğrudan yöntemde özellik tabanlı yöntemin aksine kameradan alınana verilerin ön işleme alınmadan doğrudan sisteme verilmektedir. Görüntüdeki her pikselin yoğunluğu ele alınarak ve fotometrik hata azaltılarak çalışır. Böylelikle hesaplama maliyetleri açısından daha verimlidir. Ayrıca yapılandırılmamış düz dokulu ortamlarda daha iyi performans göstermektedir.



Şekil 3.3 Özellik ve Doğrudan tabanlı yöntem (Macario Barros, 2022)

3.3. Kamera Modelleme ve Kalibrasyon

Kamera sistemi 3 boyutlu uzaydaki nesnelere 2 boyuta aktaran sistemlerdir. Temelde monoküler, stereo ve RGB-D olmak üzere 3 farklı tür kameralar vardır. Şekil 3.4’de modellemede kullanılan kamera çeşitlerinden örnekler verilmiştir.



Şekil 3.4 Kamera çeşitleri (Lu vd., 2018)

3.3.1. Monoküler kamera

Monoküler kameralar yalnızca bir adet kamera sensörüne sahip kameralardır. Basit yapıları ve uygun maliyetleri ile görsel SLAM arařtırmaları için ilgi çekicidir. Monoküler kameralarda sensör çıktısı ile kameranın konumunu tek bir görüntü ile hesaplayarak elde edilememektedir. Bunu sağlamanın yolu ise kamera görüş açısını deęiřtirmekle elde edilir. Görüntüdeki piksellerin deęiřmesi ile yakın ve uzak nesnelere arasında eřitsizlik oluşur. Bu sayede ortam yapısı, boyutları ve mesafeleri tahmin edilebilmektedir. Ancak elde edilen bu sonuçlar görecelidir ve gerçek uygulamalarda işlem yükünü algoritmalara yıkmaktadır.

3.3.2. Stereo kameralar

Stereo kameralar iki veya daha fazla kamera sensörü kullanarak ortamın 3 boyutlu özelliklerini çıkarabilen kameralardır. Stereo kameralar taban çizgisi olarak adlandırılan arasındaki mesafe bilinen iki senkron kameradan oluşmaktadır. Aradaki mesafe bilindięi için insan algısına benzer bir şekilde ortamın derinlik bilgileri elde edilebilmektedir. Taban çizgisi arasındaki mesafe arttırıldığında uzak nesnelere mesafeleri tahmin etmek kolaylaşır fakat artan eřitsizlik hatasından dolayı yakındaki nesnelere için doğruluk azalır. Stereo kamera sistemlerinin dezavantajları arasında yüksek işlem gücü gerektirmeleri, kalibrasyon ve konfigürasyon gibi işlemlerin karmaşıklığı ve sınırlı görüş alanına sahip olmaları gibi nedenler sayılabilir.

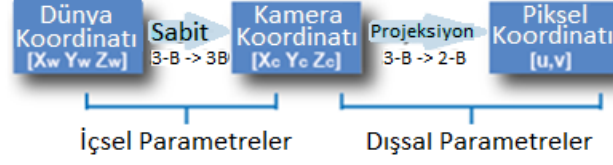
3.3.3. RGB-D kameralar

RGB-D kameralar hem RGB görüntü sağlayan bunun yanında görüntüdeki her pikselin derinlik bilgilerini de sunan kameralardır. Piyasada en yaygın bilinenler arasında Kinect ve Realsense bulunmaktadır. RGB-D kameralar yapılandırılmış kızılötesi ışık veya uçuş zamanı (ToF) prensiplerine göre çalışmaktadırlar. Uçuş-Süresi çalışma prensibi ışık hızını ele alarak toplam uçuş süresi hesaplanarak mesafe çıkarılır. Yapılandırılmış ışıklı kamera sistemlerinde ise cisimler üzerine yansıtılan ışıktaki deformeler ile derinlik bilgisi elde edilmektedir. RGB-D kameraların dezavantajlarına bakıldığında kısıtlı görme alanları, güneş ışınlarından etkilenme, dış ortamlarda çalışamama gibi sorunları bulunmaktadır.

3.3.4. Kamera kalibrasyonu

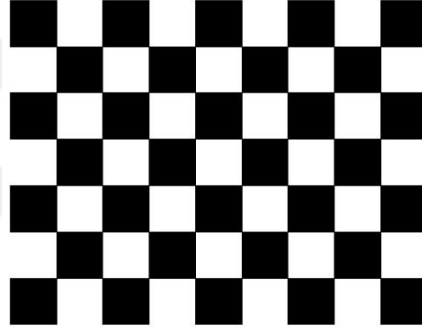
Görsel SLAM uygulaması için kamera kalibrasyonu önemli bir adımdır. İyi bir şekilde kalibre edilmiş bir kamera, SLAM performansı arttırmaktadır. Kalibrasyon sürecinde kameranın şekil 3.5’de verildięi gibi içsel ve dışsal olmak üzere iki önemli parametresi vardır. İçsel parametreler kameranın tasarımında olan odak uzaklığı, optik

merkezi, radyal distorsiyon gibi parametrelerdir. Dışsal parametreler ise doğrudan kamera ile ilgili olmayıp dünya koordinat sistemine göre dönme ve ötelenmesi ile ilgilidir.



Şekil 3.5 Kalibrasyon için içsel ve dışsal parametreler

Kamera kalibrasyon sürecinde kolay uygulanması ve yapılandırılabilir olması nedeniyle sıklıkla şekil 3.6'deki gibi dama tahtasına benzer düzlemsel bir şekil kullanılır.



Şekil 3.6 Kalibrasyon için kullanılan dama şekli

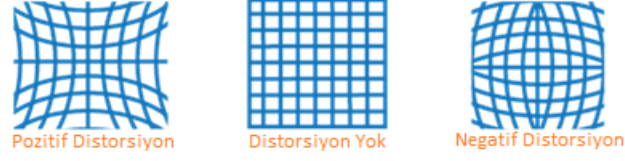
Kamera projeksiyon matrisi hesaplanırken içsel ve dışsal parametreler kullanılır. Pinhole kamera modeli için denklem 3.9'da aşağıdaki gibi belirtilir:

$$\lambda \begin{bmatrix} u \\ v \\ z \end{bmatrix} = KX = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.9)$$

Denklemden λ derinlik faktörü, u ve v koordinatları X, Y, Z olan 3B noktanın yansıtıldıktan sonraki 2B koordinatlarıdır. K içsel kalibrasyon matrisi, f_x ve f_y x ve y yönündeki odak uzaklığını, C_x ve C_y projeksiyon merkezinin koordinatlarını belirtir.

Kalibrasyon sürecinde kullanılan dama tahtasının boyutu, kare sayısı gibi özellikleri girilir. Sonrasında kameranın tüm alanı gördüğünden emin olunup çeşitli açı ve pozisyonlardan kareler alınıp kalibrasyon için gerekli veriler elde edilir. Kalibrasyon sürecinin ardından odak uzaklığı, temel nokta, radyal distorsiyon, teğetsel distorsiyon gibi parametreler elde edilir. İyi bir kalibrasyon sürecinin ardından görüntüdeki negatif veya

pozitif radyal bozunmalar düzelecektir. Şekil 3.7’de negatif ve pozitif bozunmaların görüntü üzerindeki piksellerde oluşturacağı etki verilmiştir.



Şekil 3.7. Bozunma şekiller

3.4. Özellik Çıkarımı ve Eşleme

Görsel SLAM in en önemli bileşenlerinden biride özellik çıkarımı ve eşlemedir. Görsel SLAM in ön uç adımında robotun pozisyon ve yönelimini tahmin etmek için görsel odometre çıkarımı önemli bir adımdır. Özellik tespiti görüntüdeki farklı bölgeler ve nokta özelliklerin tespitini yaparak hareket tahmininde bulunabilmeyi sağlamaktadır. Özellik çıkarımı için SIFT, SURF, FAST, BRIEF, ORB vb. çeşitli algoritmalar bulunmaktadır.

3.4.1. Harris köşe detektörü

Harris Köşe Detektörü, bilgisayarlı görme algoritmalarında sıklıkla kullanılan görüntüdeki özellikleri ve köşeleri yakalayan bir algoritmadır. Algoritma ilk olarak 1988 yılında Chris Harris ve Mike Stephe tarafından önerildi (Harris ve Stephens, 1988). Algoritmada köşe tespiti için daha önceden kullanılan her 45 derecelik kaydırılan parçalar yerine doğrudan köşe puanının farkını dikkate alarak kenarlar ile köşe arasındaki ayrımı daha iyi tespit etmiş olmaktadır. Köşe algılamada ön önemli ayırt edici unsur köşelerin iki ayrı kenarın birleşimi olmasıdır. Harris detektörü her piksel için bir pencere oluşturarak çalışır. Bu pencereler kaydırılarak piksel değerlerindeki değişimler dikkate alınır. Piksel değerlerinin önceki ve sonrası değerlerin kare toplam farkı (SSD) hesaplanarak değişimler gözlenir. Şekil 3.8’de Harris köşe dedektörü algılama yapısı verilmiştir.



Şekil 3.8 Harris köşe dedektörü algılama yapısı (Collins, 2007)

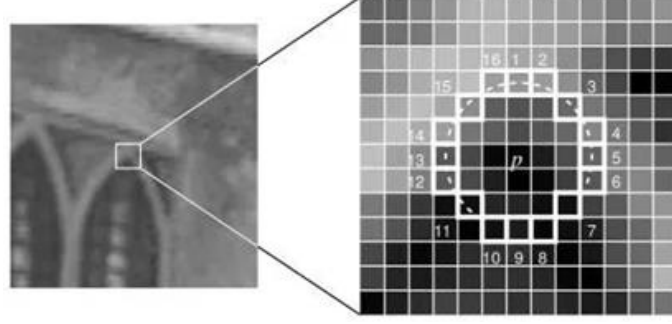
3.4.2. Shi-Tomasi köşe dedektörü

Shi-Tomasi köşe detektörü 1994 yılında J. Shi ve C. Tomasi tarafından köşe algılama algoritması olarak tanıtılmıştır (Shi ve Tomasi, 1994). Algoritma olarak Harris Köşe Detektörüne benzese de ondan daha iyi sonuçlar vermektedir. Bir görüntüdeki köşeler bahsedildiği üzere iki kenarın birleşimi olarak ele alınır. Görüntü üzerinde belirli boyutta oluşturulan pencereler ile her bir alan taranarak piksel yoğunluklarına göre köşe tespiti tahmininde bulunulur.

3.4.3. FAST corners

FAST köşe detektörü 2006 yılında Edward Rosten ve Tom Drummond tarafından yayınlandı (Rosten ve Drummond, 2006). Algoritmada makine öğrenmesi teknikleri ile geliştirilmiş olup hesaplama maliyeti ve süresi bakımından yüksek performanslı bir sonuç elde etmiştir. Bu bakımdan gerçek zamanlı uygulamalar için uygun olmuştur. Algoritmanın çalışması şu şekildedir:

- Yoğunluğu I_p olan görüntüde ilgili bir p piksel şekil 3.9'da belirtildiği gibi seçilir.
- Uygun bir eşik değeri t belirlenir.
- Kullanılan piksel üzerinde 16 piksellik bir daire varsayılır.
- Varsayılan daire içerisinde $I_p + t$ den daha parlak veya $I_p - t$ den daha koyu olan n piksel kümesi varsa p pikseli köşedir.



Şekil 3.9 FAST köşe algılayıcısı (Rosten ve Drummond, 2006)

Algoritmayı hızlandırmak adına dairedeki 1,5,9,13 numaralı piksel yoğunlukları ile I_p karşılaştırılır. Seçilen dört piksel üzerinden en az üçü $I_p + t$ nin altında değilse p pikseli köşe değildir. Aksi durumda 16 piksel kontrol edilip 12 bitişik pikselin kriterleri sağladığını kontrol edilmelidir. FAST algoritmasının uygulanmasındaki kısıtlardan biri olarak görüntüdeki x ve y eksenlerinin düzgün hizalanmış net görüntüler olmasıdır. Bu durumda algoritma köşeleri doğru yakalayamamaktadır. Bunun üstesinden gelebilmek için ise görüntüye gauss filtresi uygulanarak bulanıklaştırılmaktadır.

3.4.4. SIFT-SURF features

SIFT algoritması (Ölçekle Değişmeyen Özellik Dönüşümü) 2004 yılında D.Lowe tarafından yayınlanmıştır (Lowe, 2004). Görüntüdeki özellik tespiti, tanımlama, eşleştirme gibi özellikleri ile robotik, nesne tanıma, haritalama gibi birçok alanda kullanılmaktadır. SIFT temelde 4 adımdan oluşmaktadır. İlki özellikleri seçmek için konumları bulmak. İkinci olarak anahtar noktalarının doğru bir şekilde yerleştirilmesi. Üçüncü adımda seçilen anahtar noktalara yönlendirme atama aşaması. Son olarak anahtar noktalar bir belirleyici olarak vektör ile temsil edilerek iki görüntü arasındaki anahtar noktalar eşleştirilir.

SURF algoritması (Hızlandırılmış Güçlü Özellikler) SIFT'ten esinlenerek geliştirilmiş, SIFT'ten daha hızlı ve dayanıklı bir algoritmadır. 2006 yılında Bay H. , Tuyyelaars T. ve Van Gool L. tarafından önerilmiştir (Bay vd. , 2006). Algoritmanın yaklaşımındaki amaç kutu filtresi kullanarak gerçek zamanlı uygulamalar için hızlı ve güvenilir bir yapı sunmaktır.

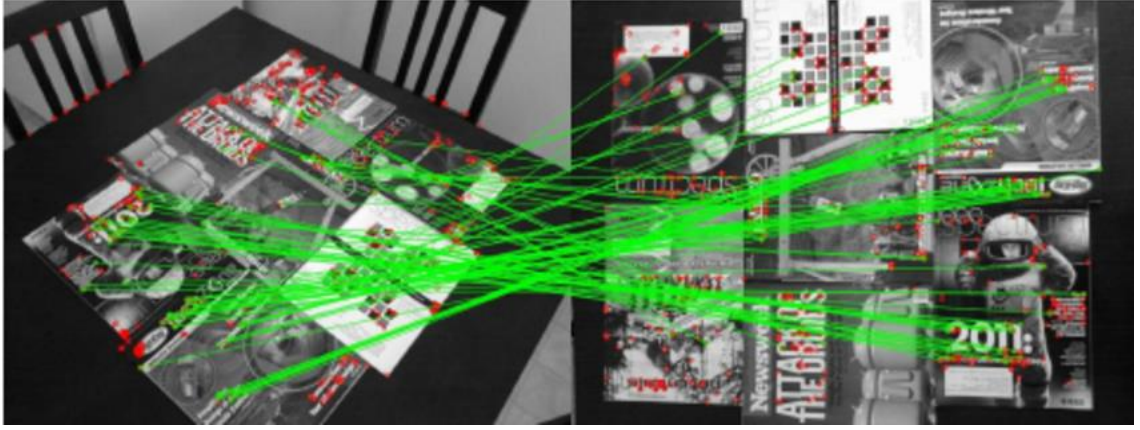
3.4.5. BRIEF descriptor

BRIEF algoritması özellikle hızlı ve güçlü olmasıyla öne çıkan özellik belirleyicisi algoritmasıdır (Calonder vd. , 2010). Yaygın olarak kullanılan SIFT ve SURF gibi algoritmalar ile karşılaştırıldığında daha iyi performans göstermektedir. BRIEF özellik noktalarını hafıza bakımından verimli olan 128-512 bitlik dizi olacak şekilde ikili

özellik vektörü ile tanımlanır. Ayrıca gürültüleri elimine etmek için gauss çekirdeği kullanarak görüntüyü yumuşatır ve kararlılığını artırır.

