

Otonom Taşıyıcı Araçlar İçin Yüksek Çözünürlüklü Harita Güncelleme

Muhammed Oğuz Taş

**YÜKSEK LİSANS TEZİ**

Elektrik Elektronik Mühendisliği Anabilim Dalı

Ekim 2018

High Definition Map Update For Autonomous Transfer Vehicles

Muhammed Oğuz Taş

**MASTER OF SCIENCE THESIS**

Department of Electrical Electronics Engineering

October 2018

Otonom Taşıyıcı Araçlar İçin Yüksek Çözünürlüklü Harita Güncelleme

Muhammed Oğuz Taş

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Elektrik Elektronik Mühendisliği Anabilim Dalı

Telekomünikasyon-Sinyal İşleme Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Dr. Öğr. Üyesi Hasan Serhan Yavuz

"Bu tez, TÜBİTAK tarafından 116E731 no'lu proje çerçevesinde desteklenmiştir."

Ekim 2018

## ONAY

Elektrik Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Muhammed Ođuz Taş'ın YÜKSEK LİSANS tezi olarak hazırladıđı “Otonom Taşıyıcı Araçlar İçin Yüksek Çözünürlüklü Harita Güncelleme” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiştir.

**Danışman** : Dr. Öğr. Üyesi Hasan Serhan Yavuz

**İkinci Danışman** : Doç. Dr. Ahmet Yazıcı

### **Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Dr. Öğr. Üyesi Hasan Serhan Yavuz

**Üye** : Dr. Öğr. Üyesi Burak Kaleci

**Üye** : Dr. Öğr. Üyesi Emre Kıyak

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN  
Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Dr. Öğr. Üyesi Hasan Serhan Yavuz danışmanlığında hazırlamış olduğum “Otonom Taşıyıcı Araçlar İçin Yüksek Çözünürlüklü Harita Güncelleme” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 23/10/2018

Muhammed Oğuz Taş

İmza

## ÖZET

Otonom robotlar, özel görevleri yerine getirmek için birçok alanda giderek daha fazla kullanılmaktadır. Akıllı fabrikalardaki otonom taşıyıcı araçlar (OTA), endüstri 4.0'ın önemli bileşenlerindendir. OTA'ların uzun vadede başarılı çalışabilmesi için kilit süreç haritalamadır. Bir OTA'nın otonom navigasyonu için geleneksel robotik haritalama yeterli olsa da, mükemmel davranışı gerçekleştirmek ve çevre ile daha iyi etkileşimler sağlamak için, yüksek çözünürlüklü (high definition - HD) haritalar gereklidir. HD haritalar elle oluşturulabilir, ancak ortamdaki nesnelerin yer değiştirmesi veya yeni nesnelerin ortama eklenmesi durumlarında haritaların güncellenmesi gerekmektedir. Bu tez çalışmasında, akıllı fabrikalarda kullanılması öngörülen OTA'lar için, gezinirken ortamdaki nesnelere bilgisayarlı görü teknikleriyle otomatik olarak algılayıp, mevcut haritadaki nesnelere değişiklik olduğunda gerekli bilgileri uygun biçimde güncelleyen bir HD harita güncelleme stratejisi önerilmiştir.

**Anahtar Kelimeler:** gezgin robot, otonom taşıyıcı araç, akıllı fabrika, hd harita, nesne tanıma ve konumlandırma

## SUMMARY

Autonomous robots are widely used in many areas to perform specific tasks. Autonomous transport vehicles (ATVs) in smart factories can be considered as an important component of industry 4.0. Mapping is the key process for a long term successful operation of an ATV. Even though conventional mapping methods are sufficient for autonomous navigation, high definition (HD) maps are required to achieve the excellent behavior of the ATV to have better interactions with the environment. HD maps can be created manually, but they should be updated when some of the objects in the environment are removed or new objects are added to the environment. In this thesis, an HD map update strategy has been proposed for ATVs, which are assumed to be used in smart factories. The methodology is based on visual object detection and localization in the environment by using computer vision techniques during the navigation. Then, it updates the HD map information if object locations in the current map change.

**Keywords:** mobile robot, autonomous transfer vehicles, smart factories, hd map, object detection and localization

## TEŞEKKÜR

Bu çalışmanın konusunun belirlenmesinde ve gerçekleştirilmesinde bana danışmanlık yapan değerli hocalarım Sayın Dr. Öğr. Üyesi Hasan Serhan YAVUZ ve Doç. Dr. Ahmet YAZICI'ya, tez aşamasında bana sabır gösteren ve destek olan eşime, maddi ve manevi desteklerini her zaman hissettiğim kıymetli anneme, babama ve kardeşlerime, bu seviyeye gelmemde emeği olan bütün hocalarıma ve İnovasyon Mühendislik personellerine teşekkür ederim.

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'nun (TÜBİTAK) 116E731 nolu "Akıllı fabrikalar için otonom taşıyıcılar ve gerekli insan-makine ve makine-makine arayüzlerinin geliştirilmesi" projesi tarafından desteklenmiştir.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	vi
<b>SUMMARY</b> .....	vii
<b>TEŞEKKÜR</b> .....	viii
<b>İÇİNDEKİLER</b> .....	ix
<b>ŞEKİLLER DİZİNİ</b> .....	xi
<b>ÇİZELGELER DİZİNİ</b> .....	xiii
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	xiv
<b>1. GİRİŞ VE AMAÇ</b> .....	1
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	4
<b>3. ROBOT İŞLETİM SİSTEMİ</b> .....	10
3.1. ROS Terminolojisi .....	10
3.2. Örnek ROS Uygulaması.....	11
3.3. ROS SLAM Paketleri.....	12
<b>4. MATERYAL VE YÖNTEM</b> .....	13
4.1. SLAM .....	14
4.2. Nesne Tanıma ve Konumlandırma .....	14
4.3. HD Harita.....	15
<b>5. BULGULAR VE TARTIŞMA</b> .....	18
5.1. Otonom Taşıyıcı Araç .....	18
5.2. Kullanılan ROS Paketleri .....	21
5.2.1. rtabmap_ros paketi .....	21
5.2.1.1. <u>Pakette bulunan önemli mesaj tipleri</u> .....	22
5.2.1.2. <u>Parametreler</u> .....	23
5.2.2. find_object_2d paketi.....	24

**İÇİNDEKİLER (devam)****Sayfa**

5.3. Testler .....	25
5.3.1. Test ortamı .....	26
5.3.1.1. <u>Test1: nesnelerin konum doğruluğu ve ortam harita bilgisinin çıkarılması</u> .....	27
5.3.1.2. <u>Test2: ortam harita bilgisinin güncellenmesi</u> .....	29
5.3.2. Örnek fabrika ortamı .....	31
5.3.2.1. <u>Test1: nesnelerin konum doğruluğu ve ortam harita bilgisinin çıkarılması</u> .....	35
5.3.2.2. <u>Test2: ortam harita bilgisinin güncellenmesi</u> .....	38
<b>6. SONUÇ VE ÖNERİLER .....</b>	<b>42</b>
<b>KAYNAKLAR DİZİNİ .....</b>	<b>43</b>

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Harita Katmanları İle Otonomi Arasındaki İlişki .....	2
2.1. Bellek Yönetimi ve Döngü Kapanması Algılama (Labbe ve Michaud, 2011) .....	5
2.2. HD Harita Katmanları (Liu vd., 2017) .....	7
3.1. Örnek ROS Uygulaması .....	11
4.1. Önerilen HD Harita Güncelleme Yöntemi .....	14
4.2. HD Harita Güncelleme Akış Diyagramı .....	16
5.1. Otonom Taşıyıcı Araç (OTA).....	18
5.2. OTA Bağlantı Şeması.....	19
5.3. Microsoft Kinect Sensörü.....	20
5.4. Robot Üzerindeki Sensörler ve İlgili Süreçler.....	21
5.5. Find-Object Uygulama Arayüz Ekranı.....	25
5.6. Örnek Test Ortamı .....	26
5.7. Test Ortamı İçin Eğitilen Nesnelere.....	26
5.8. Çıkarılan Örnek Harita .....	27
5.9. Çıkarılan Haritanın İki Boyutlu Görüntüsü .....	28
5.10. Haritanın Güncellenen Bölümü .....	29
5.11. Nesne Tanımlandığında Robot-İşaret Arasındaki Dönüşüm.....	30
5.12. Sistemin ROS Grafiği .....	30
5.13. Örnek Fabrika Ortamı.....	31
5.14. Fabrika Ortamının GAZEBO Çizimi .....	32
5.15. Eğitilen İşaretler .....	32
5.16. a.Robotun Konumu b.Robot-İşaret Arasındaki Dönüşüm Çerçevesi.....	33
5.17. Ortam Haritasında İşaretlerin Sanal Konumları .....	35
5.18. Çıkarılan Ortam Haritası .....	37
5.19. İşaret Tanımlandığında Robot Kamera Görüntüsü.....	38
5.20. Silinecek İşaret İçin Robotun Konumu .....	39
5.21. Robot ve Sanal İşaret Levhalarının Konumları .....	39

**ŞEKİLLER DİZİNİ (devam)**

5.22. Haritanın Güncellenen Bölümü..... 40



## ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
5.1. RTAB-Map Önemli Mesaj Tipleri .....	22
5.2. Çalışmada Kullanılan RTAB-Map Parametreleri.....	23
5.3. Örnek Test Ortamında Bulunan İşaretlerin Gerçek Konumları.....	27
5.4. Tanımlanan İşaretlerin Konumları.....	28
5.5. Güncelleme Sonrası İşaretlerin Konumları .....	29
5.6. Fabrika Ortamında Bulunan İşaretlerin HD Harita ve GAZEBO Konumları .....	33
5.7. Robot ile Sanal İşaret Çerçeveleri Arasındaki Mesafeler.....	35
5.8. İşaretlerin Beş Farklı Test Sonucunda Elde Edilen Minimum, Maksimum ve Ortalama Konum Değerleri .....	36
5.9. Ortalama Değerlere Göre İşaretlerin Konumlarındaki Hata Oranları .....	36
5.10. Mevcut Konumda Robot-Sanal İşaretler Arasındaki Mesafeler.....	40
5.11. İşaretlerin Güncellenmiş Konumları .....	41

## SİMGELER VE KISALTMALAR DİZİNİ

### Simgeler

$\varphi$	Enlem
$\lambda$	Boylam
$x$	Kartezyen $x$ Konumu
$y$	Kartezyen $y$ Konumu
$v$	Lineer Hız
$\omega$	Açısal Hız

### Kısaltmalar

### Açıklamalar

OTA	Otonom Taşıyıcı Araç
HD	High Definition
3B	3 Boyut
2B	2 Boyut
SLAM	Simultaneous Localization and Mapping
RGB-D	Red, Green, Blue, Depth
RTAB-Map	Real Time Appearance Based Mapping
ROS	Robot Operating System
XML	eXtensible Markup Language
IMU	Inertial Measurement Unit
OSM	Open Street Map
SIFT	Scale Invariant Feature Transform

## 1. GİRİŞ VE AMAÇ

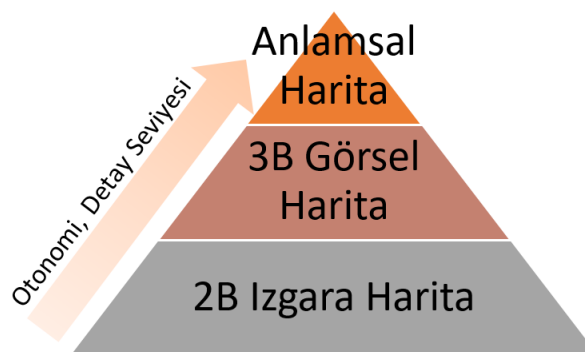
İlk olarak tüm işlevleri uzaktan bir operatör yardımıyla kontrol edilen gezgin robotlar, insanların yapmasının tehlikeli veya zor olduğu birçok uygulamada kullanılmaya başlanmıştır. Bu uygulamaların çeşitliliğinin artmasıyla birlikte, robotların gerçekleştirmesi gereken işlevler ve bu işlevlerin karmaşıklığı da artmıştır. Bunun bir sonucu olarak: sabit mesafeden takip, engel algılama, çarpışma önleme, trafik tıkanıklığını giderme, hız ve pozisyon denetimleri, rota takibi, ortama ait harita bilgisinin elde edilmesi gibi görevlerin yerine getirilebilmesi için birçok yöntem kullanılmaya başlanmıştır. Gelişen teknoloji ile birlikte, gezgin robotlar, birçok alanda yapması gereken görevleri başarılı bir şekilde yerine getirmektedir.

Endüstri 4.0'ın gerçekleştirilmesinde, otonom robotlar, büyük veri, artırılmış gerçeklik, nesnelerin interneti gibi farklı disiplinlerin önemli rolleri vardır (Yazıcı, 2016). Endüstri 4.0 çağında, esnek imalat gerekliliğinden dolayı, akıllı fabrikaların önemli bir rol oynaması beklenmektedir. İçerik farkında yapısına sahip olan akıllı fabrikalar, fiziksel dünyadan gelen nesnenin konumu ve durumu gibi içerik bilgilerini dikkate alarak, sistemin görevlerini yerine getirmesini sağlar (Lucke vd., 2008). Akıllı fabrikalarda, çevreyi algılayan ve misyona göre büyük, dinamik ve bilinmeyen bir ortamda gezinebilen ve klasik otomatik kılavuzlu araçların yerini alacak olan otonom taşıyıcı araç (OTA), malları transfer etmek için kullanılır.

Fabrikalarda bulunan OTA'ların daha güvenilir ve verimli çalışabilmesi için, içinde bulunduğu ortamı tanımlayabilmesi ve kendini bu ortamda konumlandırabilmesi gerekmektedir. OTA'lar tarafından gerçekleştirilmesi öngörülen birçok görevi etkin bir şekilde çözmek için, çevre harita bilgisine ihtiyaç vardır. Haritalama problemi, tam otonom olarak hareket edebilen mobil robotların yapılandırılmasında en önemli aşamalardan biri olarak görülmektedir. İyi bir harita oluşturmak için doğru konumlandırma bir önkoşuldur ve iyi bir konumlandırma için doğru bir haritaya sahip olmak önemlidir (Se vd., 2002). Özellikle, endüstriyel uygulamalarda temel gereksinim olan yüksek konum doğruluğu, OTA'ların uzun vadeli başarılı çalışması için kilit süreçtir.

Robotlar için geleneksel haritalama yöntemleri (Elfes, 1989; Smith vd., 1990) otonom navigasyon için yeterlidir. Ancak, yüksek hassasiyetli, taze ve kapsamlı geometrik bilgiler ve semantiklerle sürüş ortamının önceden oluşturulmuş dijital modeli olan HD haritalar, OTA'ların davranış kalitesini arttırmak ve mükemmel davranış gerçekleştirmek için daha fazla bilgi sağlayabildiklerinden, fabrikalardaki otonom davranışların gerçekleşmesi için kritik bir öneme sahiptir. Örneğin, konum bilgisini kaybetmesi durumunda, OTA park işaretini görsel olarak algılayıp, bu görseli kullanarak kendi konumunu bulabilir ve hassas şekilde park edebilir. Bu nedenle, bu tür çevre bilgileri OTA'ların sensör tabanlı kontrolü için kritik öneme sahiptir.

HD haritalar, normal haritalardan farklı olarak, santimetre seviyesinde yüksek hassasiyete, şerit/yol modelini, trafik ışık ve levhalarının üç boyutlu (3B) konumlarını ve diğer ortam elemanlarının geometrisini ve semantiğini içeren zengin bir bilgiye sahip olup, ortamın dinamikliğinden dolayı sürekli güncel veri barındırmak zorundadır (Jiao, 2018). HD harita sağladığı avantajlar yanında, ortamdan elde edilecek büyük verinin depolama çözümü ve elde edilen bu verilerin yüksek işlem hacmine sahip cihazlar tarafından işlenmesine ihtiyaç duyar. Ayrıca, gerçek zamanlı olarak güncellemeler için, hızlı veri transfer işlemi yapılması gerekirken, gecikme süresi oldukça düşük olmalıdır (Seif ve Hu, 2016). Şekil 1.1'de, harita katmanları ile otonomi arasındaki ilişkiyi gösteren bir görsel verilmiştir. Harita katmanlarında yukarıya doğru çıkıldıkça, ortama ait detay seviyesi artmakta ve robot hareketi yüksek seviye otonomiye yaklaşmaktadır. Robotun gezinmesi için temel olarak iki boyutlu (2B) harita yeterken, davranışın kalitesini ve güvenilirliğini artırmak adına, ortama ait detay içeriklerin arttığı haritalar kullanılmalıdır.



