

İç Ortamlarda Anlamsal Tabanlı Keşif Algoritmalarının Geliştirilmesi

Burak Kaleci

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ağustos 2016



Developing Semantic-Based Exploration Algorithms in Indoor Environments

Burak Kaleci

DOCTORAL DISSERTATION

Electrical and Electronics Engineering Department

August 2016

İç Ortamlarda Anlamsal Tabanlı Keşif Algoritmalarının Geliştirilmesi

Burak Kaleci

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Kontrol Kumanda Bilim Dalı
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof.Dr. Osman Parlaktuna

BAP projesi ve TÜBİTAK Yurtiçi Lisansüstü Burs Programı kapsamında desteklenmiştir

Ağustos 2016

ONAY

Elektrik-Elektronik Mühendisliđi Anabilim Dalı DOKTORA öğrencisi Burak Kaleci'nin DOKTORA tezi olarak hazırladığı "**İç Ortamlarda Anlamsal Tabanlı Keşif Algoritmalarının Geliştirilmesi**" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca oybirliđiyle deđerlendirilerek kabul edilmiştir.

Danışman : Prof.Dr. Osman Parlaktuna

İkinci Danışman :

Doktora Tez Savunma Jürisi:

Üye : Prof.Dr. Osman PARLAKTUNA

Üye : Yrd.Doç.Dr. Buđra KOKU

Üye : Yrd.Doç.Dr. Metin ÖZKAN

Üye : Yrd.Doç.Dr. Helin DUTAĞACI

Üye : Doç.Dr. Sırma YAVUZ

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, Prof.Dr. Osman Parlaktuna danışmanlığında hazırlamış olduğum “**İç Ortamlarda Anlamsal Tabanlı Keşif Algoritmalarının Geliştirilmesi**” başlıklı 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. 03/08/2016

Burak Kaleci

ÖZET

Bu çalışmada, bina içi ortamlarda, çok robotlu bir sistem için anlamsal tabanlı keşif algoritmaları geliştirmek amaçlanmıştır. Bina içi ortamlar, zeminin düz olduğu okul, hastane ve iş merkezi gibi yerler olabilir. Bununla birlikte, bu ortamlar, zehirli madde yayılımı, sel baskını ve yangın gibi afetlerden sonra rampa benzeri robotun seyrüseferini zorlaştıran engeller içeren; haberleşmenin kısıtlı olduğu ortamlara dönüşebilmektedir. Bu tez çalışması kapsamında önerilen keşif algoritmalarında her iki tip ortamın özellikleri de göz önünde bulundurulmuştur.

Bu yaklaşımda, ortamın metrik haritası elde edilirken aynı zamanda çevrimiçi olarak topolojik haritası da spektral kümeleme yöntemi kullanılarak elde edilmiştir. Bu noktada, bina içi ortamlarda, oda, koridor ve kapı anlamsal sınıflarının olacağı varsayımı yapılmıştır. Bir noktanın anlamsal sınıfına karar vermek için k-ortalama (k-means) ve öğrenen vektör nicemleme (Learning Vector Quantization) eğitim yöntemleri kullanılmıştır. Geçmiş çalışmalarda, kapı konumlarının bulunmasının zor olduğu sıklıkla dile getirilmiştir. Bununla birlikte, kapı gibi aralıkların tespit edilmesi keşfin verimliliği açısından büyük önem taşımaktadır. Bu tez çalışması kapsamında, kapıların tespit edilmesi için kural tabanlı bir yaklaşım önerilmiştir. Bir konumun, anlamsal sınıfına karar verildikten sonra bu bilgi keşfi hızlandırmak için kullanılmıştır. Bununla birlikte, bu yöntem yardımıyla robotlara oda öncelikli ya da koridor öncelikli gibi roller verilmektedir. Bu sayede, bina içi ortamın önce hangi kısımlarının keşfedileceğine karar verilebilmektedir. Robotların koordinasyonu ve görev paylaşımı için tek turlu çok öğeli müzayedeler içeren market tabanlı bir yaklaşım kullanılmıştır. Koordinasyon ve bilgi paylaşımı yapılırken haberleşme kısıtları göz önünde bulundurulmuştur.

Önerilen yaklaşım, Gazebo benzetim ortamında modellenen ESOGÜ Laboratuvar binasında, hem normal ortam hem de rampalar içeren modeller ile test edilmiştir. Testlerde iki lazer mesafe tarayıcı ve derinlik ve renk algılayıcıya (kinect) sahip P3-AT gezgin robotları kullanılmıştır. Robotları kontrol etmek amacıyla Robot İşletim Sistemi (Robot Operating System, ROS) kullanılmıştır.

Anahtar Kelimeler: Metrik ve topolojik haritalama, spektral kümeleme, anlamsal sınıflandırma, kapı bulma, lazer algılayıcı, derinlik ve renk algılayıcı, seyrüsefer, kısıtlı haberleşme, keşif, ROS.