3.4.6. ORB

ORB algoritması SIFT ve SURF algoritmalarına alternatif olarak 2011 yılında E. Rublee, V. Rabaud, K. Konokige ve G. R. Bradski tarafından önerildi (Rublee vd. , 2011). Algoritma temelde BRIEF ile FAST detektörünün bileşenlerinden oluşur. FAST algoritması ile seçilen anahtar noktalar BRIEF algoritmasında ikili özellik vektörüne dönüşür. Ayrıca ORB çıkarılan her bir anahtar noktaya çevresindeki piksel değerlerine göre bir yönelim ekler. Yönlendirilmiş özellikler ile verimli hesaplanma ve korelasyon analizi yapılır. ORB ölçek değişkenliğine karşı dayanıklı olmak için orijinal görüntünün farklı ölçeklerde alt örneklerinden bir görüntü piramidi oluşturur. Şekil 3.10'da ORB özellik eşlemesi gösterilmiştir.



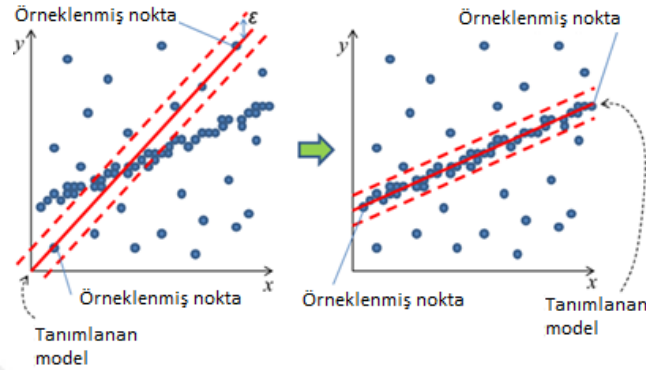
Şekil 3.10 ORB özellik tespiti (Rublee vd.,2011)

3.5. RANSAC

RANSAC (Rastgele Örnek Konsensusu) algoritması bir veri setindeki aykırı (outlier) değerleri yakalayan ve bu işlemi yinelemeli bir şekilde yapan yöntemdir. Algoritma ilk kez Fischler ve Bolles tarafından 1981 yılında yerleştirme problemine çözüm olarak önerildi (Fischler, 1981). Algoritma yinelenebilir bir şekilde verilerin alt kümelerine bir model oluşturarak bu modeller ile tüm veri kümelerini karşılaştırır ve tutarlı olup olmadıklarını kontrol eder. Şekil 3.11'de örnek işleyiş belirtilmiştir. Algoritma adımları aşağıda belirtilmiştir (Derpanis, 2005):

1. Modeli belirlemek için gereken minimum nokta sayısı rastgele seçilir.
2. Model parametreleri çözülmesi.
3. Noktalar kümesindeki kaç noktanın eşik değere uyduğunun kontrol edilmesi.

4. Kümedeki toplam nokta sayısına göre iç değer sayısının oranı tanımlanan eşik değeri aşarsa, tüm iç değerler kullanılarak model parametreleri yeniden tahmin edilir ve sonlanır.
5. Aksi durumda 1'den 4'e kadar olan adımlar tekrar edilir.

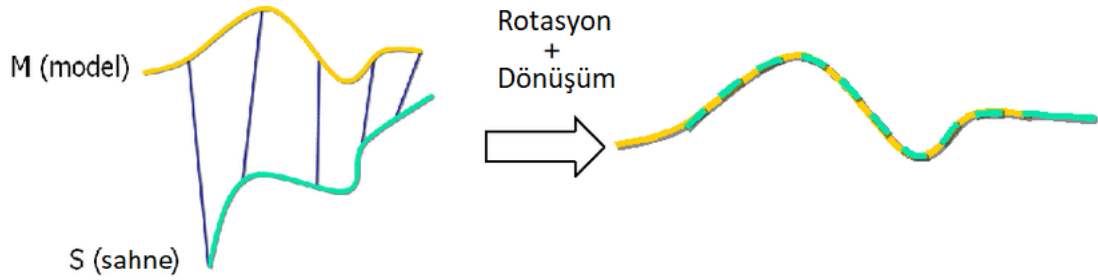


Şekil 3.11 RANSAC algoritması (Toshihiko Watanabe, 2013)

3.6. Döngü Kapama

Döngü kapama sistemi SLAM problemindeki daha önceden ziyaret edilen bölgeyi tespit edilebilmesi sağlayan konsepte verilen isimdir. Bu işlem için mevcut görüntü ile hafızadaki görüntünün eşleştirilmesi denetlenmektedir. Sistem özellikle büyük ortamlarda birken konumlandırma hatalarını optimize etmesi için önemli bir bileşendir. Literatürde döngü kapama sistemleri için LIDAR ve görüş tabanlı iki önemli yaklaşım mevcuttur.

LIDAR tabanlı döngü kapama sisteminde kullanılan en önemli yaklaşım Yinelemeli En Yakın Nokta (ICP) algoritmasıdır. ICP algoritması iki nokta bulutları arasındaki eşleşmeyi sağlayan algoritmadır (Chen ve Medioni, 1992). Özellikle robot konumlandırması için 2B veya 3B nokta bulutlarını kullanarak hedef ile referans verileri arasında dönüşümü sağlayabilmektedir. Şekil 3.12'de verilen ir referans veri seti M ile hedef veri seti S arasındaki eşleme için S 'deki her bir noktanın M 'deki en yakını bulunur. Daha sonra eşleşen noktalar arasındaki Öklid mesafelerinin toplamı hata hedefi fonksiyonu için hesaplanır ve hatayı en aza indirmek, rotasyon ve ötelemeyi hesaplamak için tekil değer ayrıştırması (SVD) kullanılır.



Şekil 3.12 ICP algoritması (Molero vd., 2011)

Görüş tabanlı döngü kapama sistemlerinde görüntüden çıkarılan özellikler kullanılarak kaynak görüntü ile arasındaki benzerlikler eşik değer referans alınarak karşılaştırılır. Özellik çıkarımı ve eşleştirme için kullanılan yöntemler daha önce bahsedilen SURF, SIFT, ORB gibi özellik çıkarımı yöntemler kullanılır. Döngü tespitinde önemli bir yöntem olan BoW (Bag of Word) yöntemi yakalanan dijital görüntüleri vektörler şeklinde temsil ederek görsel bir kelime ağacı oluşturan bir tekniktir. Yakalanan her bir yeni özellik kelime ağacında kök noktasından en küçük Hamming mesafesine sahip noktalara gidecek şekilde büyür. Her bir düğümde ki veriler hesaplanarak görüntü vektörü oluşturulur. Aşağıda denklem 3.10'da tanımlanmıştır.

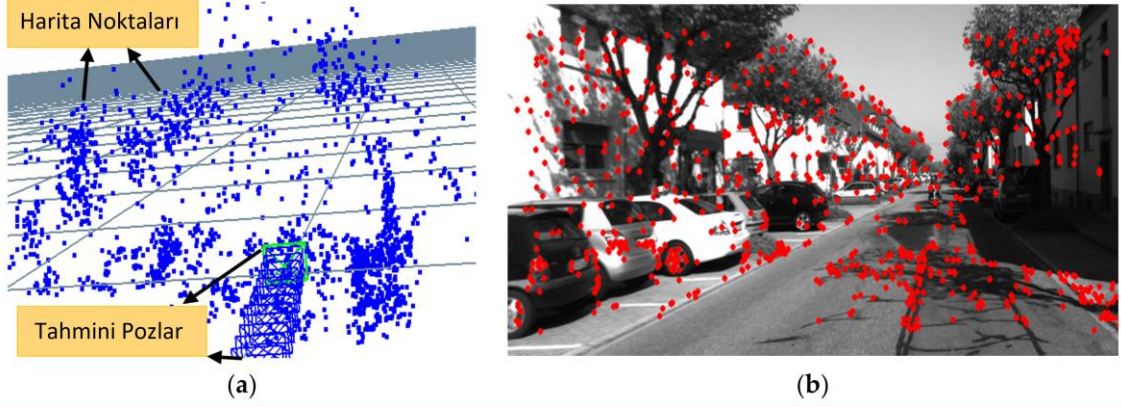
$$w_t^i = tf(i, I_t) \times idf(i), w_t^i = w_t^i / \sum w_t^{-1} \quad (3.10)$$

Elde edilen vektör temsilleri ile kaynak görüntüdeki temsil edilen vektörler ile karşılaştırma yapılarak döngü tespitinde bulunmaktadır. Bu yöntemin dezavantajlarına bakıldığında ise değişen çevre koşulları, ışıklandırma, görüntüdeki gürültüler vs. gibi olaylardan etkilenmesidir. Bu gibi sorunlar için FAB-MAP (Hızlı Görünüm Tabanlı Haritalama) gibi algoritmalar önerilmiştir.

3.7. Görsel Odometre

Görsel odometre(Visual Odometry), kameradan alınan görüntü girdilerini kullanarak kameranın hareketini tahmin etme işlemidir. Görsel odometre terimi ilk olarak Nister tarafından 2004 yılındaki makalesinde önerildi(Nister vd., 2004). Görüntüden pozisyon çıkarma fikri ilk olarak 1980'lerde Moravec öne sürdü(Moravec, 1980). Görsel odometre üzerine yapılan ilk uygulamalar NASA'nın Mars Rover aracı üzerinde olmuştur. Görsel odometre, bir araçta tekerleğin dönüş miktarını ölçen teker odometresi sistemine benzer bir çalışma prensibiyle pozisyonadaki değişimleri tahmin eder. Sistem

tekerlek odometresinde ki patinaj durumları gibi pozisyon ölçümünde hata oluşturacak etkilere karşı etkilenmez fakat düzgün bir şekilde çalışabilmesi için ortamda ışıklandırılmanın iyi düzeyde olması ve sabit dokulu nesnelerin olmaması gereklidir. Şekil 3.13’de algoritma işleyişi verilmiştir.



Şekil 3.13 Görsel odometri temsili (a) yerel 3 boyutlu harita ve (b) tespit edilen 2 boyutlu görüntü özellikleri. Her bir 3B harita noktası ile tespit edilen 2B özellik arasındaki uyumu bulmaktır. (Aladem ve Rawashdeh, 2018)

Görsel odometrenin amacı sabit bir çerçevede her bir zaman adımında verilen görüntü $X_k = T_k^{k-1} X_{k-1}$, dönüşümünü hesaplamak. Ardışık gelen görüntülerin perspektiflerini ilişkilendirilerek k , X_k adımındaki pozisyonun tahminini denklem 3.11’de sağlar.

$$T_k^{k-1} = (R_k^{k-1} \ t_k^{k-1} \ 0 \ 1) \quad (3.11)$$

$R_{k-1} \in SO(3)$ ve $t_{k-1} \in R^3$ sırasıyla $k-1$ ve k zamanındaki pozisyonlar arasındaki dönüş ve ötelemedir. Böylece denklem 3.11 takip edilerek kameranın X_0 başlangıç konumundan k zaman adımına kadar olan yörüngesi denklem 3.12’de yeniden oluşturulabilir.

$$T_k^0 = T_1^0 T_2^1 \dots T_k^{k-1} \quad (3.12)$$

Görsel odometre için özellik tabanlı (Features-Based) ve görünüm tabanlı (Appearance-Based) olmak üzere iki ana teknik vardır. Bunlardan görünüm tabanlı sistemde alınan görüntüdeki tüm piksel yoğunluklarına göre çıkarım yaparak çalışır. Özellik tabanlı yöntemde her karede belirli noktaların özelliklerini referans alarak çalışır. Sistem ilk olarak ardışık görüntü girdilerini almakla başlar. Alınan görüntüden SIFT, SURF, ORB gibi algoritmalar ile özellik çıkarımı yapılır. Sonraki adımda görüntüler arası

korelasyona bakılarak yerel arama tekniği ile sonraki görüntülerde takip edilir. Bir diğer adımda hareket tahmini yapılır. Kameradan gelen sıralı görüntüden birbirleri arasındaki pozisyon değişimine bakılır. Hareket tahmini için şekil 3.14’de belirtildiği üzere 2B’den 2B’ye, 3B’den 3B’ye ve 3B’den 2B’ye olmak üzere 3 farklı yaklaşım vardır. Son adımda ise yerel optimizasyonlar yapılarak biriken kaymaları ve hataları düzeltilir.



Şekil 3.14 Görüntüden hareket çıkarımı aşamaları

Hareket tahmini mevcut kare ile önceki kare arasındaki yer değiştirmeyi hesaplar. İki görüntü arasındaki I_{k-1} ve I_k dönüşümünün T_k , sırayla $k-1$ ve k zamanında karşılık gelen iki f_{k-1}, f_k özellik kümesinin hesaplanmasıdır. Hareket tahmini için 3 farklı yaklaşım öne çıkmaktadır.

2B’den 2B’ye Hareket Tahmini

I_k ve I_{k-1} olan iki kamera görüntüsü arasındaki geometrik ilişkiler, temel matris E olan matris ile tanımlanır. E ’nin hareket parametreleri denklem 3.13’deki gibidir;

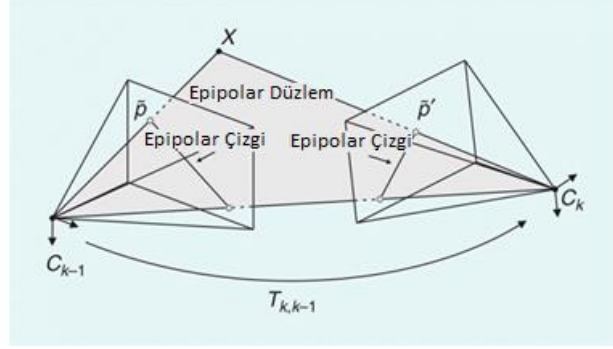
$$E_k \cong \hat{t}_k R_k \quad (3.13)$$

Buradan denklem 3.14 gelir. $t_k = [t_x, t_y, t_z]^T$

$$\hat{t}_k = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (3.14)$$

Temel matris 2B den 2B ye özellik karşılaştırmalarından hesaplanarak pozisyon değişimi hesaplanabilmektedir. Bu hareket tahmini özelliği birbirine karşılık gelen

$p \sim$ 'ün $p \sim$ özellik noktasının hangi çizgide olduğunu belirleyen şekil 3.15'de belirtilen epipolar kısıtlamadır. Bu kısıtlama $p \sim \cdot T_{k,k-1} \cdot p \sim = 0$ ile formüle edilebilir; burada $p \sim$ ' görüntüdeki özellik konumu ve $p \sim$ buna karşılık gelen diğer görüntüdeki konumudur.



Şekil 3.15 Epipolar geometri gösterimi (Fraundorfer ve Scaramuzza, 2011)

$2B'$ den $2B$ 'ye özellik karşılıkları bulunarak epipolar kısıtlama ile temel matris hesaplanabilir. Burada çözüm için Longuet-Higgins in sekiz nokta algoritmasına başvurulur. Her bir görüntü eşlemesi için aşağıda denklem 3.15'de formun bir kısıtlamasını verir;

$$[\tilde{u}\tilde{u}' \quad \tilde{u}'\tilde{v} \quad \tilde{u}'\tilde{v}' \quad \tilde{v}' \quad \tilde{u} \quad \tilde{v} \quad 1]E = 0 \quad (3.15)$$

Buradan denklem 3.16 gelir;

$$E = [e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8 e_9]^T \quad (3.16)$$

Burada sekiz nokta algoritması ile doğrusal denklem sistemini $AE = 0$ 'ı verilerek bu denklemden E 'nin değerleri hesaplanabilir. Bu denklem sistemi ise tekil değer ayrışımı (SVD) ile hesaplanabilmektedir. Aşağıda algoritma adımları verilmiştir.

1. Yeni kare l_k
2. l_{k-1} ve l_k arasında özellik eşlemesi ve çıkarımı.
3. l_{k-1} ve l_k görüntüleri için temel matrisi (essential matrix) hesaplama.
4. Temel matrisi R_k ve t_k ayırıp T_k elde etme.
5. Göreceli ölçeği hesaplayıp t_k 'yi buna göre yeniden ölçekleme.
6. Dönüşümü $C_K = C_{k-1} T_k$ 'yi hesaplanarak birleştirme ve 1. adımı tekrar et.