Şekil 1.1. Harita Katmanları ile Otonomi Arasındaki İlişki

Endüstri 4.0 içeriğinde esnek imalatın gerekliliğinden dolayı, geleceğin fabrikalarının dinamik olması beklenmekte ve fabrikada bulunan nesnelere ait özelliklerin, çalışma zamanında güvenilir ve verimli bir şekilde algılayıcılar tarafından algılanması gerekmektedir. Öngörülemeyen değişiklikler nedeniyle, ortamda bulunan nesnelere konumları değişebilmekte veya yeni nesnelere ortama ilave edilebilmektedir. Robotun ortamda güvenle gezinebilmesi için, ortamda bulunan değişiklikler anında algılanmalı ve haritaya aktarılmalıdır. Bu yüzden, otonom davranışın başarısı için, ortam harita bilgisi sürekli olarak güncellenmelidir.

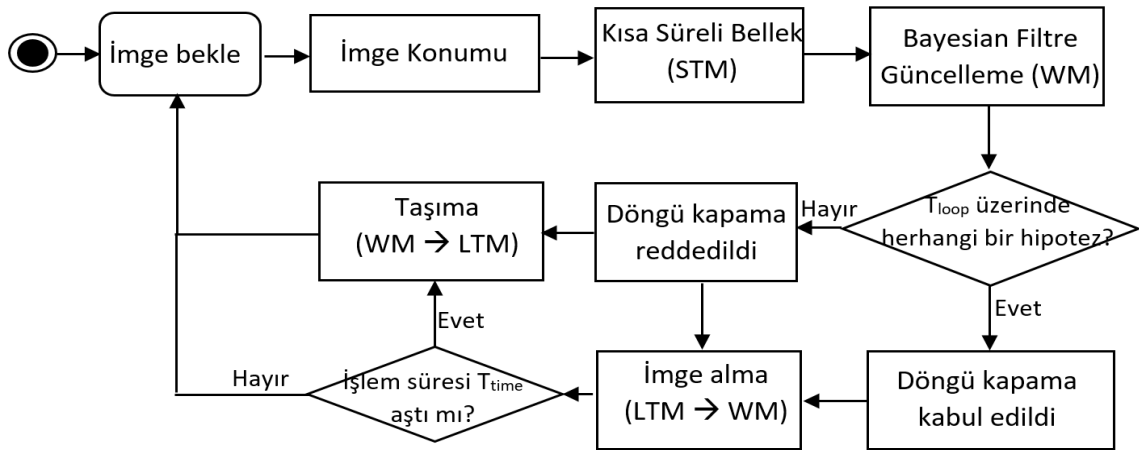
Bu çalışmada, fabrika ortamlarında kullanılması düşünülen bir OTA modeli için, bilgisayarlı görü teknikleri kullanılarak nesne tanıma ve konumlandırma uygulaması gerçekleştirilmiş ve bu uygulamanın çıktıları HD harita güncellemede kullanılmıştır. Gerçekleştirilen uygulama, OTA üzerindeki görsel ve lazer sensörlerden elde edilen bilgilerin işlenerek anlık harita çıkarımı ve çıkarılan harita ile mevcut HD haritadaki nesne bilgilerinin karşılaştırılarak, gerekli ise güncelleme yapılmasını içermektedir. Lazer sensörlerden elde edilen verilerle ortamdan çıkarılan iki boyutlu ızgara haritası, kamera verileri kullanılarak eş zamanlı konumlandırma ile elde edilen harita ile birleştirilerek üç boyutlu haritalama elde edilmekte, bir yandan da yine görsel veri üzerinde nesne tanıma yapılarak, ortamdaki nesnelere haritadaki konumları belirlenmektedir. Belirlenen nesnelere ve nesne konumları mevcut HD haritadaki bilgilerle karşılaştırılarak, güncelleme ihtiyacı olup olmadığı kontrol edilmektedir. Ortamda yeni bir nesne tespit edilmesi durumunda veya ortamda var olması gereken bir nesnenin ortamda tespit edilememesi durumlarında HD harita güncellenmektedir. Gerçekleştirilen uygulamada, nesne olarak önemli olduğu düşünülen yönlendirme işaretleri ve iş güvenliği tabelaları kullanılmıştır. Deneysel çalışmada tasarlanan senaryolar altında, harita güncellemesi başarıyla gerçekleştirilmiştir. Bir sonraki bölümde konu ile ilgili literatür araştırması verilmektedir. Üçüncü bölümde, robot işletim sistemi anlatılmaktadır. Dördüncü bölümde, tez çalışmasında kullanılan yöntem detaylarıyla verilmektedir. Beşinci bölümde ise yapılan deneysel çalışmalar tanıtılmakta ve elde edilen bulgular tartışılmaktadır. Altıncı bölümde, sonuç ve öneriler özetlenmektedir.

## 2. LİTERATÜR ARAŞTIRMASI

Robotun bilinmeyen bir ortamda güvenli bir şekilde hareket edebilmesi için ortamı önceden tanıması gerekmektedir. Robotun bir ortamda gezinmesi için en temel öge, kendini konumlandıracağı, hareket planını çıkararak güvenle gizeceği bir haritanın varlığıdır (Yu ve Amigoni, 2014). Robotik haritalama, mobil ortamlardan mekânsal veya fiziksel ortam modelleri elde etme problemini ele alır (Thrun, 2002). Leonard ve Durrant-Whyte (1991) tarafından türetilen eş zamanlı konumlandırma ve haritalama (Simultaneous Localization and Mapping - SLAM) kavramı, robotik haritalama ile eş kavram olarak kullanılmaya başlamıştır. SLAM ile amaçlanan, robotun bulunduğu noktaya ait pozisyon ve mutlak konum bilgilerinin elde edilmesi (lokalizasyon), içinde bulunduğu ortam haritasının çıkarılması, bu haritanın sürekli güncellenmesi (haritalama) ve ortam içinde bir noktadan diğer bir noktaya gidilebilmesi için gerekli yolun planlanarak izlenmesidir (gezinme) (Risgaard ve Blas, 2005). Her sensör tipinin karakteristiğinden dolayı SLAM, görüntü sensörleri ve mesafe sensörleri için farklı algoritmalarla gerçekleştirilebilir. Küçük ve statik alanlarda SLAM problemini çözmek için, 2B haritalar mesafe sensörleri yardımıyla oluşturulur (Thrun vd., 2000; Wolf ve Sukhatme, 2005; Grisetti vd., 2007; Min vd., 1997). Fakat dinamik, karmaşık ve büyük alanlarda, SLAM algoritmalarında görsel SLAM için kameralar kullanılır (Se vd., 2001; Davison, 2003; Murray ve Jennings, 1997; Audars vd., 2011; Sturm vd., 2012). Görsel SLAM görüntü tanımlama, görüntü alma, görüntü bulma, özellikleri eşleştirme gibi birçok bilgisayarlı görü tekniğini içinde barındırır. Robotik haritalama ve SLAM ile alakalı detaylı bilgiler sırasıyla Thrun (2002) ve Freese (2006) çalışmalarında verilmiştir.

Gerçek zamanlı görünüm tabanlı haritalama (Real Time Appearance Based Mapping - RTAB-Map) kavramı (Labbe (2011a)), RGB-D kamera ve 3B bilgi sağlayan herhangi bir sensörün kullanıldığı döngü kapama algılamasına dayanan bir görsel SLAM yaklaşımıdır. Döngü kapama algılama, SLAM'daki mevcut ve daha önce ziyaret edilen konumlar arasında bir eşleşme bulmaya çalışırken kat edilen süreçtir. Labbe ve Michaud (2011, 2014)'te verilen algoritma ile, büyük ölçekli ve gerçek zamanlı SLAM için bir bellek yönetim mekanizması sunulmuştur. Bu bellek yönetimi stratejisi, robotun çalışma belleğinde (WM) en yeni ve sık

gözlemlenen parçaları tutar ve diğerlerini uzun süreli belleğe (LTM) aktarır. Geçerli konum ile WM'de saklanan konum arasında bir eşleşme bulunursa, LTM'de depolanan ilişkili yerler hatırlanabilir ve güncellenebilir. Çalışmalar, döngü kapama ile hafıza yönetimi sistemleri problemlerini çözmüş ve grafik tabanlı SLAM ile büyük ölçekli, uzun vadeli ve çok katmanlı ortamlar için çevrimiçi gereksinimleri karşılamıştır. Sistemin akış diyagramı Şekil 2.1'de verilmiştir.



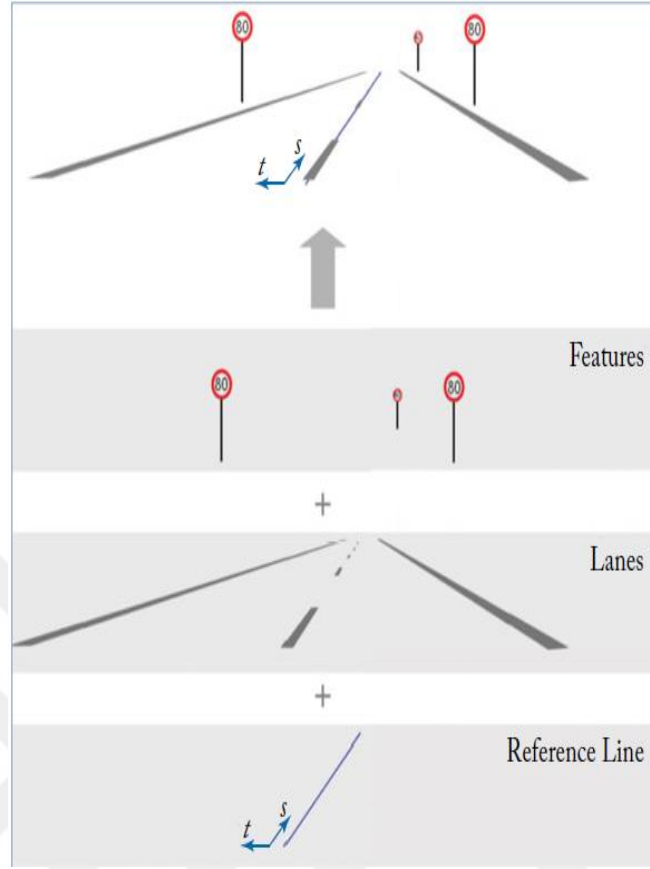
Şekil 2.1. Bellek Yönetimi ve Döngü Kapanması Algılama (Labbe ve Michaud, 2011)

Haritanın temel yapısı, esasen düğümler ve bağlantılar içeren bir grafikdir. Düğümler haritadaki her konum için odometri pozlarını saklar. Düğümlerde ayrıca, lazer taramaları, RGB görüntüleri, derinlik görüntüleri ve döngü kapatma algılaması için kullanılan görsel kelimeler gibi bilgiler de bulunur. Bağlantılar, düğümler arasındaki geometrik dönüşümleri saklar. Literatürde RTAB-Map ile alakalı çeşitli çalışmalar (Fernandes ve Alexandre, 2016; Sankrit vd., 2016; Da Silva vd., 2017) mevcuttur. Bu çalışmalardan, Fernandes ve Alexandre (2016), içerisinde RTAB-Map'in de bulunduğu üç SLAM yöntemini paralel çalıştırarak, robotun çevresini daha iyi algılayabilmesini sağlayan sağlam bir SLAM önerisi sunmuştur. Da Silva vd. (2017), içlerinde RTAB-Map'in de bulunduğu ROS ile uyumlu beş farklı SLAM paketinin incelemesini vermiştir. Sankrit vd. (2016), Kinect sensörünü kullanarak, ortamın haritasının RTAB-Map ile elde edilmesini gerçekleştirmiştir.

Otonom sürüş, navigasyon sistemi, yüksek seviye kontrol, düşük seviye kontrol gibi birçok teknolojiyi bir araya getiren karmaşık bir sistemdir. Otonom sürüş amaçları için özel olarak hazırlanmış olan haritalar, yüksek çözünürlüklü (High Definition-HD) haritalar

olarak adlandırılır (Vardhan, 2017). HD haritalama, otonom sürüşün kilometre taşıdır ve üst seviye kontrol ile navigasyon sisteminin en kritik parçasıdır. Bu haritalar yardımıyla, robotlar çok hassas hareket etmekte ve üç boyutlu uzayda kendi konumlarını hassasiyetle bulmaktadırlar. HD harita üretimi, ham veri işleme, nokta bulutu üretimi, nokta bulutu hizalama, 2B yansıma haritası oluşturma, HD harita etiketleme ve nihai harita oluşturma gibi birçok aşamayı içeren karmaşık bir süreçtir (Liu vd., 2017).

Bilinç düzeyi kararları, dijital haritalardan toplanan çevre bilgilerine bağlıdır. Bu nedenle, dijital haritaların sürekli güncellenmesi otonom davranışın başarısı için önemlidir. HD haritalar, temel katman ve değişik seviyelerde semantik katmanlardan meydana gelmekte ve fabrikalarda bulunan OTA'lar için gereken tüm geometrik ve semantik bilgileri içermektedir. Temel katman, 3B lazer tarayıcıdan çıkarılan yol yüzeyinin, 2B düzlemsel görünümünü ızgara harita olarak içermektedir. Temel katmanın üstünde, bilgi seviyelerine göre değişik semantik katmanlar vardır. Klasik haritadan farklı olarak, HD haritalar, haritaya yol şekli, eğim, eğrilik, yön gibi yol özelliklerini, şerit tipi, şerit genişliği gibi şerit özelliklerini, park alanı, yaya yolu gibi bölge özelliklerini ve ortamda bulunan diğer nesnelere ait özellikleri ekler. Ayrıntı düzeyi arttıkça, robotun haritadaki konum doğruluk düzeyi de artmaktadır. Şekil 2.2'de görüldüğü gibi, değişik seviyedeki semantik katmanlar yol işaretinin ve şeritlerin konum bilgisini içerir. Ayrıca bu işaretlerin karakteristikleri ve karşılık gelen tanımlamaları da mevcuttur. Semantik katmanlar sayesinde, otonom araçlar ortama ait tüm bilgilerin içerildiği harita ile daha yüksek hassasiyette haritadaki konumlarını bilmektedirler.



Şekil 2.2. HD Harita Katmanları (Liu vd., 2017)

Otonom aracın yerini ve otonom araç ile çevresindeki diğer nesnelere arasındaki ilişkiyi aynı anda belirleyebildiği için, SLAM tekniği HD haritalarla otonom sürüş için en çok çalışılan stratejilerden biridir (Seif ve Hu, 2016). Jiao (2018) çalışmasında, HD haritalara ait özelliklere ve katman yapısına ait detaylı bilgiler vermiştir. Bu çalışma aynı zamanda, HD harita oluşturma adımlarını anlatmakta ve harita oluşturma sürecinde elle yapılan iş miktarını en aza indirmek için kullanılan makine öğrenme tekniklerinin detaylı incelemesini vermektedir. Siam vd. (2017) HD haritaları iki kategoriye ayırmıştır. İlki, semantik bilimle tüm sahnenin 3B nokta bulutu ile modellendiği yoğun semantik nokta bulut haritalarıdır. İkincisi, genel 3B nokta bulutları yerine, semantik nesnelere dayanan landmark tabanlı haritalardır. Massow vd. (2016), yakın gelecekte birçok HD harita formatının çıkacağını öngörmektedirler. Zhao vd. (2018) HD haritanın aktif katman (yol, şerit, ve diğer nesne bilgileri), dinamik katman (diğer araçlardan gelen verilerle gerçek zamanlı güncelleme) ve analiz katmanı (büyük verilerin analiz edilmesi ve aracın güvenli sürüşü) olarak 3 katmana ayrıldığını ifade etmiştir.