SUMMARY

The main purpose of this study is to develop semantic-based exploration algorithms for a multi-robot system in indoor environments. The environments such as schools, hospitals, and business centres have smooth surfaces. Yet, these smooth surfaces can be transformed into non-smooth surfaces that makes navigation difficult and may have communication constraints after disasters such as hazardous substance release, flooding, and fire. In this study, properties of both of these environments are considered while developing exploration algorithms.

While generating the metric map of the environment, the topological map is also obtained in an online manner by applying spectral clustering to this metric map. It is assumed that there could be three broad semantic classes: room, corridor, and door in indoor environments. K-means and Learning Vector Quantization methods are used to classify robot locations. In literature, authors frequently stated that detecting door locations is a challenging task. On the other hand, it is important to be aware of openings such as doors for efficiency of the exploration. In this study, a rule-based door detection method is proposed. The semantic classes are used to speed up the exploration. Semantic information also provides us the ability to assign the roles to robots such as room-priority or corridor-priority. By these means, more important part of the indoor environment could be explored first. In order to coordinate the robot team and task allocation, a market-based method which involves single-tour multi-item auctions is used. The communication constraints are considered while coordinating the robots and information sharing.

The proposed method is tested in ESOGÜ Laboratory building which is modelled in Gazebo. Two types of environments are used in experiments. P3-AT robot with two laser scanner and a RGB-D are used in simulations. Robot Operating System (ROS) is used to control robots in simulations.

Keywords: Metric and topological mapping, spectral clustering, semantic classification, door detection, laser range finder, kinect, navigation, communication constraints, exploration.

TEŞEKKÜR

Gerek derslerimde ve gerekse tez çalışmalarında, bana danışmanlık ederek, beni yönlendiren, yüreklendiren ve her türlü olanağı sağlayan, sahip olduğu bilgi ve tecrübe ile bana her zaman yol gösteren değerli danışmanım Prof. Dr. Osman PARLAKTUNA'ya ve değerli eşi İnci PARLAKTUNA'ya teşekkürü bir borç bilirim.

Doktora dönemimde Yurtiçi Lisansüstü Burs Programı kapsamında bana maddi yönden destek olan TÜBİTAK'a teşekkür ederim.

Doktora dönemimde gerek derslerimde, gerek tezimde, gerekse akademik çalışmalarda sağladıkları katkılardan dolayı saygıdeğer hocalarım Yrd. Doç. Dr. Metin ÖZKAN, Yrd. Doç. Dr. Helin DUTAĞACI ve Yrd. Doç. Dr. Uğur GÜREL'e teşekkür ederim.

Tüm yaşamım boyunca bana yürekten inanan, desteğini esirgemeyen ve hoşgörü gösteren sevgili aileme teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
ŞEKİLLER DİZİNİ	xii
ÇİZELGELER DİZİNİ	xv
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	6
3. TEK ROBOTLU KEŞİF YÖNTEMİ	13
3.1. Geçmiş Çalışmalar	13
3.2. Gerçeklenen Yaklaşım	16
3.2.1. Sınır hücre tabanlı keşif yaklaşımı	17
3.2.2. Sistem mimarisi	20
3.2.2.1. Izgara erkini	20
3.2.2.2. Üretici erkin	30
3.2.2.3. Tüketici erkin	33
3.2.2.4. Hareket erkini	33
3.2.2.5. Erkinler arası iletişim	37
3.3. Sonuçlar	39
3.3.1. Robot kontrol arayüzü	39
3.3.2. Benzetim ortamı	40
3.3.3. Deneysel kurulum	42
3.3.4. Deneysel sonuçlar	43
4. TEK ROBOTLU TOPOLOJİK KEŞİF YÖNTEMİ	47
4.1. Geçmiş Çalışmalar	47
4.1.1. Topolojik harita oluşturma yaklaşımları	47
4.1.2. Spektral kümeleme ile topolojik harita oluşturma yaklaşımları	50
4.1.3. Tek robotlu topolojik keşif yaklaşımları	52
4.2. Gerçeklenen Yaklaşım	55
4.2.1. Normalleştirilmiş kesim problemi	55