3B'den 3B'ye Hareket Tahmini

$3B'$ den $3B$ 'ye dönüşüm yaklaşımında hareket, görüntüdeki $3B$ özellik noktalarının üçgenleme (triangulation) metoduyla hesaplanır. Birbirine karşılık gelen $3B$

noktalar arasındaki 3B Öklid mesafesinin en aza indirilmesi ile çerçeveler arası dönüşüm denklem 3.17'deki gibi hesaplanır.

$$T = \arg \min_T \sum_i |X_{1i} - TX_i|^2 \quad (3.17)$$

Denklem 3.17 ele alındığında T ardışık iki kare arası tahmini dönüşüm, $\hat{X} F_k$ karesi tarafından gözlenen 3B özellik noktası, \hat{X} önceki F_{k-1} karesine karşılık gelen 3B özellik noktasıdır. i ise dönüşümü sınırlamak için gereken minimum özellik çifti sayısıdır. Algoritma adımları aşağıda verilmiştir;

1. $l_{l,k-1}, l_{r,k-1}$ ve $l_{l,k}, l_{r,k}$ sağ ve sol stereo görüntü çiftini yakala.
2. $l_{l,k-1}$ ve $l_{l,k}$ arasındaki özellik çıkart ve eşle.
3. Üçgenleme metodunu her stereo çifti için uygula.
4. 3B özelliklerden dönüşümü hesapla.
5. Dönüşümü birleştir $C_k = C_{k-1}T_k$
6. Adım birden tekrarla.

3B'den 2B'ye Hareket Tahmini

3B'den 2B'ye olan yaklaşım 3B'den 3B'ye olan yaklaşıma benzer şekilde işler aradaki fark gerekli dönüşümü bulmak için 2 boyutlu yeniden projeksiyon hatası en aza indirilmeye çalışılır. Denklem 3.18'de maliyet fonksiyonu verilmiştir;

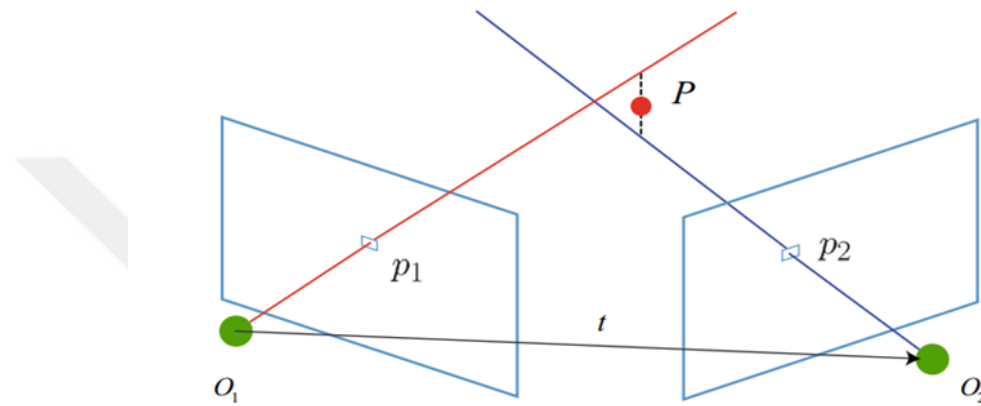
$$T = \arg \min_T \sum_i |z - f(T, \hat{X}_i)|^2 \quad (3.18)$$

Denklem 3.18 T dönüşümünü verir. Z mevcut F_k çerçevesindeki özellik noktasıdır. Fonksiyon ise T dönüşümü uygulandıktan sonra önceki çerçeveye karşılık gelen 3B özellik noktasının yeniden projeksiyon hatasıdır. i özellik çiftlerinin sayısıdır. Algoritma adımları aşağıda belirtilmiştir.

1. Yalnızca bir defa
 - 1.1. l_{k-2}, l_{k-1} iki görüntüyü al
 - 1.2. Aralarındaki özellikleri çıkart ve eşle.
 - 1.3. Üçgenleme metodu uygula l_{k-2}, l_{k-1}
2. Her iterasyonda
 - 2.1. yeni görüntüyü al l_k .
 - 2.2. l_{k-1} ve l_k arasındaki özellik çıkart ve eşle.
 - 2.3. Kamera pozunu hesapla(PnP) 3B'den 2B'ye.
 - 2.4. Üçgenleme metodunu eşleşenler için uygula.
 - 2.5. Adım 2.1.'den devam et.

Üçgenleme (Triangulation) Metodu

Hareket tahmini aşamasından sonra bu hareketlerden özellik noktalarının uzaysal konumlarını belirlenmelidir. Bu konumları ise üçgenleme (triangulation) metodu ile bulunmaktadır. Metot farklı konumlardan gözlenen aynı nirengi noktalarının o konumundan olan uzaklığının belirlenmesidir. Fikri ilk olarak Gauss tarafından önerilmiş olup astronomi ve coğrafya gibi alanlarda uygulanmıştır.



Şekil 3.16 Üçgenleme metodu (Gao ve Zhang, 2021)

Soldaki görüntü referans alındığında I_1 ve I_2 görüntüleri ele alındığında dönüşüm matrisi T 'dir. Şekil 3.16'da kameranın ana noktaları O_1 ve O_2 , p_1 özelliği I_1 'de ve buda I_2 'deki bir p_2 özelliğine karşılık gelir. Teoride O_1 , p_1 ve O_2 , p_2 çizgisi 3B harita noktası olan bir P noktasında kesişecektir. Ancak gerçek senaryolar ele alındığında görüntüdeki gürültüler, kamera modeli, kalibrasyon hataları gibi nedenlerden bu çizgiler kesişemez. Bunun üstesinden gelmek için ise en küçük kareler yöntemini başvurulur. Epipolar geometrideki tanımda x_1 , x_2 iki özellik noktasının normalleştirilmiş koordinatları olarak alındığında aşağıdaki denklem 3.19'u sağlarlar;

$$s_2 x_2 = s_1 R x_1 + t \quad (3.19)$$

Denklem 3.19 s_1 ve s_2 olan iki özellik noktasının derinliğini elde edilmelidir.

O_1 p_1 ışınının projeksiyonunu p_2 'ye yakınlaştırmak için 3B bir nokta bulunabilir. Benzer şekilde O_2 p_2 üzerinde veya iki çizginin ortasında da bulunabilir. Örnek olarak s_1 'i bulmak için yukarıdaki formülde her iki tarafı \hat{x}_2 ile çarpılarak denklem 3.20 elde edilir.

$$s_2 \hat{x}_2 = 0 = s_1 x_2 R x_1 + \hat{x}_2 t \quad (3.20)$$

$$s_2 \hat{x}_2 x_2 = 0 = s_1 x_2 R x_1 + \hat{x}_2 t.$$

Denklem 3.20'de sol taraf sıfır sağ taraf s_1 olarak ele alınabilir ve s_1 buradan hesaplanır. Böylelikle s_1 ve s_2 çerçevelerinin altındaki noktaların derinliği alınıp uzaysal koordinatları belirlenir.

3.8. Yazılım ve Donanım Bileşenleri

Robotik geliştiricileri için en önemli geliştirme araçlarından biri Robot İşletim Sistemidir (ROS). ROS'un temel amacı, robotikte tekrar eden işlerden kaçınmak ve donanım soyutlaması getirerek standartlaştırılmış çerçeve sunmaktır. Robot geliştiricileri ROS'un ortaya çıkışından önce donanımın yanı sıra, robot içindeki gömülü yazılımı tasarlamak için de önemli miktarda zaman harcamak zorundalardı. Bu ise tasarımdan, elektroniğe, matematik altyapısından, gömülü sistem programlama becerileri gerektiriyordu. Bu gibi donanıma ve diğer kısımlara harcanan durumlar ise asıl probleme odaklanmaktan geliştiricileri alıkoymaktadır. ROS, farklı disiplinlerden gelen uzmanlıklar ile ortaklaşa çalışır. ROS çerçevesi bir robotu tasarlamak, programlamak ve süreçleri standartlaştırarak donanımı yönetmeyi içerir. ROS, robotik cihaz oluşturmak için gerekli çeşitli araçları sunarak, robotik projeler üzerinde çalışması için mühendislere ve geliştiricilere gereken kolaylığı sağlamaktadır. Bu, birçok şirketin robotiğe yatırım yapmasını ve karmaşık robotik sistemleri daha hızlı tasarlamasını sağlar.

ROS fikri ilk olarak 2007 yılında E. Berger, K.Wyrobek, K. Salisbury, M.Quigley tarafından Standford Üniversitesi yapay zeka laboratuvarında üzerinde çalıştıkları şekil 3.17'de verilen PR1 robotu üzerinden gelişim gösterdi. Sonrasında süreç robotik çalışmalar yapan Willow Garage üzerinden ilerledi. Willow Garage'da PR1 robotu çalışmasının devamı niteliğinde ROS üzerinden çalışan PR2 robotunun çalışmalarına başlandı. Kısa süre sonra ilk sürüm olan ROS 0.4 Tango sürümü ve ROS üzerine belgeler ile makaleler yayınlandı (Quigley vd., 2009) . Sonrasında büyüyen topluluk için soru-cevap platformunun gelmesi, gelişen birçok paketlerin sürülmesi ile giderek yaygınlaşmaya başladı. 2013 yılına gelindiğinde ise platformun gelişimi Açık Kaynak Robotik Vakfı (OSRF) üzerinden ilerlemeye başladı. O tarihten itibaren robotik alanında dünya çapında geliştiriciler ve firmaların desteği ile birçok yeni sürümler ve araçlar geliştirildi.



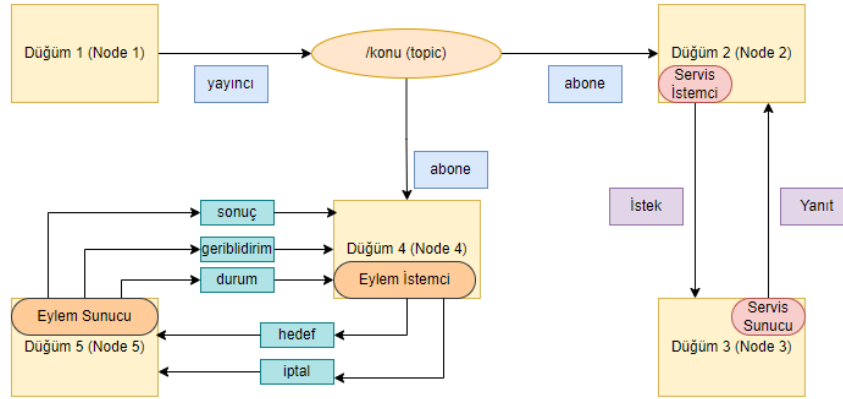
Şekil 3.17 PR1 Robot(Wyrobek, 2008) ve PR2 Robot (Benier vd., 2016)

ROS'un içinde işletim sistemi geçse de aslında bir işletim sistemi değildir. ROS daha çok dağıtılmış sistemdeki programlar için iletişimden sorumlu bir ara katman yazılımıdır. ROS, işlemler arasında mesaj aktarımı, donanım soyutlaması, sürücü işlevleri ve paket yönetimi gibi önemli bir dizi özellik sağlar. Her bir işlem parçacıkları, kontrol, aktüatör hareketleri, sensör verilerini alabilen (abone olma) ve gerekli yerlere gönderebilen (yayımcı) düğümlerde (node) gerçekleştirilir. Bu düğümler ise aralarındaki mesajlaşma için üç ana iletişim aracı olan konular (topic), servisler (service), eylem (action) dan oluşur.

Konular (Topic): Düğümler arasında veri akışlarını almak ve iletmek için kullanılır. Örnek olarak robottaki bir kameranın görüntüsü veya LIDAR sensöründeki veriler olarak düşünülebilir. Bu sensörleri okuyan düğüm, verileri göndererek herhangi bir düğümün bu konuya abone olarak gerekli bilgileri kullanabilmesini sağlar.

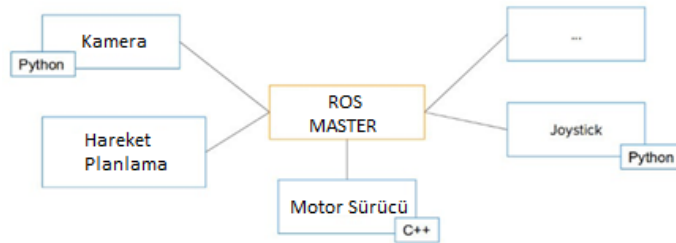
Servis (Service): Düğümler arasında kolay bir şekilde senkronize istemci/sunucu mantığında iletişim kurulmasını sağlar. Örnek olarak robottaki bir ayarı set etmek veya belirli bir sistemi aktif/pasif etmek için çok kullanılabilir: sürüş yöntemini etkinleştirme, sıcaklık verileri isteme vb.

Eylem (Action): Eşzamansız bir istemci/sunucu mimarisi sağlamak ve aynı anda geribildirim alabilme, kesintiye uğratma, iptal etme gibi servis ve konu mantığına benzer şekilde işleyen mesajlaşma sistemidir. Örnek olarak, robotu bir konuma gitmesini isteyen ve bunun için hedef konum gönderen bir istemci ve sunucunun bu durumunu eş zamansız olarak izleyebilir ve istediği zaman iptal edebilir olması gibi düşünülebilir. Şekil 3.18'de ROS mesajları çalışma şeması verilmiştir.



Şekil 3.18 ROS çalışma yapısı

ROS, yazılım için gerekebilecek parametreler için parametre sunucusu sağlar. Bu sayede düğümler kendileri için gereken parametrelere ortak havuzdan erişmiş olur. ROS'un bir diğer önemli özelliklerinden biri verileri görselleştirmeye ve hata ayıklamaya yarayan RVIZ, RQT gibi araçlar sunmasıdır. Ayrıca robotları sanal ortamda geliştirmek için Gazebo, WeBots gibi simülasyon araçlarını da desteklemektedir. ROS yazılım dilinden bağımsız çalışmaktadır. Oluşturulacak düğümler şekil 3.19'da verildiği gibi C++ veya Python la yazılıp birbirleri arasında iletişim kurabilirler.



Şekil 3.19 ROS iletişimi

ROS'ta birçok paket ve kütüphane desteğiyle gelmektedir. TF kütüphanesi ile robot çerçeveleri arasındaki dönüşümler hesaplanabilmektedir. SLAM, navigasyon, lokalizasyon, yörünge planlama gibi problemler için paketler sağlayarak hızlı bir şekilde robota entegre edilmesi sağlanır.

ROS geliştirilmeye başlandığı günden günümüze çok yol katetmiştir. Özellikle akademik araştırmalar ve şirketlerde yaygın kullanılmaktadır. Ancak endüstriye yönelik güvenlik, sertifikasyon gibi eksiklikleri gidermek için ROS2 ortaya çıkmıştır. ROS2 genel anlamda ROS1 de ki yapılarla benzer araçların yanında en önemli yeni özelliği Veri Dağıtım Hizmeti (DDS) ile gelmesidir. DDS, düğümler arası iletişim için bir ara katman

yazılımı görevi görür. DDS ara katmanı sayesinde düğümler arası iletişim için hızlı, sağlam ve güvenilir bir yapı oluşturulmuştur.



4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Çalışma kapsamında V-SLAM ile ilgili elde edilen kazanımlar çerçevesinde gerekli donanım, yazılım bileşenleri oluşturulmuştur. Bu kapsamda literatürde önemli yer edinmiş ORB-SLAM3 algoritması kullanılarak, veri setleri oluşturulmuş ve üzerinde çalışmalar gerçekleştirilmiştir. Özellikle farklı aydınlatma koşullarındaki ortamlarda kullanılan algoritma ve kamera sensörlerinin, sistem performansları üzerindeki etkileri, değerlendirilmiştir.