Zheng ve Wang'in (2017) çalışmasında, sadece lazer verisi kullanılarak, HD harita ile yüksek hassasiyetli ve güvenilir bir lokalizasyon sistemi tasarlama ve geometrinin etkisinin araştırılması amaçlanmaktadır. Busch vd. (2018), gelecekteki otomobillerin HD haritaların güncelliğini korumak için kullanabilecek sensörleri taşıyacak olması fikrini temel almışlardır. Sürekli taramalar gerçekleştirmek üzere, kavşak noktalarına statik 3B lazer tarayıcılar yerleştirilmiştir. Nesnelere takip edilerek, yörüngeleri elde edilmiştir. He vd. (2018) çalışmasında, çok ışınlı LİDAR sensörünün veri kaynağı olarak kullanıldığı, yol yapılarının haritalanması için vektör tabanlı bir yöntem önermiştir. Bu çalışmada sadece yol yapısı üstünde çalışılmış, şeritler gibi diğer özellikler gelecek çalışmalara bırakılmıştır. Ma vd. (2017), yol yüzeyinin düzgün olduğu yerlerde mobil lazer tarayıcı nokta bulutlarından gelen verilerle yol işaretlerini otomatik olarak çıkarmak için bir yöntem sunmaktadır. Jang vd. (2017) çalışmasında, otonom araç ile trafik ışıkları arasındaki göreceli konumdan aracın konumunu elde etmiştir. Bu çalışma, yolun eğimini göz önünde bulundurarak hesaplamayı gerçekleştirmiş, trafik ışığını bularak, yerini tespit etmiş ve uzaklık bilgisini hesaplamıştır. HD harita üretirken, sadece trafik ışıklarının konumlarını değil, aynı zamanda bunların türlerini de belirlemiştir. Böylece, bir trafik ışığının konumu değişse veya yeni bir trafik ışığı çevreye eklense bile, HD harita kolayca güncellenebilmektedir. Seff ve Xiao (2016) çalışmasında, yol plan özelliklerinin (kavşaklara uzaklıklar, trafik yönelimi gibi) araç üzerinde bulunan kameradan alınan RGB sokak görüntüsünden tahmin edildiği bir model geliştirmeyi amaçlamıştır. Mátyus vd. (2016), yolların oldukça detaylı segmentasyonunu, derin öğrenme ile hem hava hem de yer görüntüleri birlikte kullanarak anlamlandırmıştır. Burada, OSM'deki her yol segmentinde bulunan şeritlerin sayısı ve yerleri, park yerleri ve diğer özelliklerden faydalanılmıştır. Bauer vd. (2016), genel GNSS konumlandırma algoritmasının geliştirilmesi amacıyla, algoritmada bir şerit işaretleme bulma yönteminin entegrasyonunu önermiştir. Bu amaçla, test aracı üzerindeki ön kameradan algılanan şerit işaretleyicileri, harita içindeki konumları, türleri (sürekli veya kesikli şerit işaretleyici) ve yoldaki şeritlerin sayısı da dahil olmak üzere, HD haritalarda bulunan bilgileri kullanarak, parçacıkların konumunun düzeltilmesine yardımcı olmaktadır. Bittel vd.nin (2017) çalışmasında önerilen ana fikir, HD harita parametrelerini, HD haritaları doğrulamak veya lazer tarayıcı ve kameralardan elde edilen verilere dayanarak lokalizasyonu geliştirmek için kullanılacak sensör verisinden tahmin etmektir. Tahmin edilen parametreler arasında, ego-aracıdan karayolu sınırına olan mesafe, ego-aracın şeritlere göre oryantasyonu, şerit

sayısı ve sokak tipi yer alır. Li vd. (2017), dış duyarlı sensörlerden gelen ek bilgileri kullanarak, örneğin kameradan şerit tanıma yaparak, konum belirsizliklerini gidermişlerdir.

Sık değişen endüstriyel ortamlarda başlangıçta çizgi takip sistemleri ile kendini konumlandıran ve gezinen araçlar, fabrikaların esnekliğinin artması ile farklı stratejilerin ortaya çıkmasına ihtiyaç duymuştur. Fabrikalarda esnek sistemler sağlamak ve çevrenin olası değişikliklerine uyum sağlayarak güvenle gezecek araçlar tasarlamak için, ortam haritasının çıkarılması oldukça önemlidir. Fabrikalarda haritalama ve SLAM ile alakalı çalışmalar mevcuttur (Vasilijević vd., 2016; Martínez-Barberá ve Herrero-Pérez, 2010; Jung ve Song, 2016). Bu çalışmalardan, Vasilijević vd. (2016), yüksek doğrulukta konumlandırma için haritalamadan faydalanmış, Martínez-Barberá ve Herrero-Pérez (2010), kısmen yapılandırılmış endüstriyel ortamda faaliyet gösteren AGV sistemleri için haritalama stratejisi öne sürmüştür, Jung ve Song (2016), fabrika ortamları için grafik tabanlı bir SLAM yaklaşımını önermiştir. Fabrikalarda haritalama ve konumlandırma ile ilgili çalışmalar bulunsa da, bilindiği kadarıyla HD haritanın fabrika ortamlarında görev yapan OTA'lar üzerinde bir uygulaması gerçekleştirilmemiştir.

Gerçekleştirilen tez çalışmasının bir bölümü, Akıllı Fabrikalarda Bulunan Otonom Taşıyıcı Araçlar İçin HD Harita Güncelleme (Updating HD Map For Autonomous Transfer Vehicles in Smart Factories) başlığıyla, 6. Uluslararası Kontrol Mühendisliği ve Bilgi Teknolojileri Konferansı'nda (International Conference on Control Engineering and Information Technology – CEIT, 2018) sözlü olarak sunulmaya değer görülmüştür.

### 3. ROBOT İŞLETİM SİSTEMİ

Robot İşletim Sistemi (Robot Operating System - ROS), yazılım geliştiricilere robot uygulamaları geliştirmeleri için gerekli kütüphaneleri, araçları, donanım soyutlamasını, aygıt sürücülerini, görselleştiricileri, mesaj iletimi, paket yönetimi vb. sağlar. ROS, açık kaynak bir lisans olan BSD Lisansı ile lisanslanmıştır (Anonim, 2009). ROS sanıldığı gibi aksine işletim sistemi veya programlama dili değil, robotlar için işletim sistemi benzeri işlevler sağlayan, açık kaynak kodlu bir yazılım sistemidir. ROS'un değişik sürümleri bulunmaktadır.

ROS, Gazebo ve Rviz ortamlarıyla tam uyumlu çalışması sayesinde, çalışmaların benzetim ortamında test edilebilmesini sağlar. ROS içinde barındırdığı standart kütüphanelerin ve desteklediği cihazların dışında, geliştiriciler tarafından yazılmış kütüphanelere ve sürücülere de destek verdiği için her geçen gün kapsamını arttırmakta ve robot dünyasında standart haline gelmektedir. Ayrıca, ROS dil bağımsız bir yapıya sahip olduğu için, istenirse, aynı robot üzerinde Python, C, C++ gibi birçok popüler dille kodlama yapılabilir.

#### 3.1. ROS Terminolojisi

ROS'a ait birçok terim bulunmaktadır. Bu terimlerden en sık kullanılanları aşağıda özetlenmiştir.

*ROS Master:* Bağlantıların iletişimini sağladığı ağ tabanlı sunucudur. Sabit konfigürasyon değerlerini kaydeder.

*Paket:* ROS'a bağlanmış yazılım koleksiyonlarıdır. Bir paket ROS düğümleri, kütüphaneler, veri kümesi, yapılandırma dosyaları, üçüncü parti bir yazılım parçası veya mantıksal olarak kullanışlı bir modül oluşturan başka herhangi bir şey içerebilir.

*Yığın:* Ufak parçaların bir araya gelerek oluşturduğu dağıtıma verilen isimdir. ROS'taki paketler ROS yığınlarına ayrılmıştır.

*Düğüm:* Fonksiyonları yerine getiren yapılardır. Düğüm isimleri benzersizdir.

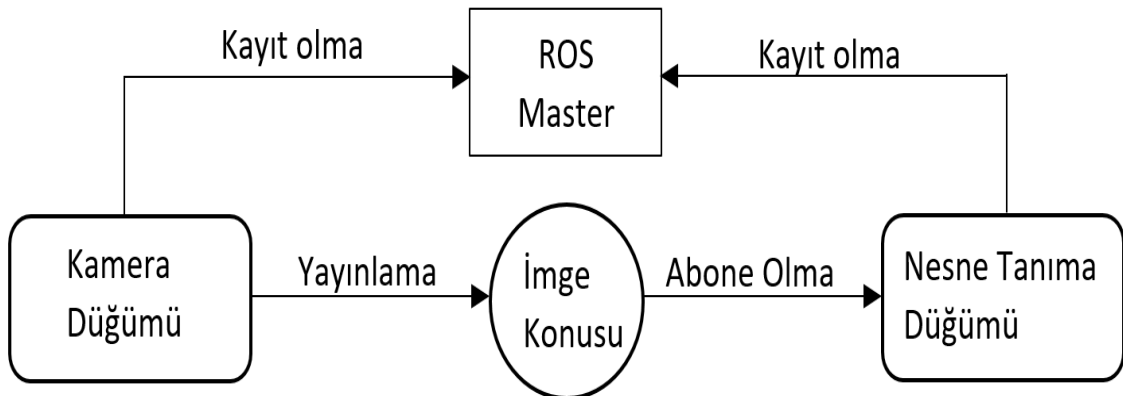
*Konu:* İçerisinde güçlü mesaj tipleri barındıran, geri çağırma fonksiyonlarına sahip bir yapıdır. İletişim için yayınlama ve abone olma yapısını kullanır.

*Servis:* Standart fonksiyonlar gibi, eşzamanlı olarak kullanılır. Birden fazla istemci olabilir.

*Mesaj:* Düğüm iletişimi mesaj gönderimi üzerine kuruludur. Mesajlar, özel tipli alanlardan oluşan bir veri yapısından ibarettir. ROS içerisinde birçok hazır mesaj standardı vardır ve özel olarak buna yenileri eklenebilir.

### 3.2. Örnek ROS Uygulaması

ROS sistemi, ağ tabanlı “ROS Master” sunucusu üzerinden düğümlerin yayın yapma ve abone olma haberleşme yöntemini kullanır. Düğümlerde sistem için gerekli işlemler yaptırılarak, elde edilen veriler konular yardımıyla yayınlanır ve başka bir düğüm, bu yayınlanan verileri konulara abone olarak kullanır. Şekil 3.1’de verilen örnekte, kamera düğümü imge konusu yardımıyla veri aktarımı yapmakta ve nesne bulma düğümü bu konu üstünden gelen veriyi kullanarak, nesneyi tanımlamaktadır.



Şekil 3.1. Örnek bir ROS Uygulaması

### 3.3. ROS SLAM Paketleri

ROS'un robotikte ara katman standardı olarak kabul edilmesinden sonra, SLAM gibi problemler için herhangi bir robotla uyumlu ROS paketleri geliştirilmeye ve açık kaynak olarak sunulmaya başlamıştır. Aşağıda ROS ile uyumlu birkaç SLAM paketinden bahsedilmiştir.

*Gmapping*: Lazer tabanlı SLAM sağlar. Lazerdan iki boyutlu doluluk ızgara haritası elde edilebilmektedir.

*Hector SLAM*: Odometri kullanmayan platformlarda kullanılabilen bir SLAM yaklaşımıdır. Sistem açık döngü kapanma yeteneği sağlamazken, birçok gerçek dünya senaryosu için yeterince doğrudur.

*ORB SLAM*: Çeşitli ortamlarda kamera yörüngesini ve sahnenin seyrek 3B yeniden yapılandırmasını gerçek zamanlı olarak hesaplayabilen çok yönlü ve doğru bir monoküler SLAM çözümdür. Döngüleri algılayabilir ve kamerayı gerçek zamanlı olarak yeniden konumlandırabilir.

*ORB SLAM 2*: Monoküler, stereo ve RGB-D kameralar için kamera yörüngesini ve seyrek 3B yeniden yapılandırılmasını gerçek zamanlı hesaplayan bir SLAM kütüphanesidir. Döngüleri algılayabilir ve kamerayı gerçek zamanlı olarak yeniden konumlandırabilir.

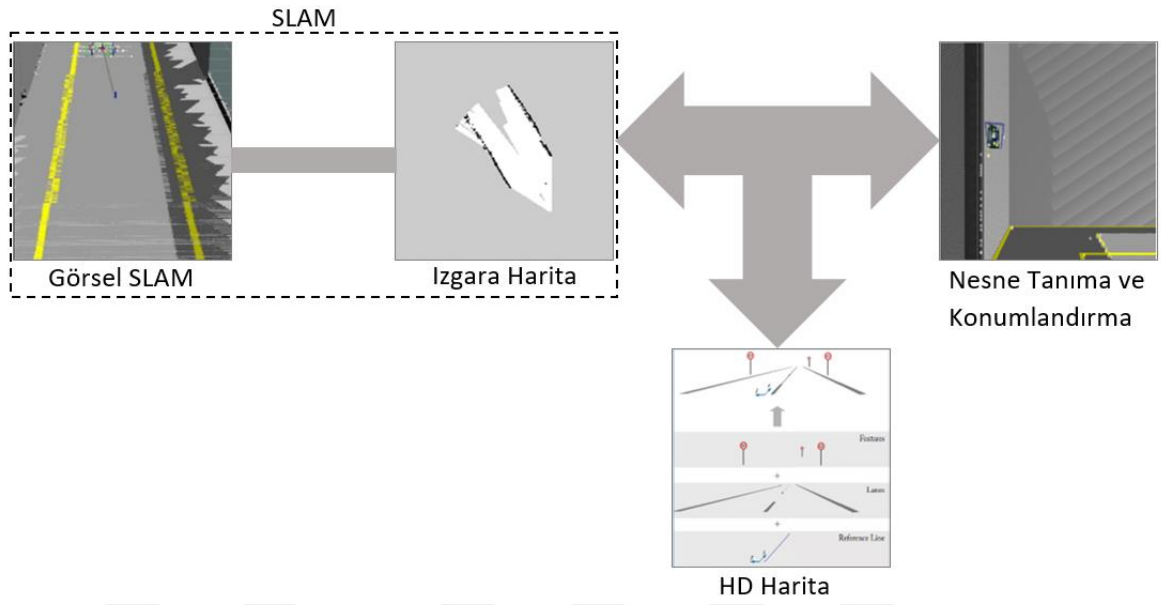
*Cartographer*: Çoklu platformlar ve sensör konfigürasyonları boyunca 2B ve 3B'de gerçek zamanlı SLAM sağlayan bir sistemdir.

*Rtabmap\_ros*: Bu paket, ortamın 3B nokta bulutlarını oluşturmak ve / veya navigasyon için 2B doluluk ızgara haritası oluşturmak için kullanılabilir. Bu paket harita grafiğinin, bir döngü kapanışı algılandığında artımlı olarak oluşturulduğu ve optimize edildiği yerdir.

#### 4. MATERYAL VE YÖNTEM

Tam otonom OTA sistemi oluşturmak için, birçok bileşenin birlikte çalışması ve bütünsel olarak optimize edilmesi gerekmektedir. OTA'lar mesafe ölçmek için sonar ve lazer sensörlerine, anlık görüntü almak için kameralara, fabrikadaki diğer robotlarla ve fabrikayla etkileşimi için iletişim gerektiren birtakım sensörlere ihtiyaç duyarlar. OTA'nın güvenli navigasyon gerçekleştirmesi için, ortamdan gerekli bilgileri sağlaması, yani çevreyi bağımsız olarak algılaması gerekmektedir. Ortamı algılayabilmek için, temel duyu olarak lazer tarayıcılar veya kamera kullanılırken, ek duyular olarak atalet ölçü birimi (inertial measurement unit – imu) ve odometri bilgileri kullanılmaktadır. Tüm bu sensör bilgilerinin birleştirilmesi robotun konumunun daha hassas bir şekilde elde edilmesini sağlamaktadır. Klasik robotik haritalarının çeşitli nesne etiketleri, nesne konumları gibi ortama ait özellik bilgilerini içermemesi, ortama ait detaylı özelliklerin içerildiği HD harita kullanımını zorunlu hale getirmiştir. HD harita, OTA'ların görevlerini başarılı bir şekilde tamamlaması için yeterli detaylarda bilgi sağlamaktadır.