4.1. Donanım ve Yazılım Bileşenleri

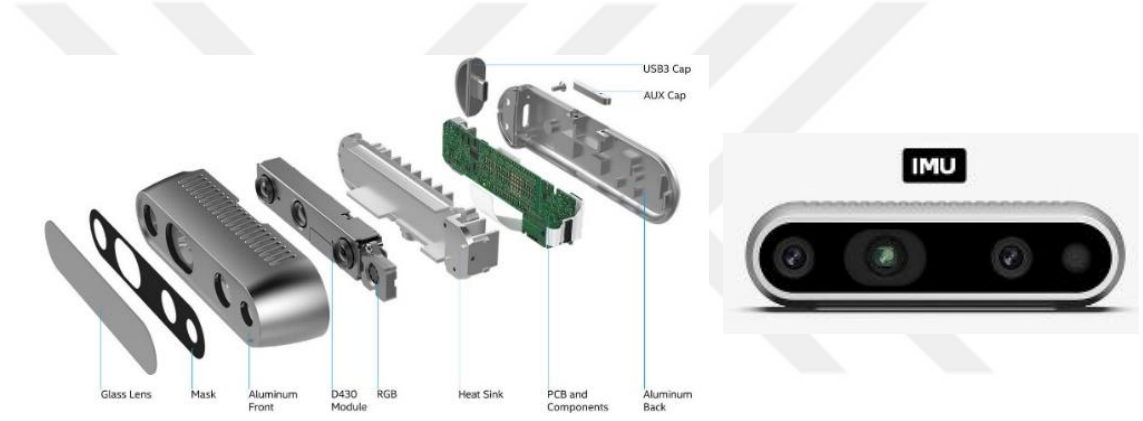
Çalışmada donanım bileşenleri kapsamında Intel i7-6700 HQ işlemci, GTX950m 4GB ekran kartı, 16 GB RAM, RealSense d435i derinlik kamerası ve kameraya dahili Bosh BMI055 6-eksen IMU sensörü, RPLİDAR-A2M8 lidar sensörü ve ışık sensörü kullanılmıştır. Oluşturulan mobil platform şekil 4.1’de verilmiştir.



Şekil 4.1 Mobil platform

Şekil 4.2’de verilen Intel RealSense derinlik kamerası, bir RGB kamera, iki kızılötesi kamera, bir kızılötesi lazer projektör ve bir eylemsizlik ölçüm birimi (IMU) ile donatılmış bir stereo görüş sisteminden oluşur. 1280x720'ye kadar aktif stereo derinlik

çözünürlüğü ve 1920x1080'e kadar RGB çözünürlüğe çıkabilmektedir. D435i'deki derinlik görüş alanı 87°x58° derece ve 90 FPS'ye kadar derinlik akışı sağlar. Stereoskopik görüntüleme ve kızılötesi nokta bulutu projeksiyonunun birleşimi ile normal koşullarda 3 metreye kadar etkili olup, robot navigasyonu, engel ve nesne algılama gibi çeşitli görevlerde kullanılabilir. Altı serbestlik derecesine sahip bir ivmeölçer ve jiroskop içeren dahili IMU, SLAM yeteneklerini, 3D taramayı, izlemeyi ve derinlik algılamayı mümkün kılar. Dahili IMU sayesinde, hareket sırasında derinlik algısını geliştirmesine olanak tanıyarak, daha iyi nokta bulutu verileri ile SLAM ve takip uygulamalarına yönelik güçlü özellik sağlar. IMU verileri, yüksek kaliteli derinlik verileriyle senkron şekilde hizalanmıştır. Bu özelliği sayesinde artırılmış gerçeklik, insansız hava ve kara araçları gibi uygulamalarda aktif olarak kullanılmaktadır.



Şekil 4.2 RealSense d435i derinlik kamerası

Geniş görüş alanı ve küresel deklanşör sensörü, robotik navigasyon ve nesne tanıma uygulamaları için idealdir. Global deklanşör sensörleri karanlık ortamlarda navigasyon için düşük ışığa karşı güçlü bir yapı sunar. Basit entegrasyonu, gelişmiş kolay ayarlanabilir kalibrasyon seçenekleri, güçlü topluluk desteği ile araştırmacılar ve endüstride kullanımı yaygındır.

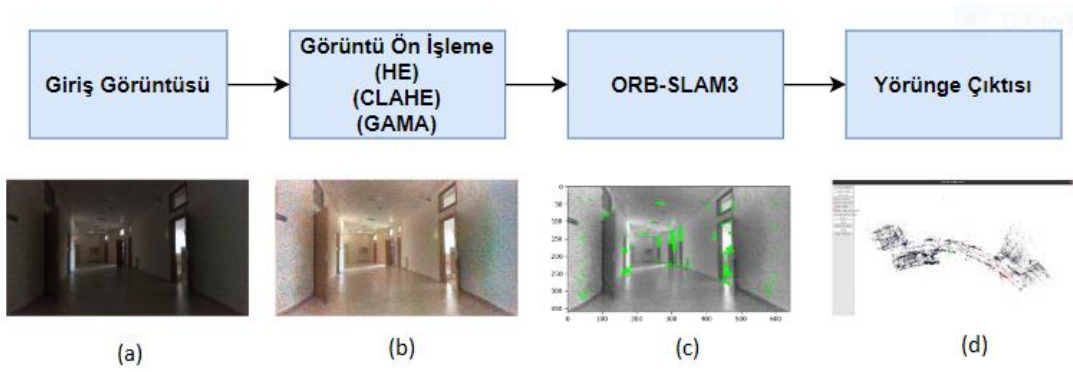
Çalışmada yazılım bileşenleri olarak Ubuntu 20.04 işletim sistemi üzerinde ROS Noetic versiyonu kurulmuştur. Yazılım dilleri olarak Python3 ve C++ dillerinden faydalanılmıştır. Görüntü işleme ve özellik çıkarımı için OpenCV kütüphanesi, optimizasyon problemlerinin çözümü için Ceres Solver ve g2o kütüphanelerinden yararlanılmıştır.

4.2. Görüntü Ön İşleme

Görsel SLAM performansını etkileyen dış etkenler olarak ortamın ışıklandırılması, ortamda bulunan doku ve yapısal özellikler, robotun hareket hızı ve

sistem üzerindeki titreşimler, ani sert hareketler, merdivenler, yürüyen merdivenler, asansörler, insan kalabalığı gibi birçok etken vardır. Bu çalışmada ortam ışıklandırılması üzerinde durularak düşük ışıklandırılmalı ortamlar için görüntü iyileştirme algoritmaları üzerinden SLAM performansı değerlendirilmesi yapılmıştır.

Bu kapsamda toplanılan görüntüler üzerinden görüntü ön işleme gerçekleştirilmiştir. Sistem şekil 4.3’de gösterildiği gibi görüntü ön işleme için alınan ham verilere histogram eşitleme, kontrast kısıtlanmalı uyarlanabilir histogram eşitleme ve gama düzeltilmesi uygulanmıştır.



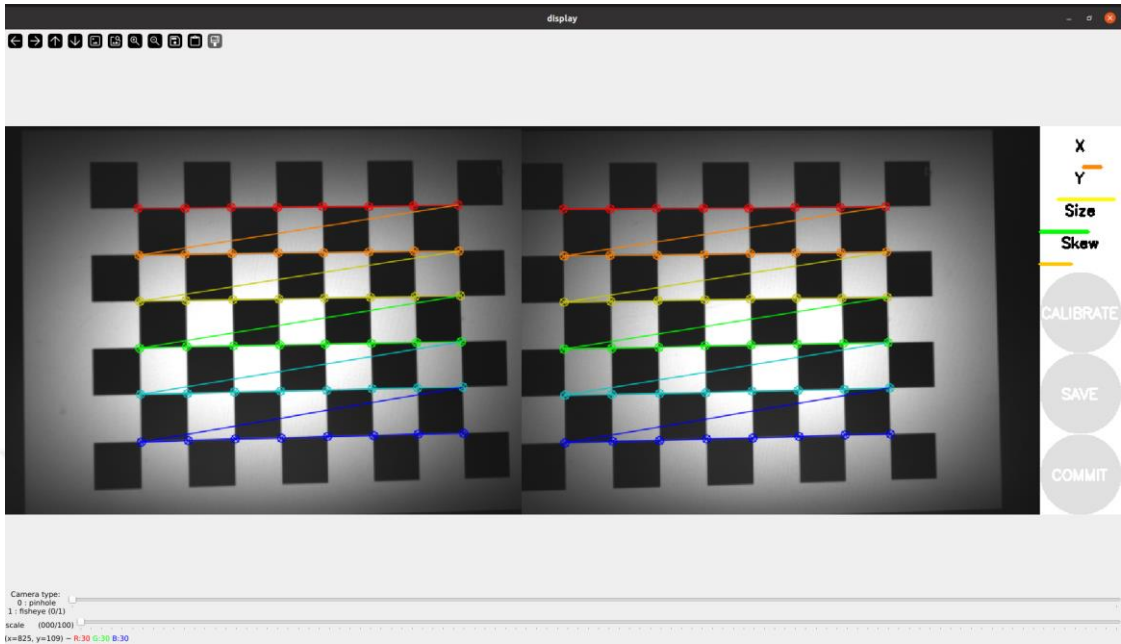
Şekil 4.3 Görüntü ön işleme sistem adımları (a) ham görüntü (b) görüntü iyileştirme uygulanan görüntü (c) iyileştirme uygulanmış görüntü üzerinden ORB-SLAM3 özellik çıkarımı (d) ORB-SLAM3 yörünge çıktısı

Histogram, bir görüntünün yoğunluk dağılımının grafiksel bir temsidir. Histogram eşitleme ise, görüntünün yoğunluk değerlerini yayararak görüntünün kontrastını iyileştirmek için kullanılan bir tekniktir. Uyarlanabilir histogram eşitlemede ise görüntü yerel parçalar ayrılır. Daha sonra bu parçalara eşitleme uygulanır. Ancak bu yöntemde görüntünün homojen bölgelerindeki gürültüyü aşırı yükseltme eğilimindedir. Bu sorunun üstesinden gelmek için kontrast sınırlaması uygulanır. Herhangi bir histogram bölmesi belirtilen kontrast sınırının üzerindeyse, bu pikseller histogram dengeleme uygulanmadan önce kırılır ve diğer bölgelere eşit şekilde dağıtılır.

4.3. Kamera Kalibrasyonu

Görsel SLAM uygulaması için kamera kalibrasyonu kritik öneme sahiptir. Farklı V-SLAM algoritmalarına farklı türde kamera çözümleri ile yaklaşılmaktadır. Bu kapsamda monoküler kamera, stereo kamera ve görsel-ataletsel kamera platformları için farklı türde kalibrasyonlar yapılmaktadır. Kalibrasyon süreci kameranın içsel (intrinsic) ve dışsal (extrinsic) özelliklerinin belirlenmesi olarak ele alınır. Monoküler ve stereo

kamera kalibrasyonu için şekil 4.4’de verilen 8x6’lık bir dama tahtası üzerinde farklı açılardan alınan görüntüler ile kalibrasyon işlemi gerçekleştirildi.



Şekil 4.4 Stereo kamera kalibrasyon işlemi

Kalibrasyon sonucu elde edilen temel parametreler odak uzaklığı (f_x, f_y), temel nokta (c_x, c_y), radyal distorsiyon ($k_1, k_2, k_3, k_4, k_5, k_6$), teğetsel distorsiyon (p_1, p_2) şeklindedir. Monoküler kamera için denklem 4.1’de odak uzaklığı ve temel nokta değerleri verilmiştir.

$$K = \begin{bmatrix} 430.215597 & 0 & 306.691343 \\ 0 & 430.531692 & 227.224800 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (4.1)$$

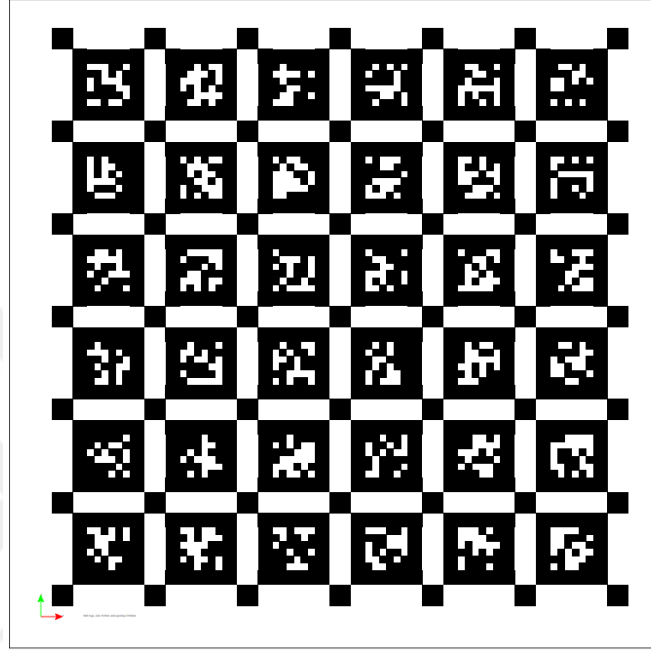
RealSense kamera için stereo kalibrasyon sonucu elde edilen parametreler sırayla sol ve sağ görüntü olmak üzere denklem 4.2 ve 4.3’te verilmiştir.

$$K = \begin{bmatrix} 928.452353 & 0 & 921.243742 \\ 0 & 660.467324 & 354.634126 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (4.2)$$

$$K = \begin{bmatrix} 934.890243 & 0 & 931.096532 \\ 0 & 656.437654 & 363.642875 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (4.3)$$

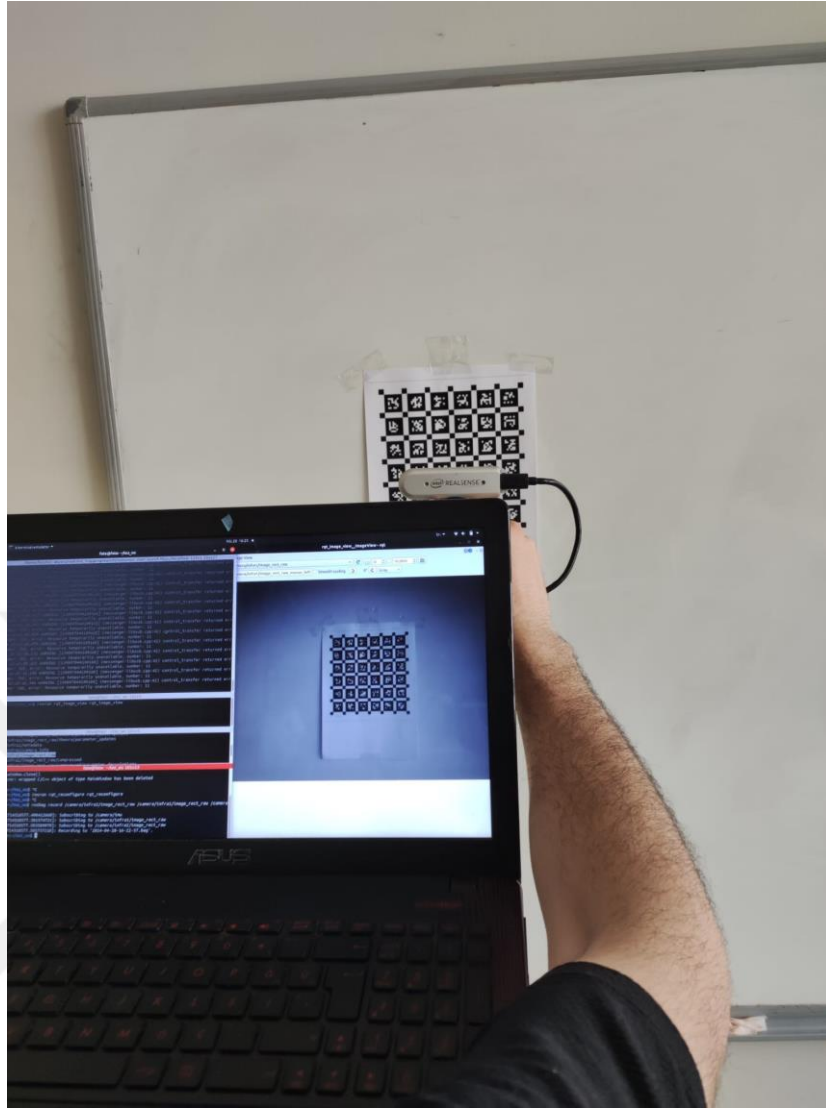
Görsel-Ataletsel SLAM sistemleri için kamera ve IMU sensörlerinin birlikte kalibre edilmesi gerekmektedir. Kamera-IMU kalibrasyonu, kamera sisteminin uzaysal ve zamansal parametrelerini kalibre edilmiş bir IMU'ya göre tahmin eder (Rehder vd.,

2016). Kalibrasyon için şekil 4.5’de verilen AprilGrid hedef kağıdı kullanılmıştır. AprilTag’lar daha az veriyi kodlamak için tasarlanmıştır. Bu sebeple kodu daha hızlı çözülebilmektedirler. Böylelikle gerçek zamanlı robotik uygulamaları için daha efektif kullanılırlar. Daha fazla özellik noktası tespiti, tutarlı ve tekrarlanabilir tespitler yapabildiği için AprilTag’ları kalibrasyon modeli olarak kullanmak avantajlardır.



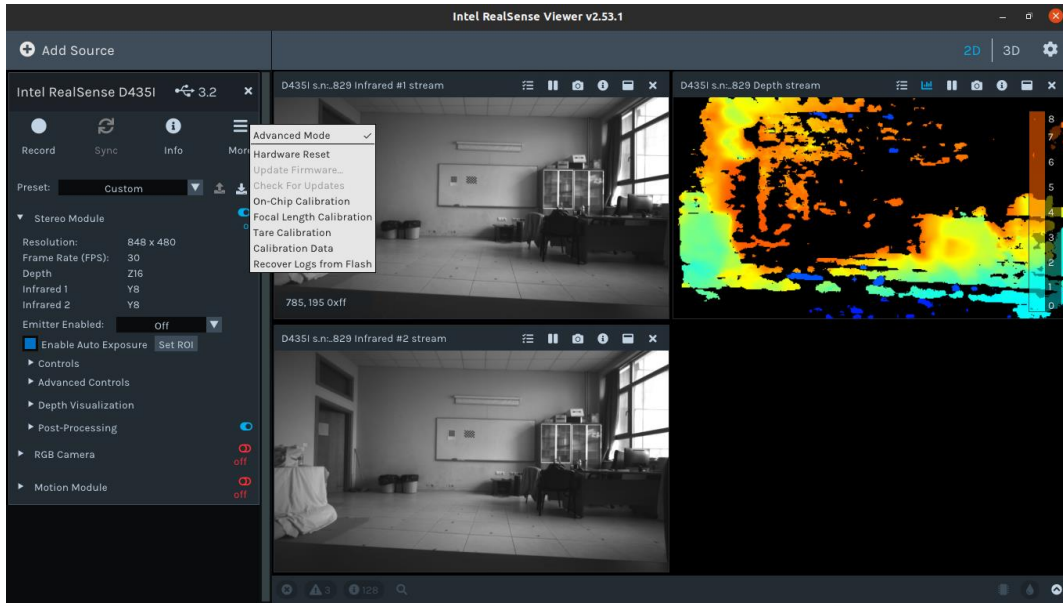
Şekil 4.5 AprilGrid

Kalibrasyona geçmeden önce belirli parametrelerin elde edilmesi gerekmektedir. Bunlar IMU’nun içsel özelliklerinin kalibre edilmesi, gürültü yoğunluğu (noise density) ve taraflı rastgele yürüyüş (bias random walk) parametrelerinin ayarlanmasıdır. Bu parametreler ayarlandıktan sonra iyi aydınlatılmış ortamda ve düşük pozlama zamanı ile hedef sabitlenerek kamera-imu sistemi tüm ekseninde (öteleme, dönme) hareket ettirilerek şekil 4.6’da gösterildiği gibi görüntüler rosbag dosyasına kayıt edilerek oluşturuldu.



Şekil 4.6 AprilGrid ile kalibrasyon

Kayıt edilen bag dosyalarındaki görüntüler kalibrasyon yazılımına verilerek kalibrasyon işlemi tamamlandı. Ayrıca şekil 4.7’de verilen Realsense-Viewer yazılımı üzerinden sunulan one-chip ve tare kalibrasyonları da uygulanarak daha iyi bir görüntü elde edildi.



Şekil 4.7 Realsense-Viewer kalibrasyon paneli

4.4. Çalışma Platformu

Çalışma kapsamında Necmettin Erbakan Üniversitesi Mühendislik Fakültesinde şekil 4.8’de verilen farklı ortamlardan görüntüler toplanmıştır. Tüm bu ortamlar için stereo kamera çifti, monoküler kamera, IMU ve ortam ışık şiddeti verileri farklı zaman ve ışıklandırma koşulları altında kayıt edilmiştir. Ayrıca V-SLAM algoritmalarını test etmek için kullanılan açık kaynaklı veri setlerinden EuRoc gibi en sık kullanılan veri seti ile çalışma gerçekleştirildi.



C blok kat 2



C blok kat 4



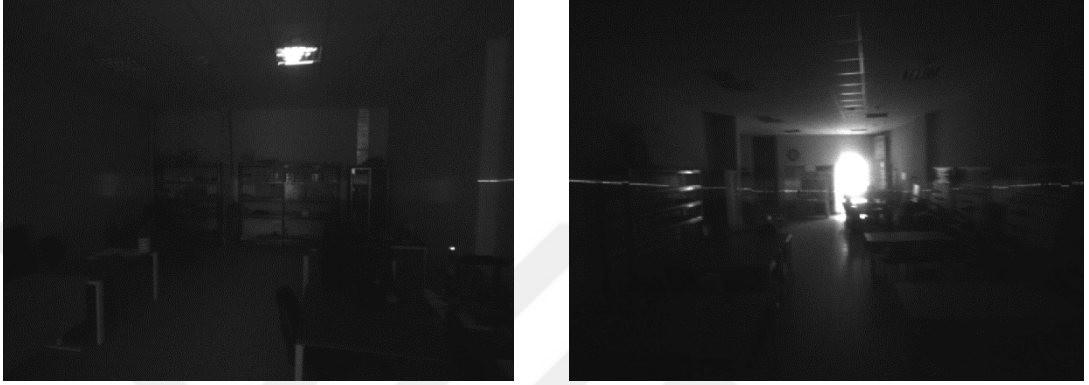
Mühendislik fakültesi kütüphanesi



Mühendislik fakültesi zemin kat

Şekil 4.8 Farklı ortamlardan toplanan görüntüler

Toplanan veri setleri, şekil 4.9 ve 4.10’da verilen ışık patlamalarının olduğu, farklı aydınlatma koşullarında toplanan görüntülerden oluşmaktadır. Her bir ortam için 640x480 piksel çözünürlükte, 30 FPS’te stereo kamera görüntüsü, 200 Hz frekansta IMU verileri ve 1 Hz frekansında ışık şiddeti sensöründen veriler alınmıştır. Kamera görüntüsü parametrelerinden pozlama ve ışık dengesi sabit tutulmuştur.



Şekil 4.9 Düşük aydınlatmalı ortamlar



Şekil 4.10 Işık patlamalarının olduğu ortamlar

4.5. Deney ve Gözlemler

Oluşturulan veri setleri üzerinde ORB-SLAM3 algoritması çalıştırılıp sonuçlar karşılaştırılmıştır. Bu kapsamda stereo-imu, verileri ile farklı ortam koşullarında yazılımlar çalıştırılmıştır. Performans analizi ve değerlendirme ölçeği için literatürde sıklıkla kullanılan mutlak pozisyon hatası (APE) ve göreceli pozisyon hatası (RPE) ölçekleri kullanılmıştır.

Göreceli pozisyon hatası (RPE) belirlenen bir Δ zaman aralığına bağlı olarak, yörüngenin yerel doğruluğunun bir ölçüsüdür. Göreceli pozisyon hatası, özellikle görsel

odometre sistemlerinin değerlendirilmesi için yörüngeden sapmanın göstergesidir. Bu tanıma göre i zaman adımıdaki göreceli poz hatası denklem 4.4'de verilmiştir:

$$E_i := (Q_i e^{-1} Q_i + \Delta)^{-1} (P_i^{-1} P + \Delta) \quad (4.4)$$

N tane kamera pozu dizisi boyunca göreceli poz hataları şu şekilde elde edilir: $m = n - /Delta$. Öteleme bileşeninin genel zaman endeksleri olan ortalama karesel hatanın kökü (RMSE) denklem 4.5'e göre şu şekilde hesaplanır:

$$RMSE(E_{1:n}, \Delta) := \left(\frac{1}{m} \sum_{i=1}^m \|\text{trans}(E_i)\|^2 \right)^{\frac{1}{2}} \quad (4.5)$$

Göreceli pozisyon hatası E_i 'nin öteleme bileşenleri $\text{trans}(E_i)$ ile gösterilir. Ölçülen aykırı değerlerin daha az etkilenmesi için bazı durumlarda ortalama hata yerine, kök ortalama kare hatası kullanılır. Bazı durumlarda ise ortalama yerine medyana hesaplamak da tercih edilir, böylelikle aykırı değerlerin etkisi azaltılmış olunur (Sturm vd., 2012).

Ardışık çerçeveleri eşleştirmeye çalışan sistemler için, çerçeve başına kaymayı $RMSE(E_{1:n})$ gösteren $\Delta = 1$ zaman parametresinin kullanılması gerekir. Bir sistemde birden fazla önceki çerçeve kullanıldığında, daha büyük Delta dağılımları uygun olabilir. Örnek olarak $\Delta = 20$, 20 Hz'de kaydedilen bir dizi görüntü için saniyedeki kaymayı verir. Başlangıç noktası ile bitiş noktasını karşılaştırmanın sıklıkla seçilen yöntemlerinden biri, $\Delta = n$ değerini ayarlamaktır. Bu ölçüm yörüngenin sonuna doğru dönme hatalarını daha fazla cezalandırdığı için aldatıcı olmaktadır (Kümmerle vd., 2009). SLAM sistemlerini değerlendirmek için tüm zaman aralıkları Δ üzerinden ortalama almak kullanışlıdır.

$$RMSE(E_{1:n}) := \frac{1}{n} \sum_{\Delta=1}^n RMSE(E_{i:n}, \Delta) \quad (4.6)$$

Denklem 4.6 yörünge uzunluğu açısından ikinci dereceden hesaplama karmaşıklığına sahiptir. Bunun için, sabit sayıda konumlardan bir dizi göreceli pozisyon örneği alınarak yaklaşık olarak tahmin edilebileceği önerilmiştir (Burri vd., 2016).

Mutlak pozisyon hatası (APE) V-SLAM sistemleri için, algoritmanın çıkardığı yörünge ile gerçek yörünge arasındaki mutlak mesafedir. Tahmin edilen yörünge ile gerçek yörünge arasındaki tutarlılığı belirlemek için kullanılan önemli ölçümdür. Her iki yörüngenin de herhangi bir koordinat çerçevesinde başlaması nedeniyle aralarındaki farkı hesaplayabilmek için hizalanması gereklidir. Bu sorunun çözümü için Horn yöntemi kullanılarak (Horn, 1987), en küçük kareler çözümünü ile tahmin edilen yörüngeyi $P_{i:n}$ temel gerçek yörüngeye $Q_{i:n}$ eşleyen bir katı cisim dönüşümü S elde edilebilir. Dönüşümden elde edilen mutlak pozisyon hatası aşağıda denklem 4.7'deki gibi hesaplanabilir:

$$F_i = Q_i^{-1}SP_i \quad (4.7)$$

Öteleme bileşenleri için, her bir bileşen tüm zaman endekslerinde ortalama karekök hatasının hesaplanmasıyla denklem 4.8'deki gibi çıkarılır:

$$RMSE(E_{1:n}, \Delta) := \left(\frac{1}{m} \sum_{i=1}^m \|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} \quad (4.8)$$

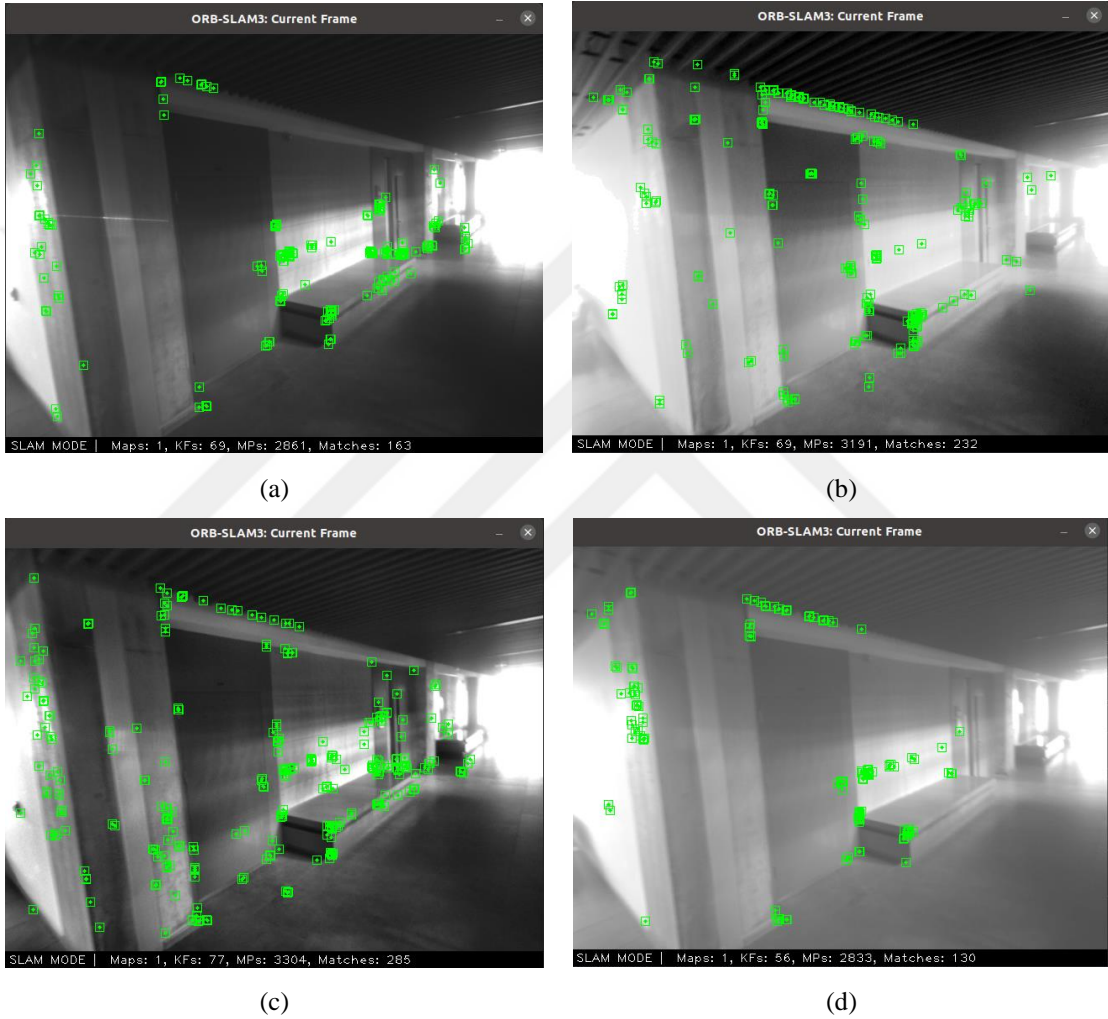
RPE, bir yörünge için genel hatasını değerlendirmek için olası tüm zaman aralıklarındaki verilerin ortalamasını alarak da kullanılabilir. Öteleme ve dönme hataları RPE tarafından önemliken, yalnızca öteleme hataları APE tarafından ele alınır. Bu nedenle RPE metriği, dönme ve öteleme hatalarını tek bir ölçümde birleştirmenin uygun bir yolunu sağlar. Ancak APE genellikle dönme hatalarını dolaylı olarak da algılar çünkü bunlar yanlış ötelenmelerde ortaya çıkar.