Bu çalışmada, otonom taşıyıcı araçların fabrika ortamında güvenli bir şekilde hareket etmesi için gerekli olan ortam haritasının görsel olarak çıkarılması, çıkarılan haritadaki işaret levhaları, yol çizgileri, yönlendirme işaretleri gibi önemli nesnelere tanımlanması ve bu nesnelere mevcut haritadaki konumlarının doğrulanması üzerine bir yöntem geliştirilmiştir. Ortamdaki nesne konumlarında değişiklikler olduğunda, bu değişiklikler tespit edilerek mevcut HD haritanın güncellenmesi sağlanmaktadır. Önerilen yöntem Şekil 4.1'de özetlenmiştir. Önerilen yöntemde HD harita, temelde ızgara haritanın bulunduğu ve bunun 3B görsel harita ile güçlendirildiği SLAM kısmından oluşmaktadır. Bu SLAM bilgisine, ortama ait tüm geometrik bilgilerin eklendiği ve ortamda bulunan nesnelere algılanarak konumlarının bulunduğu harita bilgisi eklenir. HD harita güncellenirken, öncelikle robotun gerçek zamanlı olarak tanımladığı nesnelere, veritabanında bulunan nesnelere karşılaştırılır ve nesnenin ortamda bulunup bulunmadığı kontrol edilir. Daha sonra, nesnelere ait konum bilgisindeki değişimler öncül HD harita karşılaştırılmaktadır. Eğer ilgili nesnenin konum bilgisinde değişim varsa veya ortama yeni bir nesne eklenmiş / çıkartılmışsa, yeni bilgilerle harita güncellenir. Yönteme ait detaylar, aşağıda verilmektedir.



Şekil 4.1. Önerilen HD Harita Güncelleme Yöntemi

#### 4.1. SLAM

Çalışmada 2B ızgara harita, lazer sensör yardımıyla elde edilmiştir. 2B ızgara harita, HD haritanın temelini oluşturmaktadır ve robotun temel konumlandırma işlemleri için yeterlidir. Görsel SLAM uygulamasının çözülmesi için RTAB-Map ve RGB-D kamera kullanılmıştır. RTAB-Map algoritmasında, ortamın 3B nokta ızgarası haritası, ortamın 3B nokta bulutları kullanılarak oluşturulabilir. Algoritma, bir döngü kapanması algılandığında, adım adım oluşturulmuş ve optimize edilmiş bir harita grafiği üretmektedir. Algoritmanın çıktısı, haritaya eklenen en yeni lokal grafiğdir. Harita elde etmek için algoritma, RGB görüntüsünü, derinlik bilgisini ve lazer tarama verisini kullanır. Algoritma, son konumuna ulaştığında, grafiğin optimize edilmiş durumunu tanımlar. Tüm pozisyonlar konum dizisinde tutulur ve sıralı konumlar birbirine bağlanır.

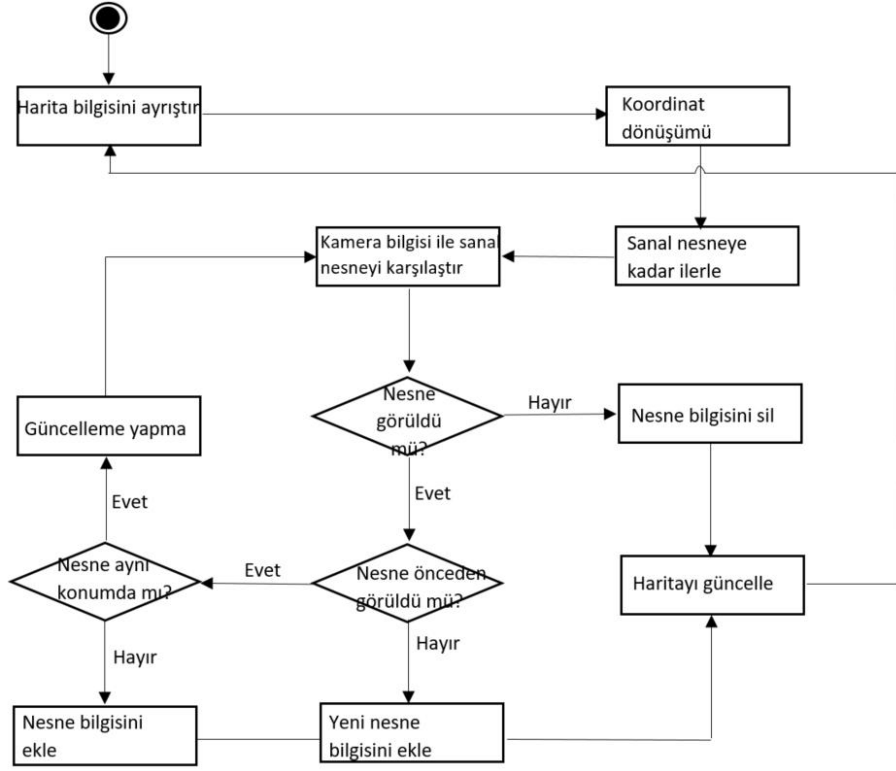
#### 4.2. Nesne Tanıma ve Konumlandırma

Otonom araçlarda kullanılan HD harita çalışmalarında, yolda bulunan her özelliğin haritaya işlenmesi gerekmektedir. Yolun genişliği, uzunluğu, eğriliği gibi yol bilgileri; park alanı, yaya alanı gibi alan bilgileri; dur işareti, park işareti gibi yol işaret ve ortamda bulunan diğer nesnelere ait geometrik bilgiler, harita standartlarında kaydedilmiştir. Ortama ait tüm

geometrik bilgileri elde etmek için grafiksel tabanlı harita olan OSM (OpenStreetMap)'den (Anonim 2017) faydalanılmış ve ortama ait bilgiler XML (eXtensible Markup Language) olarak kaydedilmiştir. Bu çalışmada, ortamdaki önceden eğitilmiş bazı nesnelere tespit etmek ve bu nesnelere yerlerini belirlemek için RGB-D kameradan çekilen görsel görüntüler kullanılmaktadır. Nesne algılama, OpenCV (Bradski, 2000) kütüphanesi kullanılarak gerçek zamanlı olarak gerçekleştirilmiştir. Nesne algılama eğitim ve test aşaması olmak üzere iki aşamadan oluşur. İlk aşamada, genellikle ikili sınıflandırıcılar, (nesneye ait görsel görüntüler) ve negatif (nesneye ait olmayan görsel görüntüler) kullanılarak sistem çevrimdışı olarak eğitilir. İkinci aşamada, dedektör sahnenin eğitilmiş nesneyi içerip içermediğine hızlı bir cevap verebilmektedir. Bu çalışmada, hem eğitim hem de test aşamalarının kolayca gerçekleştirilebildiği "Find-Object" (Labbe, 2011b) uygulaması kullanılmıştır. Bu uygulama OpenCV kütüphanesinde bulunan birçok özellik çıkarma yönteminden (SIFT, SURF, BRIEF, FAST) istenilen seçilerek uygulanabilir. Tez çalışmasında nesne eğitiminde SIFT (Scale Invariant Feature Transform) tanımlayıcısı kullanılmış ve çevredeki tüm işaret farklı açılardan eğitilmişlerdir. Eğitim aşamasında, her bir nesneye bir kimlik numarası atanarak, veri tabanına kaydedilmiştir. Test aşamasında, kameradan yakalanan görüntüler eğitilmiş nesnelere ile karşılaştırılmıştır. Dedektör sahnedeki herhangi bir nesne bulursa, nesnenin ortamdaki konumu, görüntüdeki nesne konumları ve kameradaki derinlik bilgileri kullanılarak hesaplanmıştır.

### 4.3. HD Harita

HD haritanın güncellenmesinde ortamda bulunan işaretlerin konum ve anlam bilgileri kullanılmaktadır. Bu aşamada, işaret levhalarının konumlarındaki değişimin algılanıp, haritaya eklenmesi, silinmesi veya düzeltilmesi işlemi gerçekleştirilmektedir. HD haritanın güncellenmesine ait akış diyagramı Şekil 4.2'de verilmiştir.



Şekil 4.2. HD Harita Güncelleme Akış Diyagramı

HD harita güncelleme süreci, çevre için mevcut HD haritasının ön işlemeyle başlar. HD haritada nesnelere ilgili konum bilgileri, ortamın referans koordinat sistemine dönüştürülür. Nesnelerin konum bilgisi, HD haritasındaki enlem boylam birimlerinde gösterilir, ancak gerçek dünyadaki nesnelere, navigasyon sırasında Kartezyen koordinat sisteminde temsil edilir. Koordinat dönüşümü için aşağıdaki dönüşüm formülü uygulanır. Denklemden,  $p_0(\varphi)$  ve  $p_0(\lambda)$  sırasıyla HD haritadaki referans noktasının enlem ve boylamını,  $x$  ve  $y$  Kartezyen konumları,  $\varphi$  ve  $\lambda$  HD haritadaki enlem ve boylam konumlarını verir.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \Delta\varphi & \Delta\lambda \\ \Delta x & \Delta y \end{bmatrix}^{-1} \left( \begin{bmatrix} \varphi \\ \lambda \end{bmatrix} - \begin{bmatrix} p_0(\varphi) \\ p_0(\lambda) \end{bmatrix} \right) \quad (4.1)$$

Yukarıda verilen akış diyagramında görüldüğü gibi, sanal işaret levhası ile karşılaşana kadar robot gezinme işlemini yapmaktadır. Sanal işaret levhasına yaklaşıldığında, eğer işaret kamera tarafından görünmüyorsa, harita bilgisi değişmiş demektir. Bu yüzden ilgili yerde olması gereken işaret bilgileri kameradan gelen bilgiler

doğrultusunda silinerek harita güncellenir. Eğer kamera bir trafik işareti algılıyorsa, fakat bu işaret etiketi daha önceden harita verisi içerisinde mevcut değilse, harita verisi güncel değil demektir ve yeni işarete ait tüm bilgiler dosya içerisine kaydedilerek, harita güncellenir. Böylece, SLAM işlemi sırasında, ortamda tespit edilen nesnelere, önceden tanımlanmış nesne bilgileriyle karşılaştırılır. Algılanan nesnelere doğrulandığında, nesne konumları ve bunların etiketleri haritada işaretlenir. Böylece, OTA'nın güvenli sürüşünü sağlayacak olan harita sürekli olarak güncel veriler barındırır.



## 5. BULGULAR VE TARTIŞMA

Önerilen yöntem, GAZEBO benzetim ortamında tasarlanan iki farklı ortamda test edilmiştir. GAZEBO, otonom robotlar için yazılan algoritmaları hızlıca test etme imkanı sağlayan, gürbüz fizik motoruna, yüksek kalite grafiklere ve kullanışlı grafik arayüzüne sahip bir simülasyon platformudur (Anonim, 2013). GAZEBO simülasyon ortamında yapılan testler, Ubuntu 16.04 ve ROS Kinetic kurulu, içerisinde Intel Core i7 7700 HQ 2.8GHz işlemci ve 16 GB RAM'e sahip bir bilgisayarda gerçekleştirilmiştir. Testlerde, OTA olarak, İnovasyon Mühendislik (2017) tarafından geliştirilen gezgin robot kullanılmıştır. Testlerde kullanılan araç Bölüm 5.1'de tanıtılmıştır. Bölüm 5.2'de uygulamada kullanılan ROS paketleri verilmiştir. Deneysel çalışmalar ise detaylarıyla birlikte Bölüm 5.3'te anlatılmıştır.

### 5.1. Otonom Taşıyıcı Araç

Testlerde kullanılan gezgin robot, RGB-D kamera (Kinect), sonar, laser, bumper, infrared, IMU ve enkoder sensörlerine sahip olup, GAZEBO modeli Şekil 5.1'de gösterilmiştir.



Şekil 5.1. Otonom Taşıyıcı Araç (OTA)

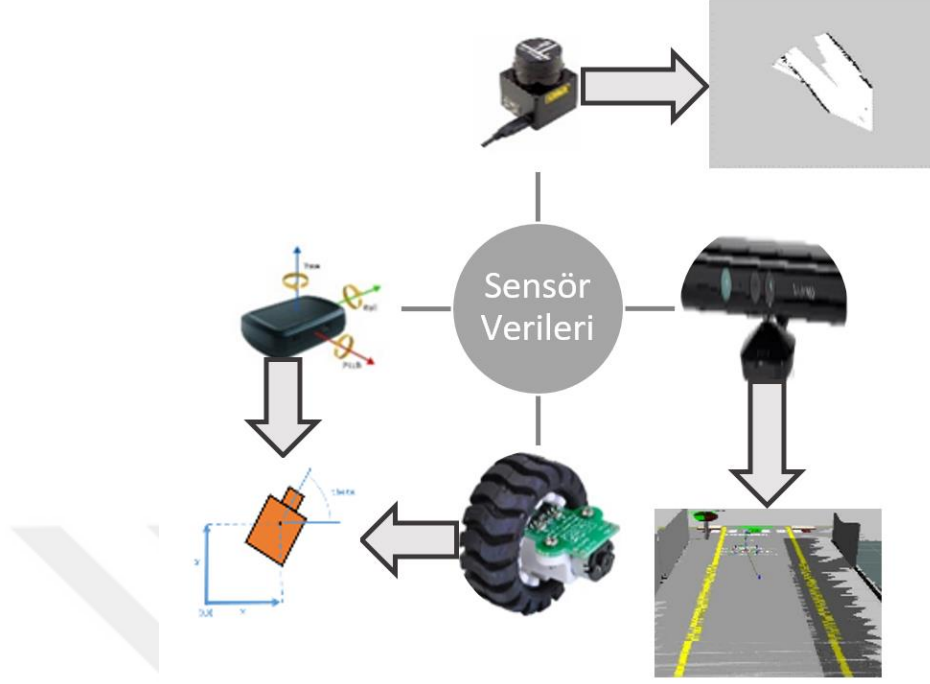


Otonom taşıyıcı araç üzerinde RGB-D sensör olarak, Microsoft (2010) tarafından geliştirilen Kinect sensörü kullanılmaktadır. Kamera, video hızındaki özellikleri yakalayabildiği için 3B haritalara olanak tanır. Kinect sensörü, ortam haritasını çıkarmanın yanı sıra, ortamdaki görüntüleri yakalamak için de kullanılmaktadır. Kinect sensörünün görseli Şekil 5.3'te verilmiştir.



Şekil 5.3. Microsoft Kinect Sensörü

OTA'nın çevresiyle etkileşimi üstünde bulunan sensörler ile sağlanmakta ve sensör sayısı arttıkça OTA'nın çevreden elde ettiği bilgiler artmaktadır. HD harita üretimi için toplanan veriler HD harita verileri ve yardımcı veriler olmak üzere iki kategoriye ayrılabilir (Jiao, 2018); ortama ait geometrik ve semantik bilgilerin içerildiği kamera ve lazer verileri HD harita verileri iken, IMU, odometri gibi aracın pozisyonunun kesin olarak tahmin edildiği ilave sensör bilgileri yardımcı verilerdir. Testlerde, lazer tarayıcı ile 2B harita, kamera ile 3B harita oluşturulmakta, IMU ve odometri gibi yardımcı veriler ile aracın pozisyonu elde edilmektedir. HD harita üretiminde kullanılan sensörler ve sensör verilerinden işlenen kavramlar Şekil 5.4'te verilen görsel üzerinde özetlenmiştir.



Şekil 5.4. Robot Üzerindeki Sensörler ve İlgili Süreçler

## 5.2. Kullanılan ROS Paketleri

Uygulamada, harita çıkarılırken `rtabmap_ros` paketi, işaret levhaları tanımlanırken, `find_object_2d` paketi kullanılmıştır. Bununla birlikte harita çıkarım işlemi `teleop_twist_keyboard` paketi ile klavye kontrolü yardımıyla gerçekleştirilmiştir. Çıkarılan harita `map_server` paketi ile kaydedilmiş ve gezinme işlemleri için hazır hale getirilmiştir. Kullanılan ROS paketleri ile ilgili detaylar aşağıdaki alt bölümlerde detaylı olarak verilmektedir.

### 5.2.1. `rtabmap_ros` paketi

Bu pakette bulunan `rtabmap` ana düğümü, harita grafiğinin bir döngü kapanışı algılandığında artırılmış olarak oluşturulduğu ve optimize edildiği yerdir. Düğümün çevrimiçi çıktısı, haritaya en son eklenen yerel grafiğdir. Abone olunan konular odometri akışı, RGB görüntüsü, derinlik bilgisi, lazer tarama akışı ve stereo kamera için gerekli konulardır. Yayınlanan konular ise, RTAB-Map'in grafiği ve son düğüm verileri, lazer taramalarıyla üretilen doluluk ızgarası, yerdeki 3B nokta bulutlarının projeksiyonundan

üretile doluluk ızgarası, 3B nokta bulutlarıyla üretilen 3B nokta haritalanması, 2B taramalarla veya 3B taramalarla oluşturulan 3B nokta bulutu haritası, global yol planlayıcısı, lokal yol planlayıcısı, rtabmap'in topolojik planlayıcısından gönderilen geçerli metrik hedefdir. Bunların dışında harita verilerini almak için, lazer taramalarından veya 3B nokta bulutlarının izdüşümünden elde edilen doluluk ızgara haritası çağırmak için ve harita verilerini yayınlamak için gerekli servisler vardır. Bu servislerle beraber, lokalizasyon veya navigasyon modunun seçileceği servislerle, bir noktaya etiket atama veya bir topolojik hedef belirleyici servisler de vardır.

Bu düğüm dışında, rtabmap'in görselleştirme arabirimini başlatan rtabmapviz düğümü, rtabmap'in rgb odometri metodunu içeren rgb odometry düğümü, rtabmap'in stereo odometri metodunu içeren stereo odometry düğümü, usb kameradan görüntü alımı için gerekli olan kamera düğümü ile map\_assembler ve map\_optimizer düğümleri gibi düğümler de mevcuttur.

### 5.2.1.1. Pakette bulunan önemli mesaj tipleri

rtabmap\_ros paketinde bulunan ve çalışmada sıklıkla kullanılan mesaj tipleri ve açıklamaları Çizelge 5.1'de verilmiştir.

Çizelge 5.1. RTAB-Map Önemli Mesaj Tipleri

Mesaj Tipi	Açıklama
nav_msgs/Odometry	Serbest uzaydaki konum, yön ve hız tahminini temsil eder.
sensor_msgs/Image	Sıkıştırılmamış imge bilgisini içerir.
sensor_msgs/LaserScan	Düzlemsel bir lazer mesafe bulucudan gelen tekil tarama. Mesafe verilerini, minimum ve maksimum mesafe değerlerini, taramanın başlangıç ve bitiş açılarını içerir.
sensor_msgs/PointCloud2	Normaller, yoğunluk vb. ek bilgiler içerebilen N-boyutlu noktalardan oluşan nokta bulutunu temsil eder.
geometry_msgs/Transform	İki koordinat çerçevesi arasında dönüşümü temsil eder.
rtabmap_ros/MapData	Optimize edilmiş grafiği ve grafik verilerini içerir.

rtabmap_ros/MapGraph	Harita odom dönüşümünü, konumları ve konumlar arasındaki bağlantıları içerir
rtabmap_ros/NodeData	Düzeltilmemiş pozlardan gelen odometri verilerini, görüntü bilgisini, lazer taramasını, base_link-camera_link arasındaki dönüşümü, sıkıştırılmış doluluk ızgarası verilerini, anahtar noktaları içerir.
nav_msgs/OccupancyGrid	Her bir hücrenin doluluk olasılığını temsil ettiği bir 2B ızgara haritasını temsil eder.

rtabmap\_ros/MapData mesajı rtabmap\_ros/MapGraph mesajı tarafından sağlanan optimize edilmiş grafiği ve rtabmap\_ros/NodeData mesajı tarafından sağlanan grafik verilerini içerir. rtabmap\_ros/MapGraph Message son konuma gelindiğinde, grafiğin optimize edilmiş halini tanımlar. Öncelikle, haritadan odom'a dönüşüm geometry\_msgs/Transform yardımıyla yapılır. Bu dönüşüm, serbest uzaydaki iki koordinat çevresinin vector3 tipinde  $x,y,z$  yönleri ile dönüşümünü ve quaternion tipinde  $x,y,z,w$  vektörleri ile oryantasyonunu sağlar. Tüm konumlar posesId dizisinde tutulur. Daha sonra, çalışma belleğindeki bütün konumlar geometry\_msgs/Pose mesaj tipi ( $x,y,z$  noktalarını konum için and  $x,y,z,w$  quaternionlarını oryantasyon için kullanır) tarafından kaydedilir. Ardışık konumlar birbirine geometry\_msgs/Transform mesaj tipi tarafından bağlanır. rtabmap\_ros/NodeData mesaj tipi ise, rtabmap\_ros/MapGraph mesaj tipinin detaylı versiyonudur. Yani bütün konumları detaylıca inceler.

### 5.2.1.2. Parametreler

Bu pakette bulunan ve çalışmada kullanılan bazı parametreler Çizelge 5.2'de verildiği gibi ayarlanmıştır.

Çizelge 5.2. Çalışmada Kullanılan RTAB-Map Parametreleri

Parametre Adı	Değer	Açıklama
RGBD/NeighborLinkRefining	True	Ardışık lazer taramalarında odometri düzeltmesi yapma.
RGBD/ProximityBySpace	True	Çalışma belleğindeki konumlarla yaklaşıklık algılama yapma (tahmini konumu kullanarak).
Mem/BadSignaturesIgnored	False	3B düğüm oluşturma için kötü görüntüleri göz ardı etme (Örneğin beyaz duvar)

Reg/Strategy	1	Kayıt stratejisi: 0=görsel, 1=ICP, 2=görsel+ICP
Icp/CorrespondenceRatio	0.2	Dönüşümü kabul etmek için eşleşen benzeşmelerin oranı
Icp/Iterations	30	Maksimum iterasyon
Icp/VoxelSize	0.025	Tek tip örnekleme voksel boyutu (0 = devre dışı)
Vis/MinInliers	10	Döngü kapanmasını kabul etmek için, minimum inlier
Vis/MaxDepth	4.0	3B görsel kelimeler maksimum derinlik (0 = sonsuz)
Vis/InlierDistance	0.1	3B görsel kelimeler karşılık gelen mesafeler
RGBD/AngularUpdate	0.01	Robot hareket ettiğinde harita güncelleme
RGBD/LinearUpdate	0.01	Robot hareket ettiğinde harita güncelleme
Rtabmap/TimeThr	700	Harita güncellemesi için maksimum süre (ms). (0 = sonsuz)
Mem/RehearsalSimilarity	0.45	Tekrar benzerliği
Mem/NotLinkedNodesKept	False	Bağlantı düğümlerini veri tabanı içinde tutma
Reg/Force3DoF	True	3 serbestlik dereceli dönüşüme zorlama (z, roll, pitch parametreleri 0'a ayarlanacaktır.)
Mem/IncrementalMemory	True	SLAM mod, aksi takdirde lokalizasyon modu

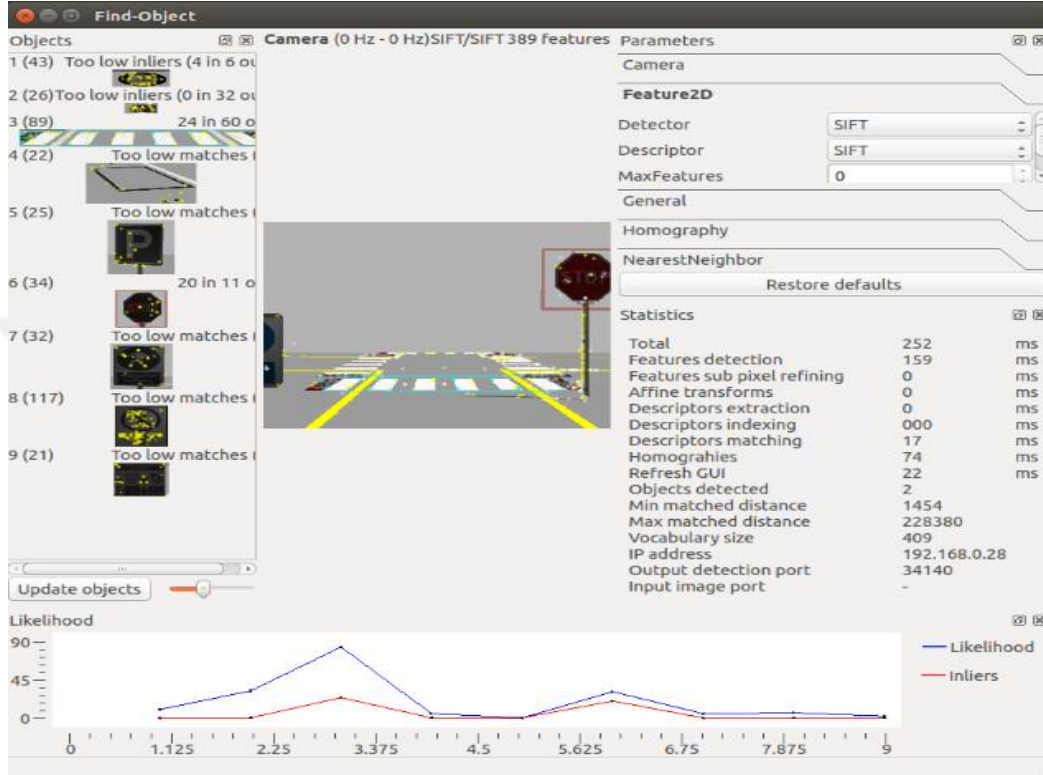
### 5.2.2. find\_object\_2d paketi

find\_object\_2d ros paketi, SIFT, SURF, FAST, BRIEF ve diğer özellik dedektörleri ve tanımlayıcılarının OpenCV uygulamalarını denemek için basit bir Qt arayüzü sunar. Uygulama çalışma zamanında herhangi bir parametreyi değiştirebilme ve her zaman yeniden derlemeye gerek kalmadan özellik dedektörleri ve tanımlayıcılarını test edebilme özelliklerine sahiptir. Herhangi bir kamera yardımıyla nesnelere kimlik ve konum ile ROS'ta algılanabilir ve yayınlanabilir. Bu paket, Find-Object uygulamasının bir ROS entegrasyonudur.

Uygulamanın çalışabilmesi için, yayınlanan imge konusuna (sensor\_msgs/Image) abone olunması gerekmektedir. Tanımlanan nesnelere konumları, yönleri, boyutları *objects* konusu altında yayınlanır. Tanımlanan nesnelere pozisyonları *tf* ros paketi üzerinden yayınlanır. Ayrıca bu paket sayesinde tanımlanan nesnelere harita üzerinde gösterilebilmektedir.

Bu paket içerisinde kullanıcı dostu bir arayüz mevcuttur. Bu arayüzdeki parametreler hem kod üzerinden hem de arayüz üzerinden ayarlanabilmektedir. Find-Object arayüzü Şekil

5.5'te verilmiştir. Arayüzün sol kısmında veri tabanında bulunan nesnelere, sağ kısmında ayarlar, orta bölümde robot kamerasının gördüğü alan ve orta alt bölümde olasılık ekranı görünmektedir.



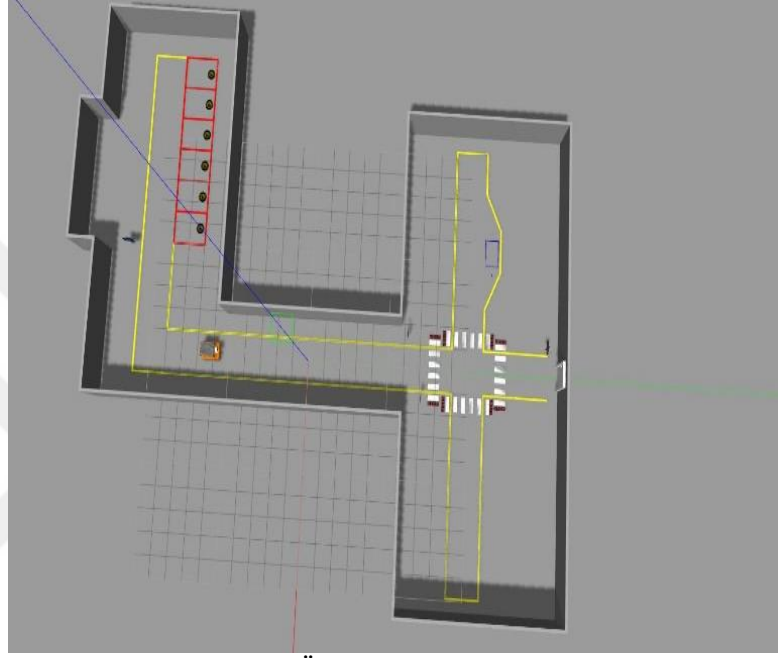
Şekil 5.5. Find-Object Uygulama Arayüz Ekranı

### 5.3. Testler

Testler iki farklı ortamda gerçekleştirilmiştir. İlk ortam, yöntemin test edildiği birinci test ortamı olarak adlandırılmıştır. İlk ortamda yapılan testlerde, bu ortamda bulunan işaret levhalarının konum doğrulukları elde edilmiş ve ortam harita bilgisi çıkarılmıştır. Daha sonra, çıkarılan harita bilgisi ve ortamda bulunan nesne bilgileri kullanılarak, haritadaki değişiklikler algılanmış ve harita güncellenmiştir. İkinci testte ESOGU'de oluşturulan örnek bir fabrika ortamının modeli GAZEBO'ya aktarılmıştır. Ortamda bulunan nesnelerin konum doğrulukları farklı açı / mesafeden test edilmiş ve ortam harita bilgisi çıkarılmıştır. Bu harita bilgisine ortamda bulunan nesnelerin bilgisi eklenmiş ve haritada değişiklik algılandığında harita güncellenmiştir. Testlere ait detaylar aşağıda mevcuttur.

### 5.3.1. Birinci test ortamı

GAZEBO benzetim ortamında mobil robotun ve algoritmanın testlerinin gerçekleştirildiği örnek ortam oluşturulmuştur. Örnek ortam Şekil 5.6’da gösterilmiştir.



Şekil 5.6. Örnek Test Ortamı

Ortamda 4 farklı işaret levhası ve 4 farklı yol çizgisi bulunmaktadır. Ortamda bulunan sarı yol çizgisi robotun hareket alanını, beyaz yol çizgileri yaya geçidini, kırmızı yol çizgileri robot park alanını ve mavi yol çizgileri robotun bekleme alanını göstermektedir. Testlerde üzerinde çalışılan bazı işaretler Şekil 5.7’de verilmiştir.



Şekil 5.7. Test Ortamı İçin Eğitilen Nesneler

Ortamda bulunan işaretlerin GAZEBO ortamındaki gerçek konumları Çizelge 5.3'te verilmiştir.

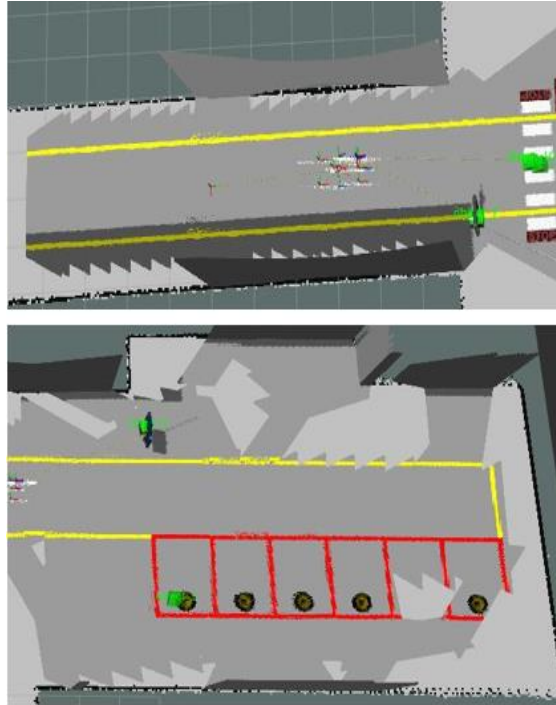
Çizelge 5.3. Örnek Test Ortamında Bulunan İşaretlerin Gerçek Konumları

İşaret ID	İşaret Adı	X Konumu (m)	Y Konumu (m)
1.1	Park Alanı 1	-5,597958	-7,380200
3	Yaya Geçidi	0,000709	7,897850
5	Park Alanı İşareti	-4,800490	-11,642100
7	Yaya Geçidi İşareti	-1,138134	6,344435

Örnek ortam için iki farklı test yapılmıştır. İlk testte, nesnelerin konum doğrulukları kontrol edilmiş ve ortamın harita bilgisi elde edilmiştir. İkinci testte ise, ortamdaki değişiklikler algılanarak harita güncellenmiştir.

#### 5.3.1.1. Test1: nesnelerin konum doğruluğu ve ortam haritasının çıkarılması

Otonom Taşıyıcı Araç bilinmeyen bir ortamda klavye yardımıyla gezdirilerek, ortam haritasını çıkarmıştır. Çıkarılan haritada ilgili işaretlerin tanımlandığı kısımlar Şekil 5.8'de verilmiştir.