SLAM performansını ölçmek için genellikle mutlak pozisyon hatası ve göreceli pozisyon hatası kullanılarak değerlendirilir. Bu metrikleri ölçmek için ise yer gerçekliğini (ground truth) bilmek gereklidir. Yer gerçekliğini hesaplamak için genellikle harici hareket yakalama sistemleri veya dış ortamlar için GPS ve IMU sensörlerinden faydalanılır. Hareket yakalama sistemlerinin maliyetli ve sınırlı kullanım alanları olduğu için çalışmada toplanan veri setinde yer gerçekliğini ölçmek için LIDAR ve IMU sensörleri kullanılarak cartographer-SLAM den faydalanılmıştır. Çalışma ortamındaki koşullar ışık patlamalarının yoğun olduğu, farklı ışık şiddeti içeren verilerden oluşmaktadır. Görüntü iyileştirme yöntemlerinden histogram eşitleme, kontrast kısıtlanmalı uyarlanabilir histogram eşitleme ve gama düzeltme gibi algoritmalar değerlendirilmiştir. Yapılan iyileştirmeler özellik çıkarımı algoritmaları üzerinden test edilip sonuçlar gözlemlenmiştir. Şekil 4.11'de ham görüntüye histogram eşitleme uygulanmış hali verilmiştir.



Şekil 4.11 Histogram eşitleme sonucu görüntü çıktıları

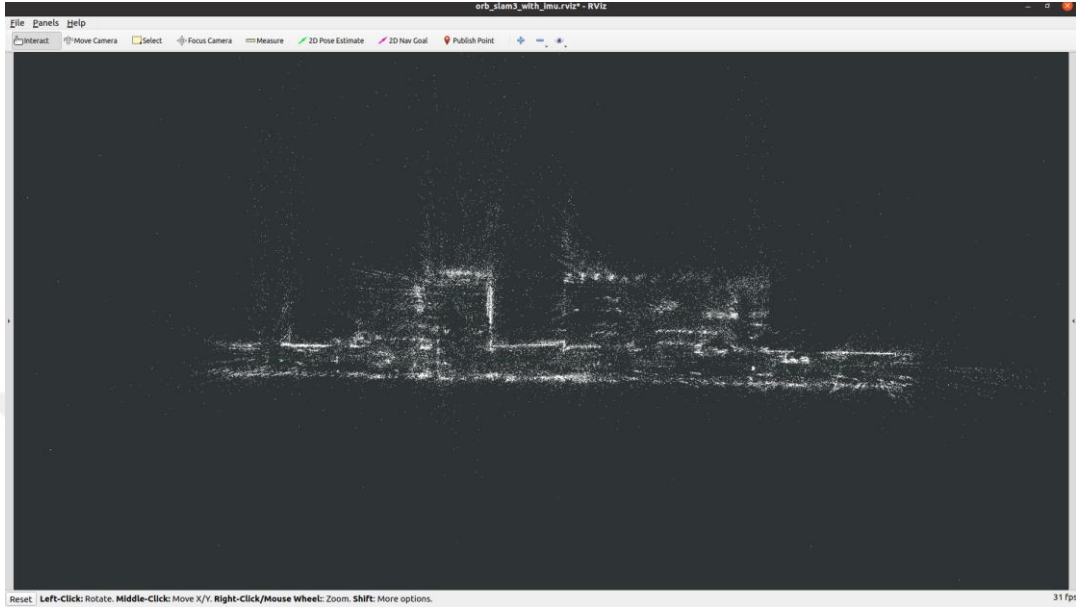
Düşük ışık şiddetinin olduğu ortamlarda SLAM algoritması görüntüdeki özellik noktalarını belirlemesi zorlaşır ve bu da hareket takibini engeller. Sistemde eklenen görüntü aydınlatma ile karanlık ve loş ortamlarda daha iyi özellik noktaları belirlenerek sistemin daha sağlam yapıda olması beklenmektedir. Yapılan iyileştirmeler üzerinden görüntüler ORB özellik tespiti sonuçları aşağıda şekil 4.12’de verilmiştir.



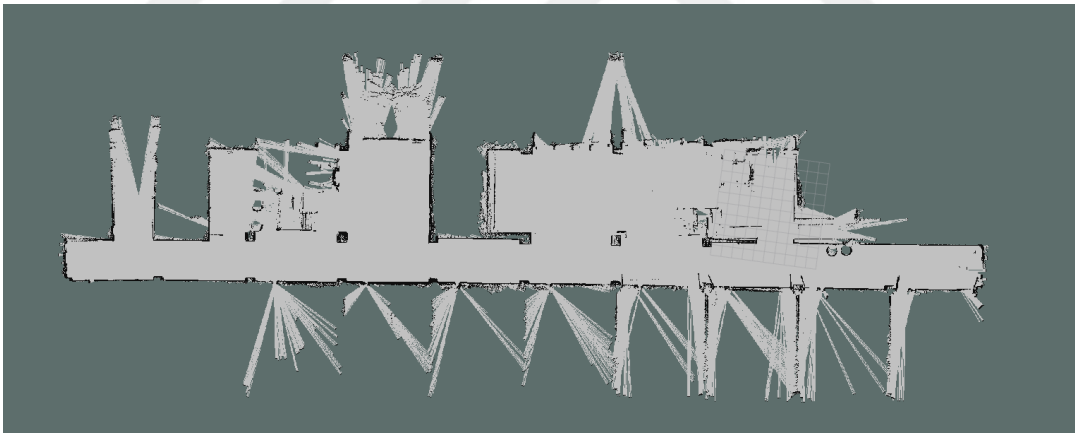
Şekil 4.12 Görüntü iyileştirme uygulandıktan sonraki ORB özellikleri (a) orijinal, (b) histogram eşitleme, (c) kontrast sınırlamalı uyarlanabilir histogram eşitleme, (d) gama düzeltme

Şekil 4.12 incelendiğinde yapılan iyileştirmeler sonucu en iyi eşleme ve özellik çıkarımı kontrast kısıtlamalı uyarlanabilir histogram eşitlemede elde edildiği görülmektedir. İkinci en iyi eşleme ise normal histogram eşitlemede, üçüncü olarak ise ham verilerin eşleşmesinde görülmektedir. En az eşleme ise gama düzeltmesi uygulanan görüntüde olmuştur.

Üniversite içerisinde toplanan verilerden elde edilen haritalama sonucu şekil 4.13’de gösterilmiştir. Harita ORB-SLAM3 ve cartographer-SLAM yöntemiyle çıkarılmıştır.



(a)

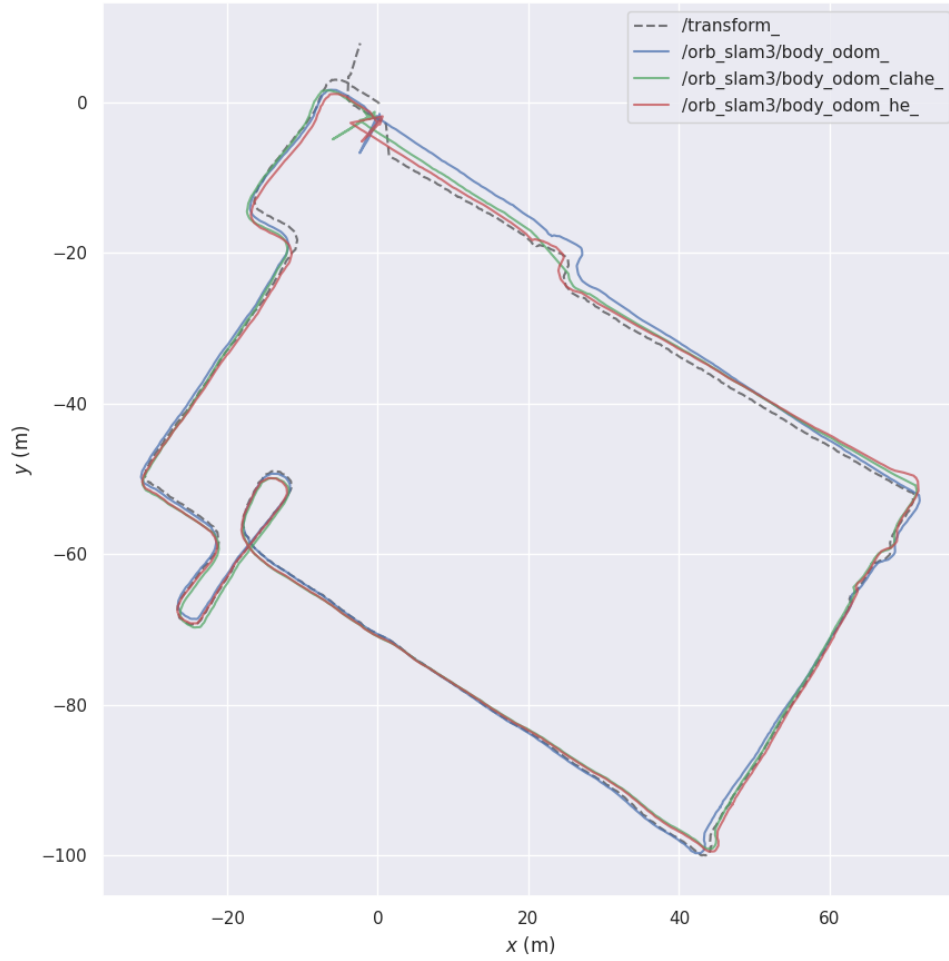


(b)

Şekil 4.13 Mühendislik fakültesi zemin kat (a) ORB-SLAM3 3B nokta bulutu haritası ve (b) 2B Cartographer haritası

Yapılan iyileştirmeler sonucu ORB-SLAM3 algoritmasının ürettiği yörüngeler EVO-ROS paketi kullanılarak çizdirilmiştir. SLAM performans sonuçları için tahmini konumları ile yer gerçeği arasındaki tutarlılığı mutlak pozisyon hatası (APE) ve göreceli pozisyon hatası (RPE) üzerinden değerlendirilmiştir.

Mühendislik fakültesi 1. katta toplanan veriler üzerinden ORB-SLAM3 yörünge çıktıları şekil 4.14’te verilmiştir.



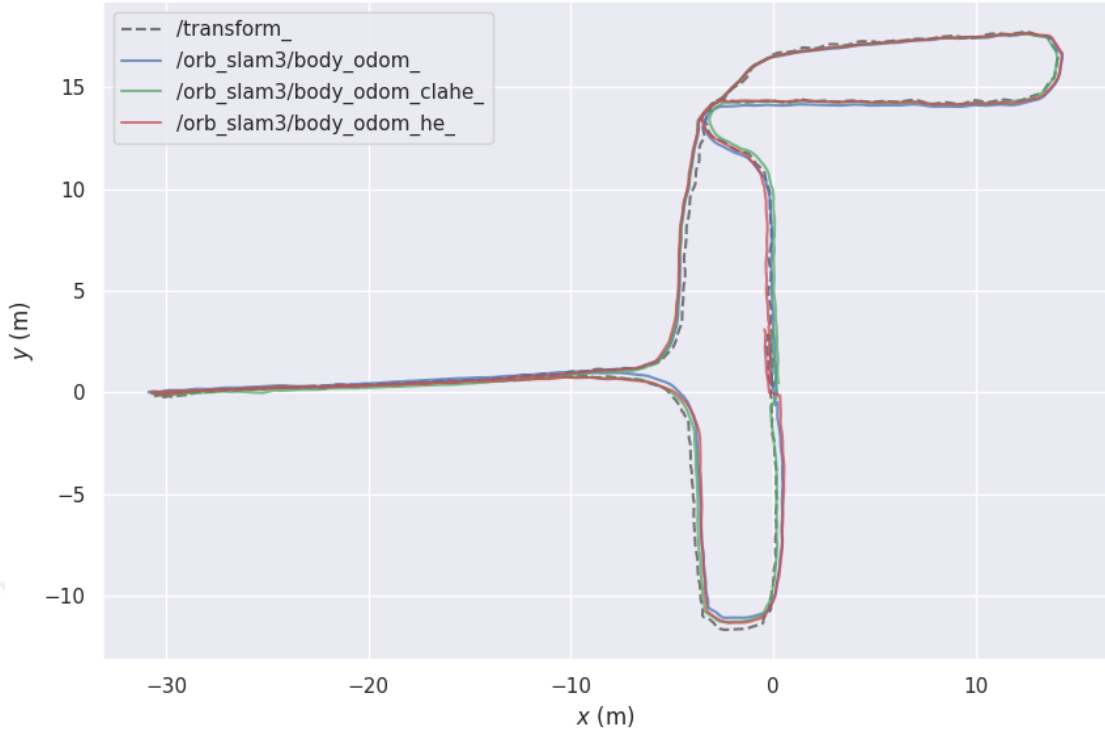
Şekil 4.14 Mühendislik fakültesi kat 1 yörünge çıktıları

Çizelge 4.1 SLAM sonuçları(mühendislik fakültesi kat 1 verisi)

ORB-SLAM3				
Kamera Tipi	Metrik	Ham veri	Histogram Eşitleme	Kontrast Kısıtlamalı UHE
Stereo	APE	2.1948	1.9775	1.7642
	RPE	0.5239	0.5166	0.3456

Çizelge 4.1’de yer alan sonuçlar incelendiğinde stereo haritalama yönteminde kontrast kısıtlamalı uyarlanabilir histogram eşitleme uygulanan veri setinin en iyi sonucu çıkardığı görülmüştür. İkinci olarak ise normal histogram eşitleme uygulanan veriler daha iyi sonuç vermiştir.

Şekil 4.15’ de mühendislik fakültesi kütüphanesinin olduğu verilerin ORB-SLAM3 yörünge çıktıları verilmiştir.



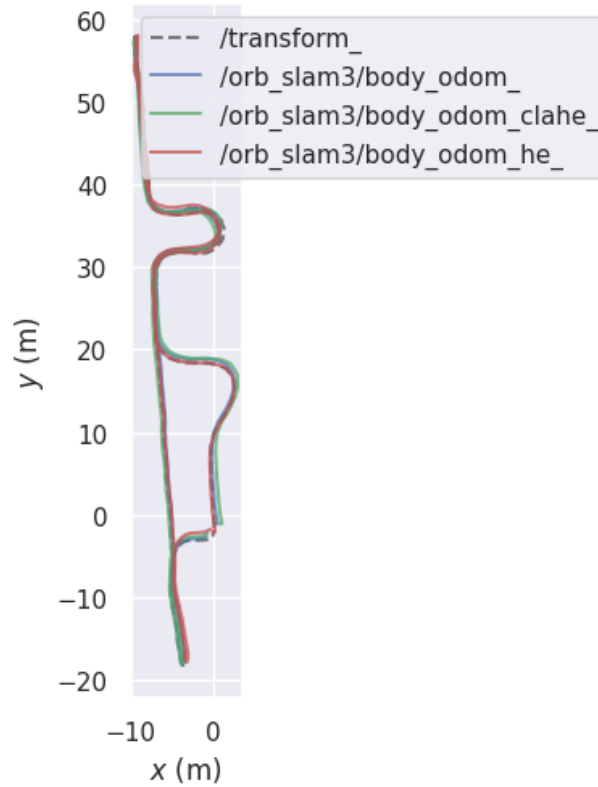
Şekil 4.15 Mühendislik fakültesi kütüphane yörünge çıktıları

Çizelge 4.2 SLAM sonuçları(mühendislik fakültesi kütüphane verisi)

ORB-SLAM3				
Kamera Tipi	Metrik	Ham veri	Histogram Eşitleme	Kontrast Kısıtlımalı UHE
Stereo	APE	0.4877	0.4052	0.4690
	RPE	0.4803	0.3597	0.4070

Çizelge 4.2’de yer alan sonuçlar histogram eşitleme uygulanan veri setinin diğerlerine göre daha iyi sonuçlar çıkardığı görülmüştür. İkinci olarak ise incelendiğinde kontrast uyarlanabilir histogram eşitleme kısıtlımalı iyi sonuç vermiştir.

Şekil 4.16’ de mühendislik fakültesi zemin katın olduğu verilerin ORB-SLAM3 yörünge çıktıları verilmiştir.



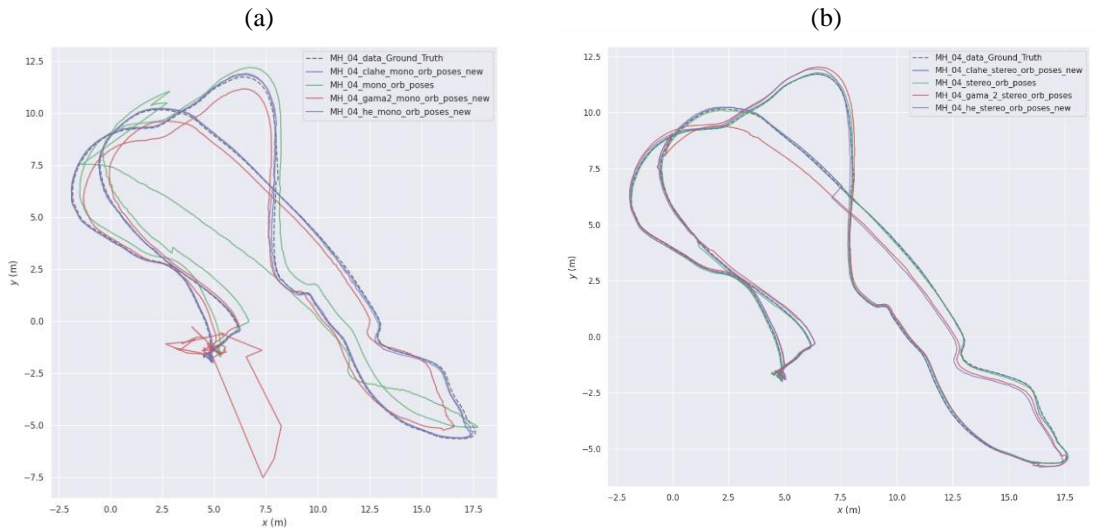
Şekil 4.16 Mühendislik fakültesi zemin kat yörünge çıktıları

Çizelge 4.3 SLAM sonuçları(mühendislik fakültesi zemin kat verisi)

ORB-SLAM3				
Kamera Tipi	Metrik	Ham veri	Histogram Eşitleme	Kontrast Kısıtlamalı UHE
Stereo	APE	1.8116	1.9996	1.7329
	RPE	0.4169	0.4062	0.3818

Çizelge 4.3’de yer alan sonuçlar incelendiğinde kontrast kısıtlamalı histogram eşitleme uygulanan veri setinin diğerlerine göre daha iyi sonuçlar çıkardığı görülmüştür. Ham veriler ile normal HE uygulanmış veriler ise birbirlerine yakın sonuçlar vermiştir.

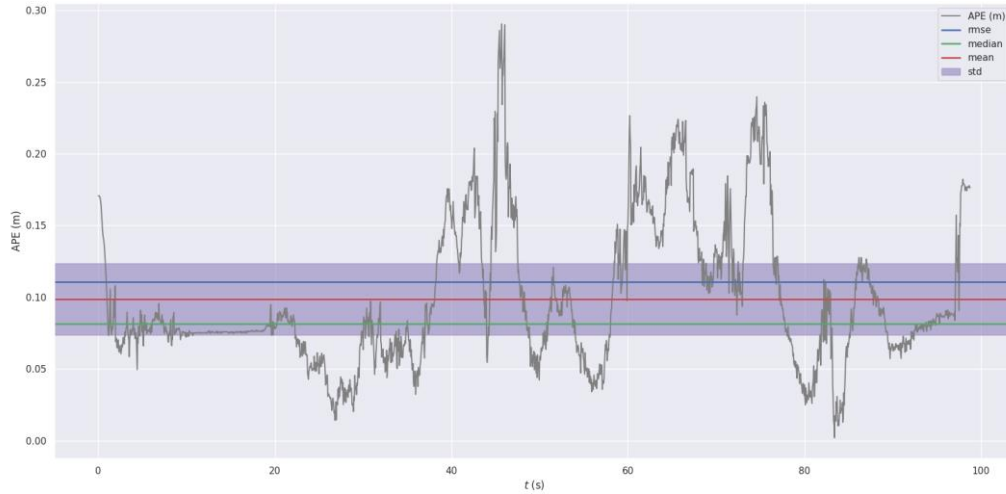
İnsansız hava araçları için üretilmiş euroc açık kaynak veri setinden MH_04 olan veri setinin yörünge sonuçları şekil 4.17’de verilmiştir (Burri vd., 2016). Veri setinde özellikle ışıklandırmanın zayıf olduğu bölümler bulunduğu için deneyle uyumlu olması beklenmektedir.



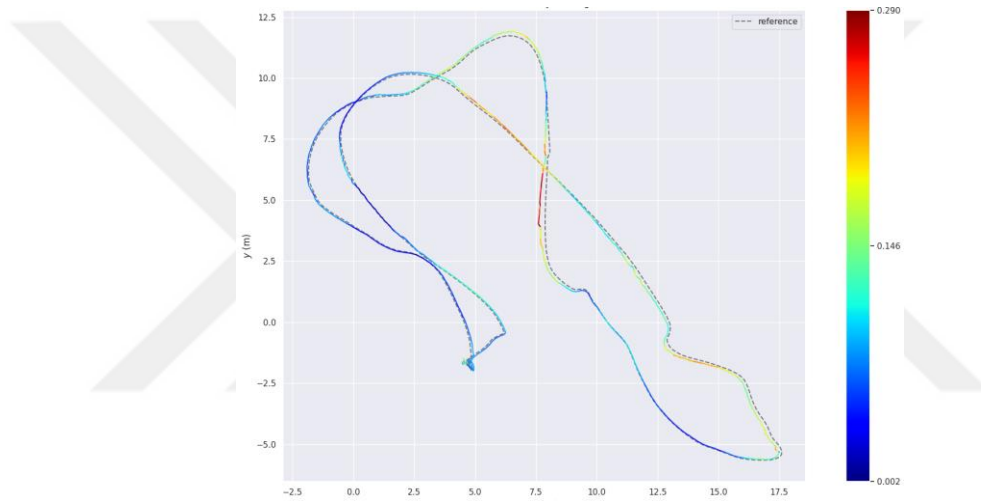
Şekil 4.17 Euroc MH_04_difficult veri seti yörüngeleri (a) monoküler kamera (b) stereo kamera

Çizelge 4.4 SLAM sonuçları(euroc MH_04_difficult veri seti)					
ORB-SLAM3					
Kamera Tipi	Metrik	Ham veri	Histogram Eşitleme	Kontrast Kısıtlı UHE	Gama Düzeltme
Monoküler	APE	1.2683	0.1106	0.0844	1.8234
	RPE	0.1389	0.0721	0.0716	1.3336
Stereo	APE	0.0765	0.2199	0.0608	0.2917
	RPE	0.0751	0.0852	0.0078	0.0852

Çizelge 4.4’de yer alan sonuçlar incelendiğinde hem stereo hem monoküler haritalama yönteminde kontrast kısıtlı histogram eşitleme uygulanan veri setinin diğerlerine göre daha iyi sonuçlar çıkardığı görülmüştür. İkinci en iyi sonuç gösteren monoküler kamera için normal histogram eşitleme uygulanan veri seti, stereo için ise ham veriler daha iyi sonuç vermiştir. Gama düzeltmesi uygulanan verilerde ise daha büyük bozulmalar gözlemlenmiştir. Şekil 4.18 en iyi APE sonucunu veren mono-SLAM yöntemi ayrıntılı gösterimi verilmiştir. Şekil 4.19’da ise yörünge sonucu gösterilmiştir.

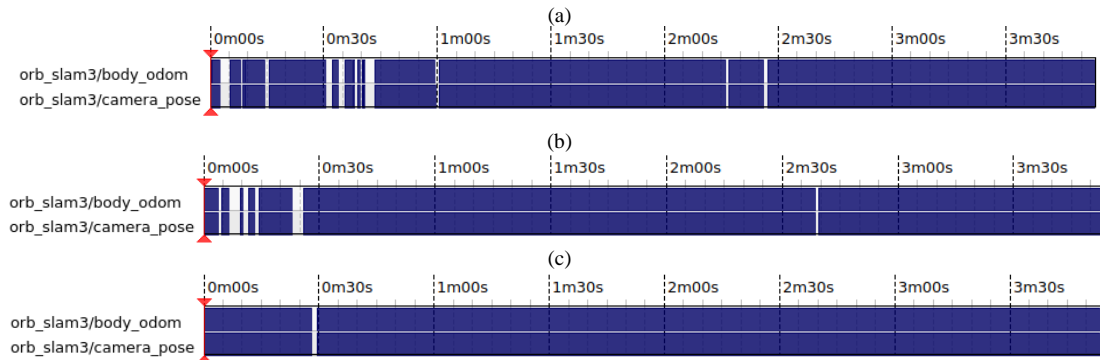


Şekil 4.18 Histogram eşitleme ile çıkarılan mono-SLAM APE sonuçları

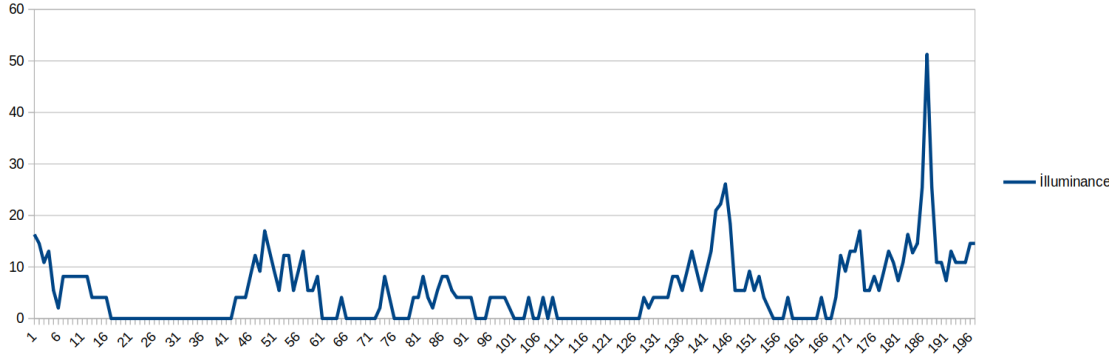


Şekil 4.19 Histogram eşitleme ile çıkarılan mono-SLAM yörüngesi

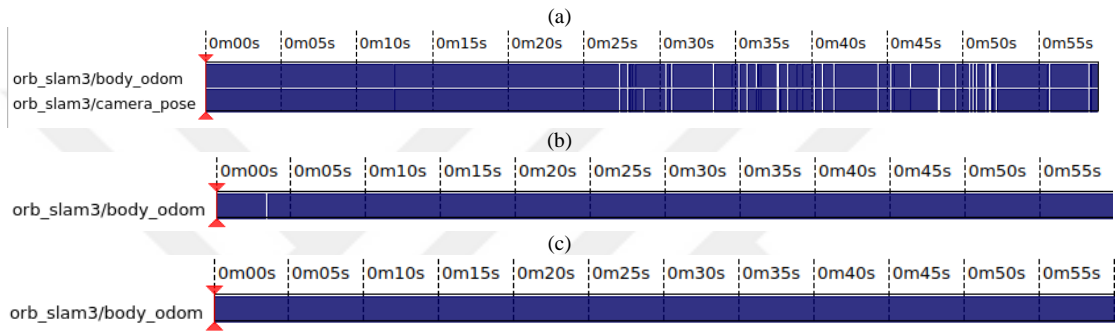
İçerisinde daha karanlık ortam ve ışık patlamalarının olduğu, düz desensiz duvarlar barındıran daha zorlu veri setlerinin bulunduğu ortamdan alınan verilerin ORB-SLAM3 üzerinden elde ettiği sonuçlardaki yaşanan yörünge takibinin kayıp edilmesi şekil 4.20 ve 4.22’de verilmiştir. Şekil 4.21 ve 4.23’te ortamın lux değerleri verilmiştir.



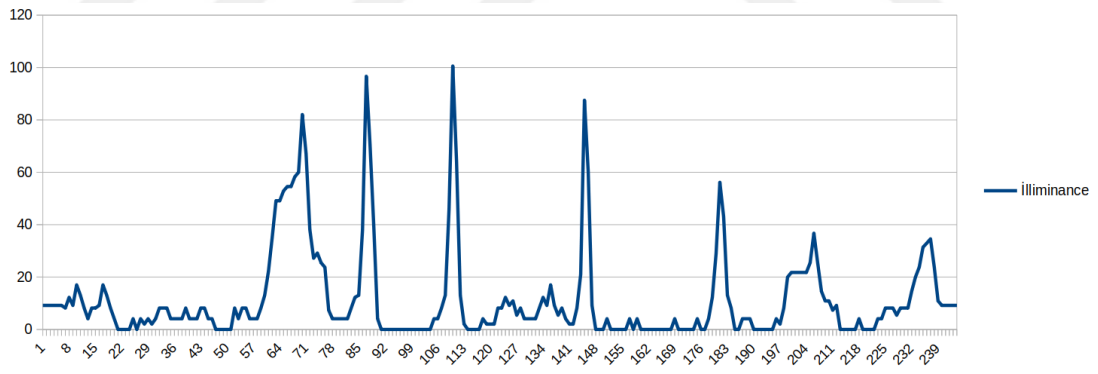
Şekil 4.20 Fakülte C-Blok kat 4 SLAM performansı (a) ham veri, (b) HE uygulanmış, (c)kontrast kısıtlatmalı HE uygulanmış veriler



Şekil 4.21 Fakülte C-Blok kat 4 lux değerleri



Şekil 4.22 Fakülte kat 2 SLAM performansı (a) ham veri, (b) HE uygulanmış, (c) kontrast kısıtlı HE uygulanmış veriler

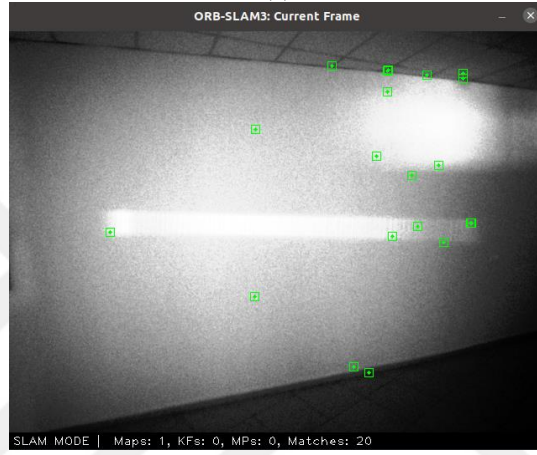


Şekil 4.23 Fakülte C-Blok kat 4 lux değerleri

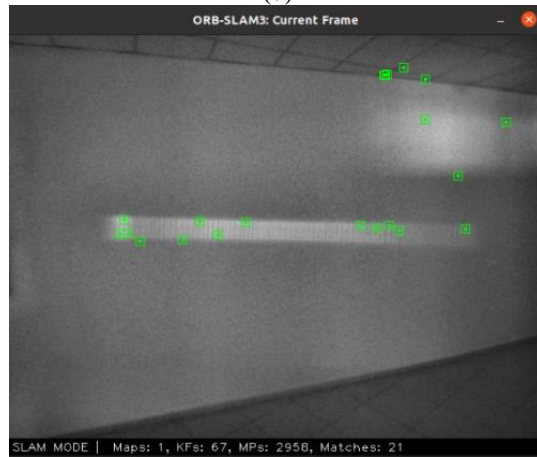
Desensiz, düz duvarlar V-SLAM için özellik çıkarımı yapmayı zorlaştırmaktadır. Şekil 4.24 incelendiğinde ham görüntünün hiçbir özellik noktası çıkaramadığı gözlemlenmiştir. HE uygulanmış görüntülerde özellik noktası çıkartmaya ve sistemin haritalamaya devam ettiği gözlemlenmektedir.