Şekil 5.8. Çıkarılan Örnek Harita

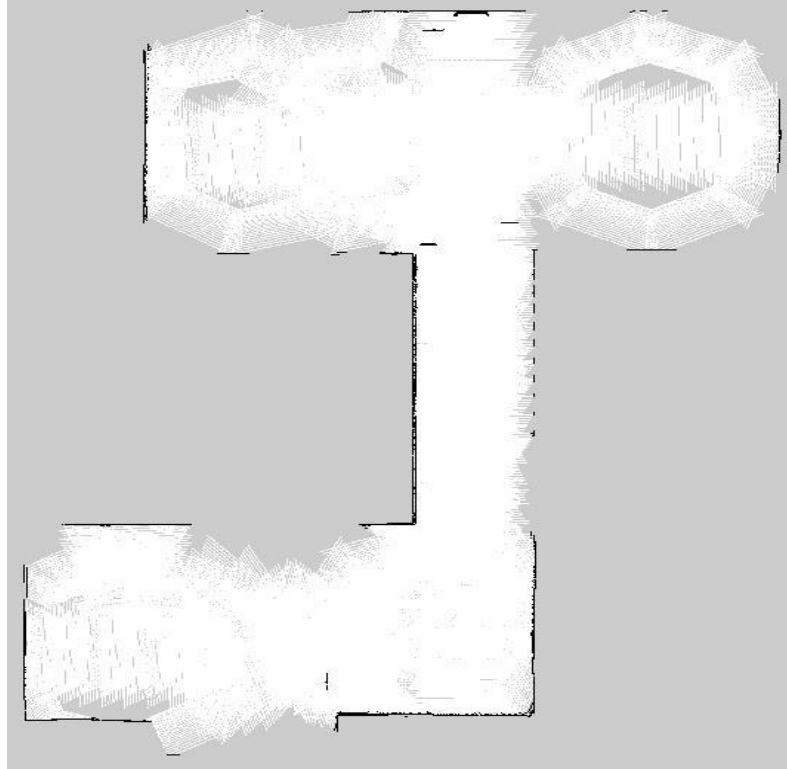
İşaretlerin tanımlanıp, konumlarının öncül HD haritada bulunan konumlarla karşılaştırılması işlemi yapılmaktadır. Çizelge 5.4'te tanımlanan işaretlerin konum bilgilerinin kaydedildiği dosya içeriği gösterilmiştir.

Çizelge 5.4. Tanımlanan İşaretlerin Konumları

İşaret ID	İşaret Adı	X Konumu (m)	Y Konumu (m)
1.1	Park Alanı 1	-5,697688	-7,358555
3	Yaya Geçidi	0,000481	7,878962
5	Park Alanı İşareti	-4,815551	-11,541763
7	Yaya Geçidi İşareti	1,232589	6,405317

Çizelge 5.3 ve Çizelge 5.4 karşılaştırıldığında, GAZEBO'da gerçek konumlarında bulunan nesnelere ile, tanımlanan nesnelere konumlarının birbirlerine oldukça yakın olduğu görülmektedir. Bu sonuç, robotun ortam haritasını çıkarması esnasında ortamda bulunan değişikliklerin kolay algılanmasını ve haritayı güncellemesini sağlar.

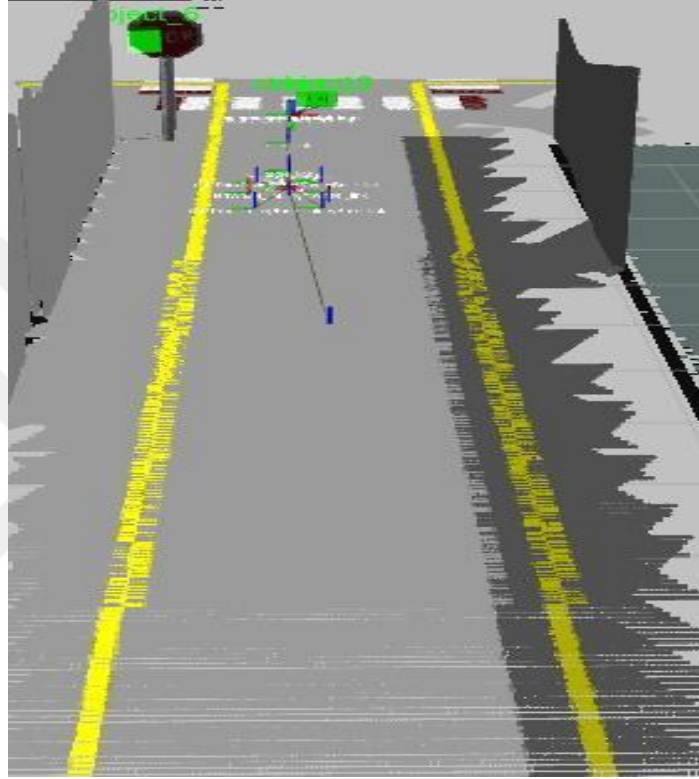
Çıkarılan tüm ortam haritası 68 mb'lık bir alanda depolanmaktadır. Çıkarılan ortam haritasına ait ızgara haritası Şekil 5.9'da verilmiştir.



Şekil 5.9. Çıkarılan Haritanın İki Boyutlu Görüntüsü

### 5.3.1.2. Test2: ortam harita bilgisinin güncellenmesi

İkinci testte, çevreye “Dur” işareti eklenir ve “Yaya Geçit İşareti” (Bkz. Şekil 5.7) ortamdan çıkarılır. Şekil 5.10, haritanın güncellenmiş bölümünü gösterir.



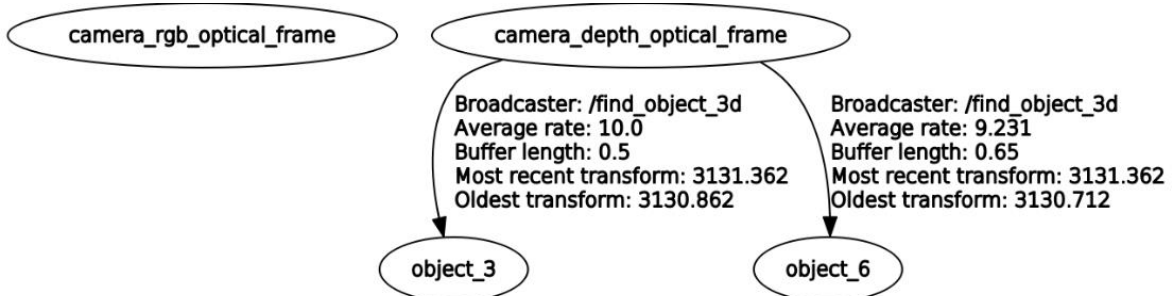
Şekil 5.10. Haritanın Güncellenen Bölümü

Harita güncellendiğinde, ortamdan kaldırılan ve çevreye eklenen işaretler de dahil olmak üzere, son durumdaki nesnelere konumları Çizelge 5.5'te verilmiştir.

Çizelge 5.5. Güncelleme Sonrası İşaretlerin Konumları

İşaret ID	İşaret Adı	X Konumu (m)	Y Konumu (m)
1.1	Park Alanı 1	-5,697688	-7,358555
3	Yaya Geçidi	0,000481	7,878962
5	Park Alanı İşareti	-4,815551	-11,541763
6	Dur İşareti	-1,373628	6,016787
7	Yaya Geçidi İşareti	Yok	Yok

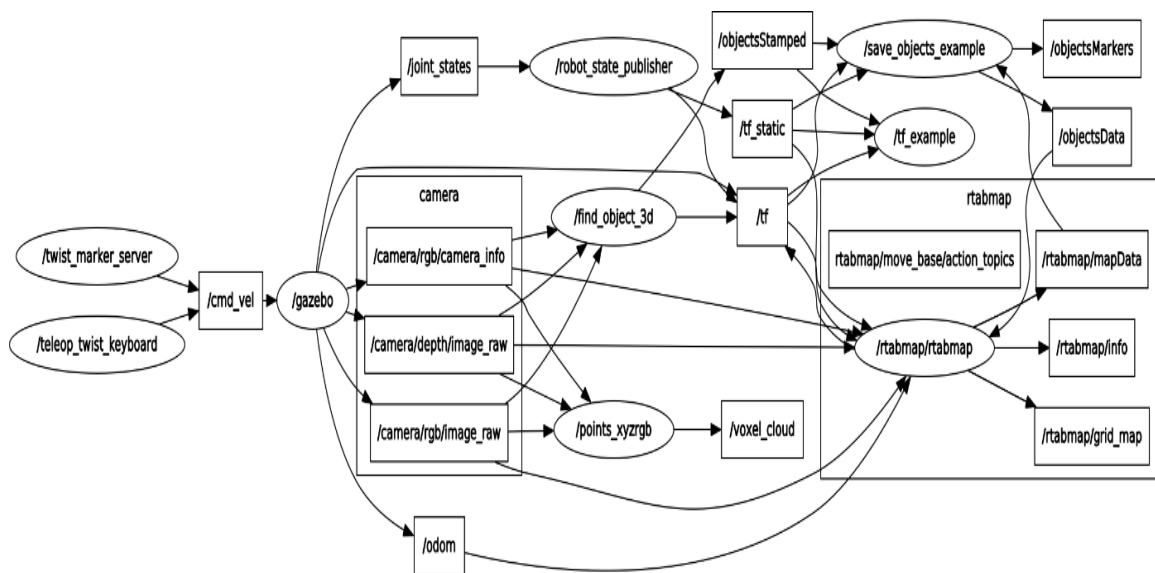
Ortamda bulunan nesnelere tanımlandığında, nesne ile robot kamera çerçevesi arasında bir dönüşüm gerçekleştirilir. Herhangi bir nesne tanınmazsa, kamera çerçevesi herhangi bir nesne ile bağlantı kuramaz. Fakat nesne tanımlandığında, örneğin Şekil 5.11’de tanımlanan “3” ve “6” numaralı nesnelere için, kamera derinlik çerçevesi ile tanımlanan nesnelere arasında `find_object` paketi içerisinde bulunan `tf` düğümü ile dönüşüm gerçekleşir.



Şekil 5.11. Nesne Tanımlandığında Robot İle İşaret Arasındaki Dönüşüm

Bu deneyler diğer nesnelere için ekleme ve çıkarma durumlarında tekrar edilmiş ve benzer sonuçlar elde edilmiştir. Sonuç olarak, HD haritanın OTA tarafından başarıyla güncellendiği görülmektedir.

Şekil 5.12’de bu örnek ortamda yapılan testlerde kullanılan ROS grafiği gösterilmiştir.



Şekil 5.12. Sistemin ROS Grafiği

### 5.3.2. Örnek fabrika ortamı

Önerilen yöntem için, ESOGU’de örnek bir fabrika ortamı kurulmuştur. Fabrika ortamına ait simülasyon modeli GAZEBO’da oluşturulmuş ve tasarlanan OTA ile testler gerçekleştirilmiştir. Ortama ait detaylar ve yapılan testler aşağıda verilmiştir.

Oluşturulan örnek ortam 11,76 m × 4,62 m boyutlarında olup, içerisinde işaret levhaları, şeritler, bölgeler, veri toplama cihazları ve diğer makineler bulunmaktadır. ESOGU’de oluşturulan örnek fabrika ortamı Şekil 5.13’te verilmiştir.



Şekil 5.13. Örnek Fabrika Ortamı

Gerçekte oluşturulan ortamın GAZEBO simülasyon platformundaki modeli Şekil 5.14’te gösterilmiştir. Ortam park alanı gibi bölgeleri, bazı önemli işaret levhalarını ve OTA’yı içerir. Sarı ile gösterilen yol çizgileri OTA’nın hareketini gerçekleştirdiği alanı, beyaz ile gösterilen çizgiler yaya yolunu, kırmızı çizgi ile gösterilen bölge ise robot park alanını ifade eder. Tanımlanan tüm işaret levhaları ve yol çizgileri fabrika standartlarına göre belirlenmiştir.



Şekil 5.14. Fabrika Ortamının GAZEBO Çizimi

Akıllı fabrikalarda çok fazla dinamik ve statik nesne bulunmaktadır. Ortamdaki bazı nesnelere işaret levhaları, yola ait bilgiler, insan veya diğer robotlar gibi otonomi için yüksek öneme sahip iken, bazı nesnelere ise, robotun yönlendirilmesinde daha az etkili olan duvarlar, raflar, vb. nesnelere yer verilmektedir. HD haritanın varlığı için, ortamda bulunan dönüm noktalarının tanımlanması ve bunların konumlarının doğru bir şekilde bulunarak haritada işaretlenmesi gerekir. Bu testte, sadece ortamda bulunan işaret levhaları eğitilmiştir. Eğitilen işaretler, fabrika ortamlarında sıklıkla kullanılan ve fabrika standartlarına uygun işaret levhaları olup, Şekil 5.15'te gösterilmiştir.



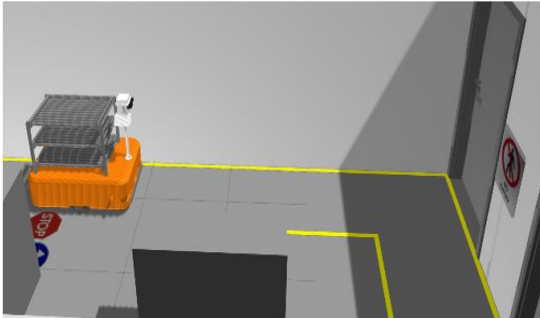
Şekil 5.15. Eğitilen İşaretler

HD haritanın güncellenmesinde, sadece ortamda bulunan işaret levhalarının konum değişimi dikkate alınmaktadır. Öncül HD harita koordinat bilgilerini enlem-boylam cinsinden tutmaktadır. Bu işaretlerin ortam haritasında nerede konumlandığını gösterebilmek için Denklem 4.1’de verilen dönüşüm formülü kullanılarak işaretlerin Kartezyen sisteme çevrilmesi gerekmektedir. Çizelge 5.6’da, ortamda bulunan işaret levhalarının enlem-boylam cinsinden konumları ile, GAZEBO’da işaretlerin Kartezyen konum bilgileri verilmiştir. Her işarete ait bir ID, işaret ismi,  $x$  konumu ve  $y$  konumu XML dosyası içerisine kaydedilmiştir.

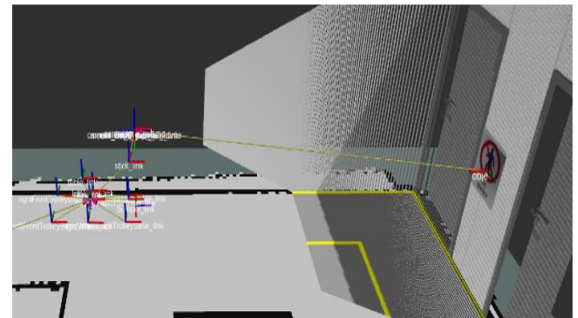
Çizelge 5.6. Fabrika Ortamında Bulunan İşaretlerin HD Harita ve GAZEBO Konumları

ID	İşaretAdı	Pozisyon(Enlem)	Pozisyon(Boylam)	Pozisyon(x)	Pozisyon(y)
1	Park	-0,00520132144	-0,01019512304	4,015525	0,237827
2	Yaya	0,001545193317	-0,00074507644	6,300000	1,800000
2	Yaya	0,001436094079	0,010662636959	9,030000	1,800000
3	Ok	0,007051594389	0,001895234619	6,930000	3,150000
4	Dur	-0,00005417232	0,019435891634	6,930000	3,570000
4	Dur	0,003560223956	0,005400489614	7,770000	2,310000
4	Dur	0,003545543256	0,015930686639	10,290000	2,310000
4	Dur	0,008794832823	0,001897714959	11,130000	1,470000
6	Girilmez	0,006123579642	0,022368995860	11,829900	2,933290

Bu ortam için gerçekleştirilen test senaryolarında, robot ile işaret levhaları arasındaki mesafe hesaplanmıştır. Şekil 5.16a ve Şekil 5.16b’de sırasıyla örnek bir işaret tanımlaması esnasında robotun ortamdaki yeri ve işaretin robot kamerasında algılanmasından sonra ROS tarafından kamera ile işaret arasında oluşturulan dönüşümün gösterimi mevcuttur.