(a)



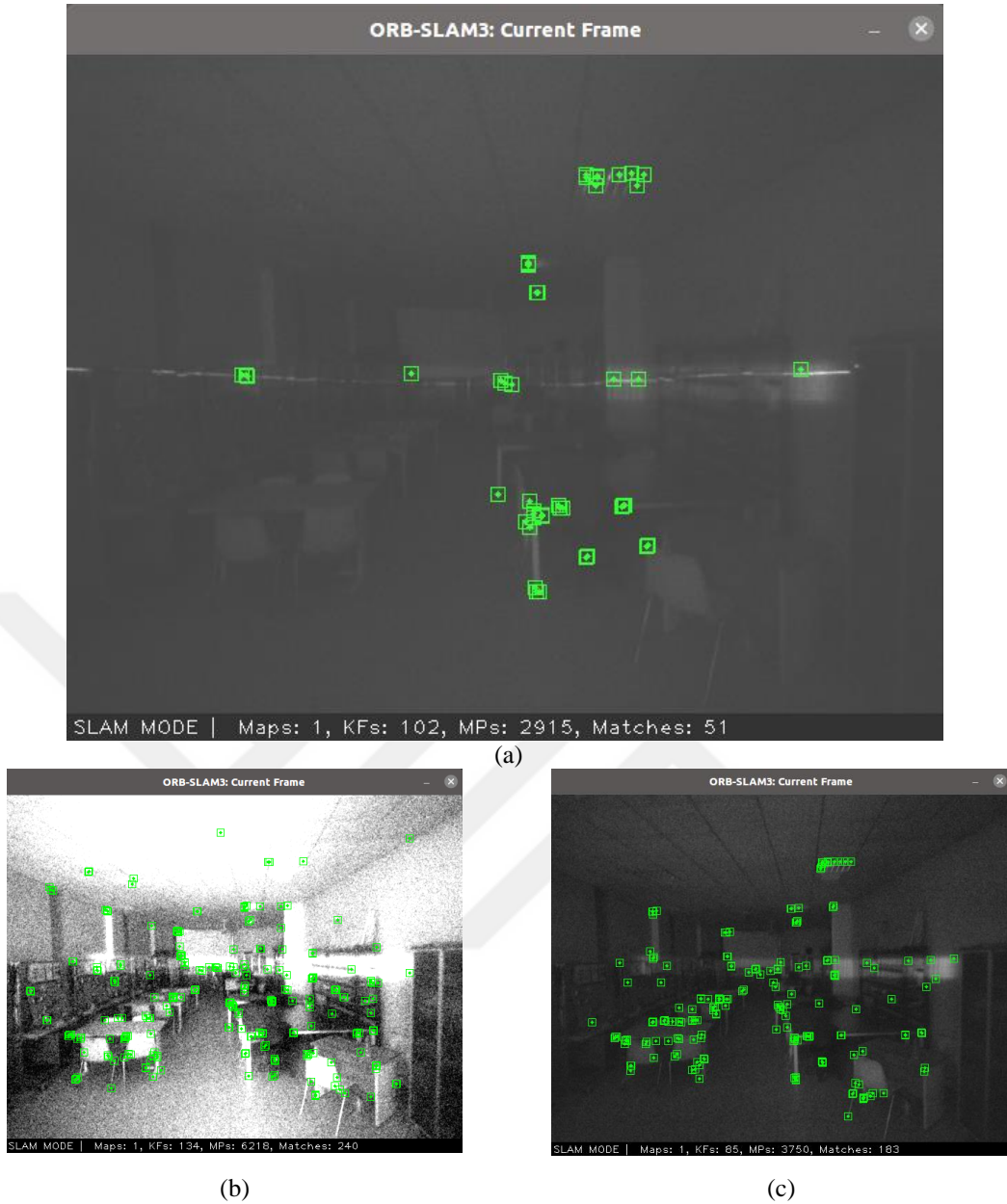
(b)



(c)

Şekil 4.24 ORB-SLAM özellik çıkarımı (a) ham veri, (b) HE, (c) kontrast kısıtlamalı uyarlanabilir HE

Şekil 4.24’de ham görüntünün hiç özellik noktası çıkaramadığı için 0 eşleşme görülürken HE uygulanmış resimlerde 20 civarı eşleme bulunmuştur.



Şekil 4.25 Karanlık ortamda ORB-SLAM özellik çıkarımı (a) ham veri, (b) HE, (c) kontrast kısıtlı HE uygulanabilir HE

Şekil 4.25'e bakıldığında karanlık ortamda ham görüntü oldukça az özellik noktası çıkartırken HE uygulanmış görüntülerde 3-4 kat daha fazla özellik noktalarının çıkartıldığı görülmüştür. Böylelikle sistem haritalamaya devam edip daha sağlam yapıda olmayı sağlamaktadır.

5. SONUÇLAR VE ÖNERİLER

Yapılan tez çalışması kapsamında eş zamanlı haritalama ve konumlandırma problemi için görsel SLAM yöntemi ele alınmıştır. Geleneksel SLAM yöntemlerinden günümüze kadar gelen farklı yaklaşımlardan bahsedilip birbirlerine göre avantajları ve dezavantajları incelenmiştir.

Görüntü işleme tekniklerinin gelişimi ile SLAM problemindeki gelişmeler uzun süredir araştırmacılar tarafından ilgi görmüştür. Derin öğrenme tekniklerinin gelişimi, düşük maliyetli donanımlara erişimin kolaylaşması ve robotiğe artan ilgi ile birlikte görsel SLAM alanında ciddi ilerlemeler kaydedilmiştir. Mobil robotların ile insanlar arasında iş birliği içinde çalışması adına artan yüksek gereksinimler ile robotların gelişmiş çevre algısı ve semantik yöntemler kullanımı giderek önem kazanmıştır.

Gerçek dünya koşulları ele alındığında görsel SLAM performansını etkileyen birçok dış etkenler vardır. Ortam ışıklandırması, çevrede bulunan doku ve yapısal özellikler, robotun hareket hızı ve sistem üzerindeki titreşimler, ani sert hareketler, sık değişen dinamik ortamlar gibi birçok olumsuz etken bulunmaktadır. Bu gibi sorunların üstesinden gelmek için birçok araştırmalara devam edilmektedir.

İyi bir SLAM performansı için görsel SLAM'in sensör verilerini aldığı ön uç kısmında görüntüden özellik çıkarımı önemli bir adımdır. Bu adımda daha çok özellik çıkarımı sistemin devamlılığı ve sağlam olması adına önem arz etmektedir. Bunun için sensör girdilerine yönelik iyileştirme sistemi kararlı olmasına destek olmaktadır. Bu amaçla çalışmada özellikle düşük aydınlatma koşullarındaki bir ortam için görüntü iyileştirilmesine odaklanılmıştır.

Yapılan tez çalışmasında literatürde kendini kanıtlamış algoritmalar çerçevesinde görüntü iyileştirme algoritmaları kullanarak özellik çıkarımı ve SLAM performansına olan etkileri araştırılmıştır. Üzerinde kamera ve lidar sensörü bulunduran mobil platform üzerinden farklı aydınlatma seviyelerinde ve farklı ortamlardan veriler toplanılmıştır. Toplanan ham görüntülere görüntü iyileştirme algoritmalarından gama düzeltme, histogram eşitleme ve kontrast kısıtlamalı uyarlanabilir histogram eşitleme uygulanmıştır. Görüntü ön işleme uygulanan veri çıktıları ORB-SLAM3 algoritmasına aktarılmıştır. SLAM performansının ölçülebilmesi için referans yer gerçekliğini, LIDAR ve IMU sensörleri kullanılarak Cartographer-SLAM üzerinden alınmıştır. ORB-SLAM3 yörünge çıktısı ile Cartographer-SLAM yörünge çıktıları EVO-ROS paketi kullanılarak, grafikler elde edilip, mutlak pozisyon hatası ve göreceli pozisyon hatası

hesaplanılmıştır. Deney sonuçları ele alındığında görüntü üzerinde yapılan iyileştirmeler farklı aydınlatma koşulları ve farklı ortamlarda, farklı sonuçlar verdiği gözlemlenmiştir. Tüm veri setleri ele alındığında kontrast kısıtlamalı uyarlanabilir histogram eşitleme uygulanan görüntülerde en iyi pozisyon hatası ölçüldüğü gözlemlenmiştir. Normal histogram eşitleme uygulanan görüntüler ile ham görüntüler karşılaştırıldığında özellikle düşük ışıklandırılmalı bölgelerde HE daha iyi sonuç verirken nispeten daha iyi aydınlatma olan bölgelerde ham veriler daha iyi sonuç elde etmiştir. Gama düzeltme uygulanan görüntülerde ise daha büyük bozulmalar meydana gelmiştir. Özellik çıkarımı açısından yapılan iyileştirmeler incelendiğinde KKUHE’de en iyi eşleşme ve özellik çıkarımı görülmüştür. HE uygulanan görüntülerde ise özellikle ışık patlamalarının olduğu kısımlarda ham verilere göre daha kötü sonuçlarla karşılaşmıştır. Bazı karanlık ortamlarda gama düzeltme uygulanan görüntülerde algoritma daha fazla özellik noktası çıkartılmasına rağmen, pozisyon hatasının daha fazla olduğu görülmüştür. Özellik noktasının fazla olması doğrudan pozisyon hatasını olumlu yönde etkilemediği bazı ortamlarda yanıltıcı özellik noktaları yörüngeyi daha da olumsuz etkilediği gözlemlenmiştir. Tez kapsamında KKUHE uygulanan görüntülerde ORB-SLAM3 algoritmasının hem özellik noktası çıkarımı ve eşleşmesinde, hem de pozisyon hatalarındaki iyileştirmesi SLAM performansı açısından daha kararlı ve hassas olduğunu göstermektedir.

Görsel SLAM uzun yıllardır gelişimiyle birlikte, robotik ve farklı alanlarda yaygın olarak kullanılmaya başlandı. Bununla birlikte, SLAM çevresel etkenler, karmaşık, dinamik, yapılandırılmamış ve büyük ölçekli ortamlara karşı keşfedilmeyi bekleyen çözümler aranmaktadır. Son yıllara da gelişen derin öğrenme tabanlı sistemler bu konuya el atsa da hala çözümü bekleyen birçok sorunla karşı karşıyadır. Derin öğrenmenin daha da gelişmesiyle birlikte gelecekte, SLAM bu alana kayarak semantik VSLAM’in geliştirilmesi ve uygulanması beklenmektedir.

6. KAYNAKLAR

- Aladem, M., & Rawashdeh, S. A. (2018). Lightweight visual odometry for autonomous mobile robots. *Sensors*, 18(9), 2837.
- Andersone, I. (2019). Heterogeneous map merging: State of the art. *Robotics*, 8(3), 74.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (pp. 404-417). Springer Berlin Heidelberg.
- Bernier, C. Collaborative Robot Series: PR2 from Willow Garage. 2016. Erişim adresi: <https://blog.robotiq.com/bid/65419/Collaborative-Robot-Series-PR2-from-Willow-Garage>
- Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., ... & Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157-1163.
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11* (pp. 778-792). Springer Berlin Heidelberg.
- Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D. (2021). Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6), 1874-1890.
- Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, 10(3), 145-155.
- Chen, W., Shang, G., Ji, A., Zhou, C., Wang, X., Xu, C., ... & Hu, K. (2022). An overview on visual slam: From tradition to semantic. *Remote Sensing*, 14(13), 3010.
- Collins, R., (2007). Lecture 06: Corner Detection. [PDF belgesi]. https://www.cse.psu.edu/~rtc12/CSE486/lecture06_6pp.pdf adresinden alınmıştır.
- Davison. (2003, October). Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision* (pp. 1403-1410). IEEE.
- Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6), 1052-1067.
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2), 99-110.

- Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2013). 3-D mapping with an RGB-D camera. *IEEE transactions on robotics*, 30(1), 177-187.
- Engel, J., Schöps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision* (pp. 834-849). Cham: Springer International Publishing.
- Evans, J., Krishnamurthy, B., Pong, W., Croston, R., Weiman, C., & Engelberger, G. (1989). HelpMate™: A robotic materials transport system. *Robotics and Autonomous Systems*, 5(3), 251-256.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.
- Forster, C., Pizzoli, M., & Scaramuzza, D. (2014, May). SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)* (pp. 15-22). IEEE.
- Fraundorfer, F., & Scaramuzza, D. (2011). Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4), 80-92.
- Gao, X., & Zhang, T. (2021). *Introduction to visual SLAM: from theory to practice*. Springer Nature.
- Grisetti, G., Tipaldi, G. D., Stachniss, C., Burgard, W., & Nardi, D. (2007). Fast and accurate SLAM with Rao-Blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1), 30-38.
- Harris, C., & Stephens, M. (1988). A combined corner and edge detector *Alvey Vision Conference*The Plessey Company Pic (pp. 147-151). United Kingdom.
- Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4), 629-642.
- Kerl, C., Stuckler, J., & Cremers, D. (2015). Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras. In *Proceedings of the IEEE international conference on computer vision* (pp. 2264-2272).
- Klein, G., & Murray, D. (2007, November). Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality* (pp. 225-234). IEEE.
- Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., & Kleiner, A. (2009). On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 27, 387-407.
- Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on robotics and Automation*, 7(3), 376-382.

- Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3), 314-334.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91-110.
- Lu, Y., Xue, Z., Xia, G. S., & Zhang, L. (2018). A survey on vision-based UAV navigation. *Geo-spatial information science*, 21(1), 21-32.
- Macario Barros, A., Michel, M., Moline, Y., Corre, G., & Carrel, F. (2022). A comprehensive survey of visual slam algorithms. *Robotics*, 11(1), 24.
- Molero, R., Scherer, S., Chamberlain, L., & Singh, S. (2011). Navigation and control for micro aerial vehicles in GPS-denied environments.
- Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598.
- Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2003, August). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI* (Vol. 3, No. 2003, pp. 1151-1156).
- Moravec, H. P. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Stanford University.
- Mourikis, A. I., & Roumeliotis, S. I. (2007, April). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation* (pp. 3565-3572). IEEE.
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5), 1147-1163.
- Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5), 1255-1262.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... & Fitzgibbon, A. (2011, October). Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality* (pp. 127-136). Ieee.
- Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011, November). DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision* (pp. 2320-2327). IEEE.

- Nistér, D., Naroditsky, O., & Bergen, J. (2004, June). Visual odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. (Vol. 1, pp. I-I). Ieee.
- Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).
- Rehder, J., Nikolic, J., Schneider, T., Hinzmann, T., & Siegwart, R. (2016, May). Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4304-4311). IEEE.
- Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (pp. 430-443). Springer Berlin Heidelberg.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision (pp. 2564-2571). Ieee.
- Shi, J., & Tomasi, C. (1994). Computer vision and pattern recognition. In Proceedings CVPR (Vol. 94, pp. 593-600).
- Smith, R. C., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4), 56-68.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In 2012 IEEE/RSJ international conference on intelligent robots and systems (pp. 573-580). IEEE.
- Sundram, J., Nguyen, V. D., & Soh, G. S. (2018). Development of a miniature robot for multi-robot occupancy grid mapping. 3 rd Int Conf on Advanced Robotics and Mechatronics.
- Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), 21-71.
- Wyrobek, K. A., Berger, E. H., Van der Loos, H. M., & Salisbury, J. K. (2008, May). Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In 2008 IEEE International Conference on Robotics and Automation (pp. 2165-2170). IEEE.
- Yousif, K., Bab-Hadiashar, A., & Hoseinnezhad, R. (2015). An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4), 289-311.

Zhu, D., Ma, Y., Wang, M., Yang, J., Yin, Y., & Liu, S. (2022). LSO-FastSLAM: A new algorithm to improve the accuracy of localization and mapping for rescue robots. *Sensors*, 22(3), 1297.