Şekil 5.16a. Robotun Konumu



5.16b. Robot-İşaret Arasındaki Dönüşüm

Dönüşüm işleminde, robot konumu ile tanımlanan işaret arasındaki konum farkları elde edilmektedir. Şekil 5.16b’de verildiği üzere, robotun konumu  $p_r = (x_r, y_r)$  ve işaretin konumu  $p_s = (x_s, y_s)$  olsun. Dönüşüm çerçevesi sonucunda robot ile işaret arasındaki mesafe  $d_{rs} = (d_x, d_y)$  aşağıdaki denklemlerde verildiği gibi elde edilmektedir.

$$d_{rs} = (d_x, d_y) = (x_s - x_r, y_s - y_r) \quad (5.1)$$

Şekil 5.16a’da verilen pozisyonda iken robotun gerçek konumu (7,308609, 4,103419) metre, işaretin gerçek konumu (11,829900, 2,932290) metredir. Bu durumda iken  $d_{rs}(\text{gerçek})$  değeri (4,521291, -1,71129) metre olarak hesaplanmaktadır.

Robot kamerası işareti algıladığında, ROS tarafından oluşan otomatik çerçeve dönüşümü ile hesaplanan değer  $d_{rs}(\text{deneysel})$  (4,523917, -1,187529) metre olarak bulunmuştur.

İki sonuç arasındaki hata oranına aşağıdaki formül yardımıyla bakılabilir.

$$d_{rs}(\text{Hata Oranı } \%) = \left( \left| \frac{d_{rs}(\text{gerçek}) - d_{rs}(\text{deneysel})}{d_{rs}(\text{gerçek})} \right| \right) \times 100 \quad (5.2)$$

Denklem 5.2 kullanılarak elde edilen  $x$  ve  $y$  konumlarındaki hata oranları aşağıda verilmiştir.

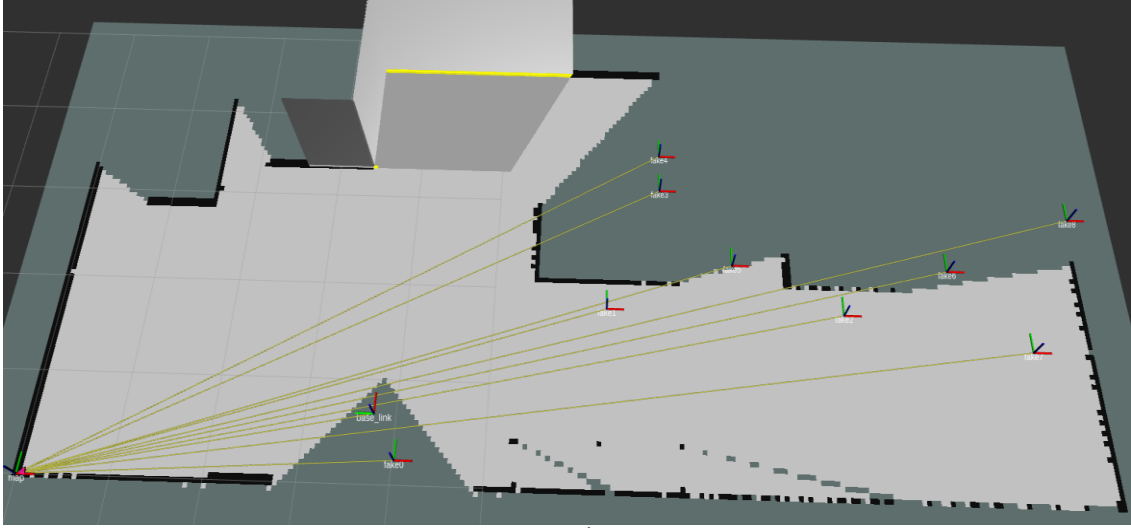
$$d_{rs}(x \text{ Yönünde Hata Oranı } \%) = \left( \left| \frac{4,521291 - 4,523917}{4,521291} \right| \right) \times 100 = \%0,058047$$

$$d_{rs}(y \text{ Yönünde Hata Oranı } \%) = \left( \left| \frac{-1,171129 + 1,187529}{-1,171129} \right| \right) \times 100 = \%1,400358$$

Yukarıdaki hata oranlarından anlaşılacağı üzere, gerçek konumlar arasındaki mesafe ile, ROS tarafından oluşturulan çerçeve üzerinden elde edilen mesafeler oldukça yakın sonuçlar vermektedir.

### 5.3.2.1. Test1: nesnelerin konum doğruluğu ve ortam haritasının çıkarılması

İlk testte, ortamda bulunan nesnelerin konum doğruluğu işlemi farklı mesafe ve açılardan test edilmiş ve doğrulanmıştır. Ayrıca ortama ait harita çıkarılmıştır. Robot hareketine başlamadan önce, HD haritada bulunan değerlere göre, ROS üzerinden sanal çerçeveler oluşturularak, işaretlerin konumlarını belirlemiştir. Şekil 5.17’de HD harita içerisinde konumları bilinen işaret levhalarının, sanal çerçeveler sayesinde konumları harita üzerinde belirlenmiştir.



Şekil 5.17. Ortam Haritasında İşaretlerin Sanal Konumları

Çizelge 5.7’de, harita üzerindeki sanal işaretler ile robot arasındaki mesafeler  $x$  ve  $y$  eksenlerine göre verilmektedir.

Çizelge 5.7. Robot ile Sanal İşaret Çerçeveleri Arasındaki Mesafeler

ID	İşaretAd	$d_{rs}(x)$	$d_{rs}(y)$
1	Park	-0,413767	-0,199489
2	Yaya	1,057684	-2,541020
2	Yaya	0,939274	-5,268450
3	Ok	2,379090	-3,228977
4	Dur	2,798695	-3,247193
4	Dur	1,503449	-4,031756
4	Dur	1,394156	-6,549384
4	Dur	0,518515	-7,352163
6	Girilmez	1,949074	-8,114824

Çizelge 5.7 ve Şekil 5.17 beraber değerlendirildiğinde, işaret ile robot arasındaki  $x$  yönündeki konum farkı pozitif ise işaret robotun önünde, negatif ise arkasında bulunuyor demektir. Eğer, işaret ile robot arasındaki  $y$  yönündeki konum farkı negatif ise, işaret robotun sağ tarafında, pozitifse sol tarafında bulunuyor demektir. Yukarıdaki tabloda bulunan robot-ışaret mesafeleri robotun `base_link`'i dikkate alınarak elde edilmiştir. Sanal işaretler yardımıyla, robot harita üzerinde işaretlerin konumlarını bilmekte ve harita güncellemesi için bu konum bilgilerini ve işaretlere ait etiket bilgilerini kullanmaktadır.

Ortam haritasının çıkarılması işleminden önce, ortamda bulunan işaret levhalarının tanınması için, *Find-Object* uygulaması kullanılmış ve robotun işaret konumlarını belirlemedeki başarısı test edilmiştir. Her bir işaretin konum bilgisinin doğruluğu robot ortamda gezinirken beş farklı açı / mesafeden test edilmiştir. Her testte tanımlanan işaretin konum bilgisi tutulmuştur. Daha sonra testler boyunca elde edilen konum bilgilerinin en küçük, en büyük değerleri ile ortalama değerleri hesaplanmıştır. Her işarete ait hesaplanan değerler Çizelge 5.8'de verilmiştir. Hesaplamalarda kameranın yüksekliği ile işaretin yerden yüksekliği dikkate alınmamıştır.

Çizelge 5.8. İşaretlerin Beş Farklı Test Sonucunda Elde Edilen Minimum, Maksimum ve Ortalama Konum Değerleri

ID	İşaretAd	Min (x)	Max (x)	Ort (x)	Min (y)	Max (y)	Ort (y)
1	Park	4,007336	4,045846	4,025237	0,225950	0,248209	0,235849
2	Yaya	6,286563	6,331794	6,313439	1,714419	1,866889	1,777893
2	Yaya	8,840233	9,055507	8,974152	1,734438	2,029244	1,824017
3	Ok	6,944601	7,004015	6,978841	3,090500	3,21411	3,146749
4	Dur	7,002428	7,022489	7,011744	3,521199	3,633282	3,571976
4	Dur	7,705308	8,051906	7,797774	2,322215	2,413173	2,360931
4	Dur	10,018161	10,35227	10,206698	2,200270	2,456398	2,332782
4	Dur	10,768403	11,081817	10,986966	1,377716	1,589975	1,483439
6	Girilmez	11,752718	11,792106	11,765097	3,013716	3,085530	3,060192

Her işaretin konumuna ait her iki eksendeki hata oranları Denklem 5.3 yardımıyla hesaplanmış ve Çizelge 5.9'da gösterilmiştir. Formüllerde verilen deneysel konum bilgileri için ortalama değerler kullanılmıştır.

$$işaret_{x,y}(Hata Oranı \%) = \left( \left| \frac{işaret_{x,y}(gerçek) - işaret_{x,y}(deneysel)}{işaret_{x,y}(gerçek)} \right| \right) \times 100 \quad (5.3)$$

Çizelge 5.9. Ortalama Değerlere Göre İşaretlerin Konumlarındaki Hata Oranları

ID	İşaretAd	Ortalama(x) Değerine Göre % Hata Oranı	Ortalama(y) Değerine Göre % Hata Oranı
1	Park	0,241856	0,831865
2	Yaya	0,213314	1,228144
2	Yaya	0,618467	1,334289
3	Ok	0,704776	0,103200
4	Dur	1,179567	0,055356
4	Dur	0,357454	2,204814
4	Dur	0,809541	0,986251
4	Dur	1,285123	0,914204
6	Girilmez	0,547793	4,326255

Çizelge 5.9 incelendiğinde, robotun tanımladığı işaretlerin konumlarını hassasiyetle elde ettiği görülmektedir. Bu sayede, olası değişikliklerde işaretlere ait bilgiler kolaylıkla güncellenebilmektedir.

Görsel SLAM kullanılarak elde edilen ortama ait harita Şekil 5.18’de verilmiştir. Çıkarılan ortam haritası kaydedilerek, navigasyon işlemleri için hazır hale getirilmiştir. Çıkarılan ortam haritası 175 mb yer kaplamaktadır. Bu test için, robotun hızı  $v = 0,239149$  m/s ve  $\omega = 0,678297$  rad/s olarak ayarlanmıştır.

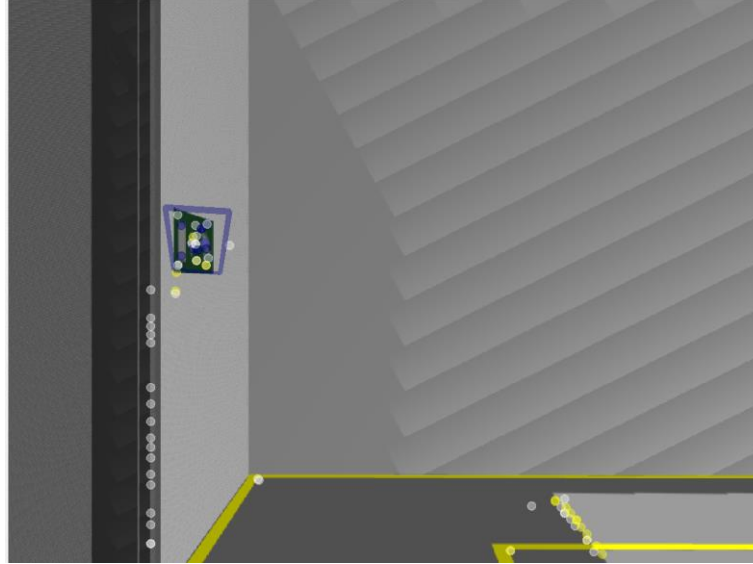


Şekil 5.18. Çıkarılan Ortam Haritası

### 5.3.2.2. Test2: ortam harita bilgisinin güncellenmesi

İkinci testte ise, önerilen HD harita güncelleme algoritması test edilmiştir. Bunun için ortama işaret eklenip, ortamdaki işaret çıkarılmış ve doğru bir şekilde algılamaya yapılarak, haritaya ait bilgiler güncellenmiştir. Ayrıca, güncellenen kısma ait harita elde edilmiştir. Bu testte, ortama işaret ID'si 5 olan "Çıkış" işareti eklenmiş ve ortamdaki işaret ID'si 6 olan "Girilmez" işareti çıkarılmıştır. Ortamda değişiklik meydana geldiğinde robot bu değişiklikleri algılayarak, ortamda bulunan işaretlere ait bilgileri kolayca güncelleyip, kaydetmiştir.

Ortama yeni bir işaret eklendiğinde, robotun algılaması işlemi oldukça kolay olmaktadır. Robot gezinirken, ortamda yeni bir işaret bilgisini Şekil 4.2'de verilen algoritma yardımıyla elde etmektedir. Ortamda bulunan nesnelerin farklı açılardan eğitilmesi işaret tanımlamanın başarısını artırmaktadır. Şekil 5.19'da robotun belirtilen konumda iken, algıladığı işaret görünmektedir.

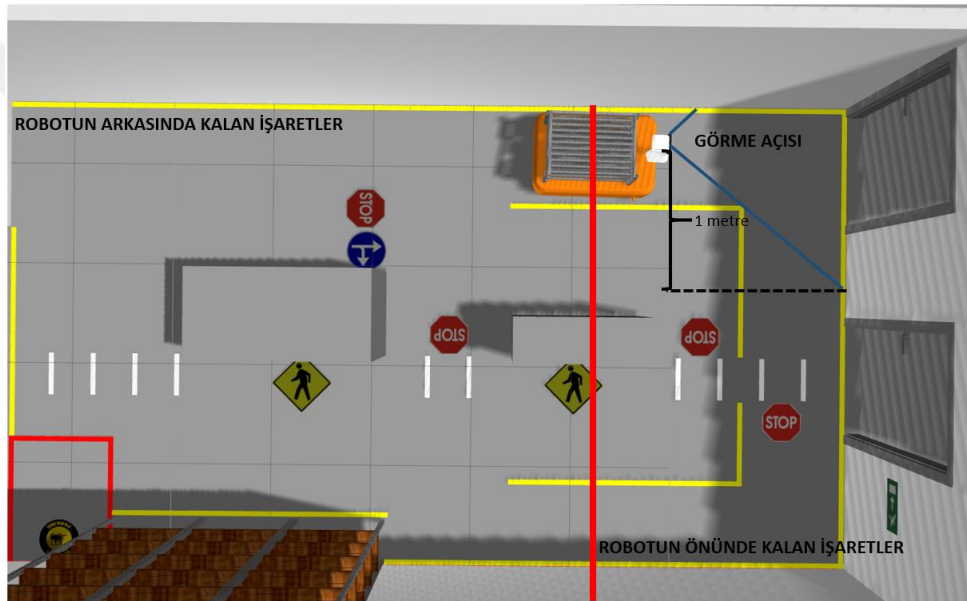


Şekil 5.19. İşaret Tanımlandığında Robot Kamera Görünütüsü

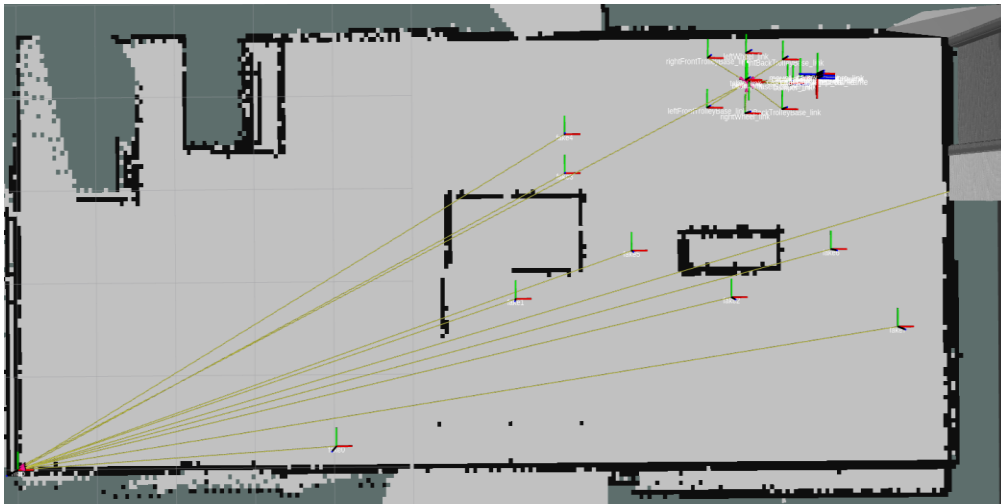
Ortamda bulunan bir işaret kaldırıldığında, robotun bunu algılaması işlemi, işaret ekleme işleminden biraz daha karmaşık bir şekilde gerçekleşmektedir. Öncelikle, sanal çerçeveden alınan bilgiler kullanılarak, işaretlerin haritadaki konumları belirlenir. İşaretin silinmesi için, robot kamerasının görüş alanındaki belirli bir bölgeye bakılır. Daha sonra,

robotun önündeki sanal çerçeveler dikkate alınarak, sadece önünde bulunan işaretler ile ilgilenilir. Robotun mevcut konumunun arkasında bulunan işaretler değerlendirmeye alınmaz. Son olarak, robotun ilgili görüş alanında, sadece işarete ait sanal çerçeve bilgisi görmesi (kamerada işaret bilgisi gelmemesi) durumunda işaret silinir.

Robot konumu (9,182399, 4,089137) iken, işaret levhasını silme algoritmasının işleyişine ait görsel Şekil 5.20’de, işaretlerin sanal çerçeve bilgilerinin haritadaki gösterimi Şekil 5.21’de verilmiştir.



Şekil 5.20. Silinecek İşaret İçin Robotun Konumu



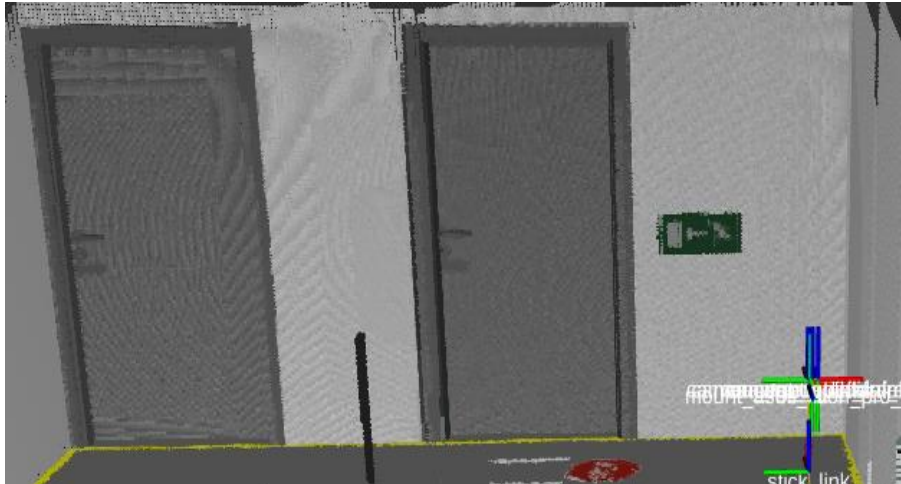
Şekil 5.21. Robot ve Sanal İşaret Levhalarının Konumları

Robotun mevcut konumunda iken işaretler ile arasındaki mesafeler Çizelge 5.10’da verilmiştir.

Çizelge 5.10. Mevcut Konumda Robot-Sanal İşaretler Arasındaki Mesafeler

ID	İşaretAd	$d_{rs}(x)$	$d_{rs}(y)$
1	Park	-5,073995	-3,974127
2	Yaya	-2,286740	-2,348512
2	Yaya	-0,107522	-2,283193
3	Ok	-2,239221	-0,983825
4	Dur	-2,249270	-0,563945
4	Dur	-1,379364	-1,803486
4	Dur	1,138326	-1,744190
4	Dur	1,999772	-2,562853
6	Girilmez	2,584090	-0,962112

Çizelge 5.10’da kırmızı ile gösterilen hücreler, mevcut konumda iken robotun önünde bulunan işaretleri, mavi ile gösterilen hücreler, robotun görme açısında bulunan işaretleri göstermektedir. Her iki şartı da sağlayan işaret ID’si 6 olan “Girilmez” işaretidir. Ortamda kamera tarafından işarete ait bir algılama yapılamadığı ve sadece sanal çerçeve bilgisi mevcut olduğu için, bu işaretin ortamdaki çıkarıldığı anlaşılmaktadır. Çizelge 5.10 ve Şekil 5.20’den anlaşılacağı üzere, sadece ID’si 6 olan işaret silinecektir. Silme işlemi bittikten sonra, ortam haritası güncellenir ve HD harita içeriği son hali ile değiştirilerek kaydedilir. Ayrıca, silinen işarete ait sanal çerçeve bilgisi de haritadan çıkarılır. Ortamın güncellenmiş kısmına ait harita bilgisi Şekil 5.22’de, işaret kısımlarına ait güncellenmiş HD haritaya ait bilgi Çizelge 5.11’de verilmiştir.



Şekil 5.22. Haritanın Güncellenen Bölümü

Çizelge 5.11. İşaretlerin Güncellenmiş Konumları

<b>ID</b>	<b>Ad</b>	<b>Pozisyon (x)</b>	<b>Pozisyon (y)</b>
1	Park	4,015525	0,237827
2	Yaya	6,300000	1,800000
2	Yaya	9,030000	1,800000
3	Ok	6,930000	3,150000
4	Dur	6,930000	3,570000
4	Dur	7,770000	2,310000
4	Dur	10,290000	2,310000
4	Dur	11,130000	1,470000
5	Çıkış	11,722228	0,859305
6	Girilmez	Yok	Yok

İşaret ekleme ve çıkarma testleri diğer nesnelere üzerinde de gerçekleştirilerek, benzer sonuçlar elde edilmiştir. Bu durum HD haritaya ait işaret konum bilgilerinin OTA tarafından kolaylıkla güncellenebildiğini göstermektedir.

## 6. SONUÇ VE ÖNERİLER

Endüstri 4.0 döneminde, akıllı fabrikalar önemli bir rol oynamaktadır ve OTA'lar ürünleri transfer etmek için önemli araçlar olarak düşünülmektedir. Haritalama, bu tür araçların uzun vadeli başarılı operasyonları için kilit süreçtir. Geleneksel haritalama yöntemleri, navigasyon için yeterince iyi çözümler sağlayabilmesine rağmen, HD harita, OTA'nın daha iyi davranışlar sergilemesi için çevre hakkında daha fazla bilgi sağlar.

Bu çalışmada, nesnelere otomatik olarak algılayan, konumlarını belirleyen ve mevcut gözlemi HD harita ile karşılaştırarak, ortamda herhangi bir değişiklik olduğunda haritayı güncelleyen bir yöntem sunulmuştur. GAZEBO simülasyon ortamında gerçekleştirilen farklı ortam ve farklı senaryolardaki testlerde başarılı sonuçlar elde edilmiştir. Çalışmalarla, ortamda bulunan nesnelere konumlarının yüksek hassasiyette bulunduğu gösterilmiş ve HD harita bilgisinin, ortamda değişiklik meydana geldiğinde kolayca güncellenmesi işlemi gerçekleştirilmiştir. Bu çalışmada önerilen HD harita güncelleme stratejisi, akıllı fabrikalarda esnek üretim gereksinimlerini karşılamak için oldukça yararlı olabilir.

Gelecek çalışmalarda, çıkarılan ortam harita bilgisi ve ortama ait tüm geometrik ve semantik bilgiler kullanılarak, robotların fabrika içerisinde güvenli sürüş sağlanması gerçekleştirilecektir. Geleceğin fabrikalarında dinamik ve çok sayıda nesnenin olduğu düşünülerek, ortamda bulunan nesnelere sayısı artırılacak ve simülasyon ortamı biraz daha büyütülecektir. Ayrıca, geleceğin fabrikalarında uyum içerisinde çalışacak çoklu robot sistemlerinden dolayı, testler birden çok robotun bulunduğu ortamlarda gerçekleştirilecektir.

## KAYNAKLAR DİZİNİ

- Anonim, 2009, Robot Operating System (ROS), <http://www.ros.org/>, erişim tarihi: 03.10.2018
- Anonim, 2017, Open Street Map (OSM), <https://www.openstreetmap.org>, erişim tarihi: 03.10.2018
- Anonim, 2018, Gazebo, <http://gazebo.org/>, erişim tarihi: 03.10.2018
- Audras, C., Comport, A., Meilland, M., Rives, P., 2011, Real-time dense appearance-based SLAM for RGB-D sensors, Australasian Conf. on Robotics and Automation, Vol. 2
- Bauer, S., Alkhorshid, Y., Wanielik, G., 2016, Using High-Definition maps for precise urban vehicle localization, Intelligent Transportation Systems (ITSC), pp. 492-497
- Bittel, S., Rehfeld, T., Weber, M., Zöllner, J. M., 2017, Estimating high definition map parameters with convolutional neural networks, Systems, Man, and Cybernetics (SMC), pp. 52-56
- Bradski, G., 2000, The OpenCV Library, Dr.Dobb's Journal of Software Tools
- Busch, S., Quehl, J., Brenner, C., 2018, High Definition Mapping Using LiDAR Traced Trajectories, 38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung
- Da Silva, B. M., Xavier, R. S., do Nascimento, T. P., Gonsalves, L. M., 2017, Experimental evaluation of ROS compatible SLAM algorithms for RGB-D sensors, Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 2017 Latin American pp. 1-6
- Davison, A. J., 2003, Real-time simultaneous localisation and mapping with a single camera. International Conference on Computer Vision, Vol. 2, pp. 1403-1410
- Elfes, A., 1989, Using occupancy grids for mobile robot perception and navigation. Computer, (6), 46-57.
- Fernandes, M., Alexandre, L. A., 2016, SLAMfusion: Fusing SLAM methods for improved robustness, Autonomous Robot Systems and Competitions (ICARSC), pp. 229-234
- Frese, U., 2006, A discussion of simultaneous localization and mapping, Autonomous Robots, 20(1), 25-42
- Grisetti, G., Stachniss, C., Burgard, W., 2007, Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics, 23(1), 34-46.

### KAYNAKLAR DİZİNİ (devam)

- He X., Zhao, J., Sun, L., Huang, Y., Zhang, vd., 2018, Automatic Vector-based Road Structure Mapping Using Multi-beam LiDAR, arXiv preprint arXiv:1804.07028.
- İnovasyon Mühendislik, 2017, OYA-1003, <https://github.com/inomuh/OYA-1003>, erişim tarihi: 03.10.2018
- Jang, C., Cho, S., Jeong, S., Suhr, J. K., Jung, vd., 2017, Traffic light recognition exploiting map and localization at every stage, *Expert Systems With Applications*, 88, 290-304.
- Jiao, J., 2018, Machine Learning Assisted High-Definition Map Creation, *Annual Computer Software and Applications Conference (COMPSAC)*, pp. 367-373
- Jung, M., Song, J. B., 2016, Graph SLAM for AGV using Geometrical Arrangement based on Lamp and SURF Features in a Factory Environment, *Control, Automation and Systems (ICCAS)*, pp. 844-848
- Labbé M., 2011a, RTAB-MAP, <http://introlab.github.io/rtabmap/>, erişim tarihi: 03.10.2018
- Labbé M., 2011b, Find-Object, <https://introlab.github.io/find-object/>, erişim tarihi: 03.10.2018
- Labbé, M., Michaud, F., 2011, Memory management for real-time appearance-based loop closure detection, *Intelligent Robots and Systems (IROS)*, pp. 1271-1276
- Labbé, M., Michaud, F., 2014, Online global loop closure detection for large-scale multi-session graph-based SLAM, *Intelligent Robots and Systems (IROS)*, pp. 2661-2666
- Leonard, J. J., Durrant-Whyte, H. F., 1991, Simultaneous map building and localization for an autonomous mobile robot, *Intelligent Robots and Systems' IROS'91. IEEE/RSJ International Workshop*, pp. 1442-1447
- Li, F., Bonnifait, P., Ibanez-Guzman, J., 2017, Estimating localization uncertainty using multi-hypothesis map-matching on high-definition road maps, *International Conference on Intelligent Transportation Systems (ITSC)*
- Liu S., Li L., Tang J., Wu S., Gaudiot J.L., 2017, *Creating Autonomous Vehicle Systems, Synthesis Lectures on Computer Science* 6 (1), i-186
- Lucke, D., Constantinescu, C., Westkämper, E., 2008, Smart factory-a step towards the next generation of manufacturing, *Manufacturing Systems and Technologies For the New Frontier*, pp. 115-118

**KAYNAKLAR DİZİNİ (devam)**

- Ma, H., Pei, Z., Wei, Z., Zhong, R., 2017, Automatic Extraction of Road Markings From Mobile Laser Scanning Data, *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42.
- Martínez-Barberá, H., Herrero-Pérez, D., 2010, Autonomous navigation of an automated guided vehicle in industrial environments, *Robotics and Computer-Integrated Manufacturing*, 26(4), 296-311.
- Massow, K., Kwella, B., Pfeifer, N., Häusler, F., Pontow, J., Radusch, vd., 2016, Deriving HD maps for highly automated driving from vehicular probe data, *Intelligent Transportation Systems (ITSC)*, pp. 1745-1752
- Máttyus, G., Wang, S., Fidler, S., Urtasun, R., 2016, Hd maps: Fine-grained road segmentation by parsing ground and aerial images, *Computer Vision and Pattern Recognition*, pp. 3611-3619
- Microsoft, 2010, Microsoft Kinect, <https://developer.microsoft.com/en-us/windows/kinect>, erişim tarihi: 03.10.2018
- Min, B. K., Cho, D. W., Lee, S. J., Park, Y. P., 1997, Sonar mapping of a mobile robot considering position uncertainty. *Robotics and Computer-Integrated Manufacturing*, 13(1), 41-49.
- Murray, D., Jennings, C., 1997, Stereo vision based mapping and navigation for mobile robots, *Robotics and Automation*, Vol. 2, pp. 1694-1699.
- Riisgaard S., Blas M.F., 2005, SLAM for Dummies A Tutorial Approach to Simultaneous Localization and Mapping, [https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam\\_blas\\_repo.pdf](https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf), erişim tarihi: 03.10.2018
- Sankrit, H., Panwala, B. J., Mudgal, P., Patil, C. G., 2016, Indoor SLAM using Kinect Sensor, *International Journal of Science Technology and Engineering (IJSTE)*, 2(10), 1226-1231.
- Se, S., Lowe, D., Little, J., 2001, Vision-based mobile robot localization and mapping using scale-invariant features, *Robotics and Automation*, Vol. 2, pp. 2051-2058
- Se, S., Lowe, D., Little, J., 2002, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *The International Journal of Robotics Research*, 21(8), 735-758.
- Seff, A., Xiao, J., 2016, Learning from maps: Visual common sense for autonomous driving, arXiv preprint arXiv:1611.08583

### KAYNAKLAR DİZİNİ (devam)

- Seif, H. G., Hu, X., 2016, Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry, *Engineering*, 2(2), 159-162.
- Siam, M., Elkerdawy, S., Jagersand, M., Yogamani, S., 2017, Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges, *Intelligent Transportation Systems (ITSC)*, pp. 1-8
- Smith, R., Self, M., Cheeseman, P., 1990, Estimating uncertain spatial relationships in robotics, *Autonomous Robot Vehicles* pp. 167-193
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012, A benchmark for the evaluation of RGB-D SLAM systems, *Intelligent Robots and Systems (IROS)*, pp. 573-580
- Thrun, S., 2002, Robotic mapping: A survey, *Exploring Artificial Intelligence in the New Millennium*, 1(1-35), 1.
- Thrun, S., Burgard, W., Fox, D., 2000, A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, *Robotics and Automation, ICRA'00*, Vol. 1, pp. 321-328
- Vardhan, H., 2017, <https://www.geospatialworld.net/blogs/hd-maps-autonomous-vehicles/>, erişim tarihi: 03.10.2018
- Vasiljević, G., Miklić, D., Draganjac, I., Kovačić, Z., Lista, P., 2016, High-accuracy vehicle localization for autonomous warehousing, *Robotics and Computer-Integrated Manufacturing*, 42, 1-16.
- Wolf, D. F., Sukhatme, G. S., 2005, Mobile robot simultaneous localization and mapping in dynamic environments, *Autonomous Robots*, 19(1), 53-65.
- Yazıcı A., 2016, Endüstri 4.0 ve Otonom Robotlar, *Elektrik Mühendisliği*, 459, s.38
- Yu, W., Amigoni, F., 2014, Standard for Robot Map Data Representation for Navigation, *International Conference on Intelligent Robots and Systems, Workshop on "Standardized Knowledge Representation and Ontologies for Robotics and Automation"*
- Zhao, J., Liang, B., Chen, Q., 2018, The key technology toward the self-driving car, *International Journal of Intelligent Unmanned Systems*, 6(1), 2-20.
- Zheng, S., Wang, J., 2017, High definition map-based vehicle localization for highly automated driving: Geometric analysis, *Localization and GNSS (ICL-GNSS)*, pp. 1-8