4.2.2.	Topolojik haritanın oluşturulması	57
4.2.3.	Keşifte topolojik haritanın kullanılması	64
4.3.	Sonuçlar	67
4.3.1.	Deneysel kurulum	67
4.3.2.	Deneysel sonuçlar	68
5.	ÇOK ROBOTLU KEŞİF YÖNTEMİ	76
5.1.	Geçmiş Çalışmalar	77
5.1.1.	Çok robotlu sistemlerde koordinasyon ve keşif yaklaşımları	77
5.1.2.	Çok robotlu sınır hücre keşif yaklaşımları	83
5.1.3.	Haberleşmenin kısıtlı olduğu durumlardaki keşif yaklaşımları	93
5.2.	Gerçeklenen Yaklaşım	104
5.2.1.	Market tabanlı koordinasyon ve görev paylaşırma	104
5.2.2.	Metrik ve topolojik haritaların birleştirilmesi	106
5.2.2.1.	Metrik haritaların birleştirilmesi	109
5.2.2.2.	Topolojik haritaların birleştirilmesi	110
5.2.3.	Çarpışma önleme protokolü	116
5.3.	Sonuçlar	117
5.3.1.	Deneysel kurulum	118
5.3.2.	Deneysel sonuçlar	119
6.	ANLAMSAL TABANLI KEŞİF YÖNTEMİ	129
6.1.	Geçmiş Çalışmalar	130
6.1.1.	Anlamsal sınıflandırma ve haritalama yaklaşımları	130
6.1.2.	Kapı bulma yaklaşımları	134
6.1.3.	Anlamsal sınıflandırma tabanlı keşif yaklaşımları	136
6.2.	Gerçeklenen Yaklaşım	138
6.2.1.	Yönelim normalizasyonu	140
6.2.2.	K-Ortalama (KO) ile sınıflandırma	143
6.2.3.	Öğrenen vektör nicemleme (ÖVN) ile sınıflandırma	143
6.2.4.	Markov model	145
6.2.5.	Kural tabanlı kapı bulma yaklaşımı	146
6.2.6.	Anlamsal sınıflandırma tabanlı keşif yaklaşımı	150
6.3.	Sonuçlar	155
6.3.1.	Freiburg 79 test verisi sonuçları	155
6.3.2.	Freiburg 79 yörünge verisi sonuçları	157
6.3.3.	ESOGÜ yörünge verisi sonuçları	159
6.3.4.	Kural tabanlı kapı bulma algoritması parametre analizi	162
6.3.5.	ESOGÜ yörünge verisi anlamsal sınıflandırma ve kapı bulma sonuçları	166

6.3.6. ESOĞÜ tek robotlu anlamsal sınıflandırma tabanlı keşif sonuçları .	168
6.3.7. ESOĞÜ çok robotlu anlamsal sınıflandırma tabanlı keşif sonuçları .	170
7. AFET ORTAMINDA ANLAMSAL TABANLI KEŞİF YÖNTEMİ	177
7.1. Geçmiş Çalışmalar	177
7.2. Gerçeklenen Yaklaşım	179
7.3. Sonuçlar	183
7.3.1. Deneysel Kurulum	183
7.3.2. Tek robotlu afet ortamı keşif sonuçları	183
7.3.3. Çok robotlu afet ortamı keşif sonuçları	187
8. SONUÇLAR VE ÖNERİLER	188
KAYNAKLAR DİZİNİ	190
Özgeçmiş	212

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1 Sarı test alanı örnekleri (NISTTestAlanları, 2016)	2
1.2 Turuncu test alanı örnekleri (NISTTestAlanları, 2016)	2
1.3 Kırmızı test alanı örnekleri (NISTTestAlanları, 2016)	3
3.1 Sınır Kenar Hücrelerin Belirlenmesi	17
3.2 Sınır Hücrelerin Belirlenmesi	19
3.3 Sistem Mimarisi	20
3.4 Eğimin birden küçük olduğu durumda DDA algoritmasının sonucu	24
3.5 Eğimin birden büyük olduğu durumda DDA algoritmasının sonucu	24
3.6 Koordinat Sistemleri	25
3.7 Izgara erkini algoritmasının sonuçları	28
3.8 Standart A* Algoritması	34
3.9 Genişletilmiş A* Algoritması	35
3.10 Erkinler Arası İletişim (Sıralama Diyagramı)	38
3.11 Test ortamı (Gazebo)	42
3.12 En çok kenar hücrelerine sahip sınır hücre kriteri	44
3.13 En çok kenar hücrelerine sahip sınır hücre örneği	44
3.14 En yakın sınır hücre kriteri	45
3.15 En yakın sınır hücre kriteri örneği	45
3.16 Market sınır hücre kriteri	46
4.1 Kesim ve en küçük kesim örnekleri (Cut, 2016)	55
4.2 Metrik Harita Örneği	58
4.3 İlginlik Matrisi Örneği	58
4.4 Çok fazla sayıda küme örneği	59
4.5 Çok az sayıda küme örneği	60
4.6 Örnek metrik harita için özvektörler	61
4.7 Uyarlamalı düğüm oluşturma örneği	63
4.8 Uyarlamalı yöntem ile elde edilen kümeleme sonuçları	64
4.9 Uyarlamalı yöntem ile elde edilen topolojik harita	65
4.10 Topolojik Harita ile Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)	66
4.11 Test ortamı (Gazebo), Metrik ve Topolojik Harita Örneği	67
4.12 A* en çok kenar hücrelerine sahip sınır hücre kriteri	69
4.13 Dijkstra en çok kenar hücrelerine sahip sınır hücre kriteri	70
4.14 A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (1)	70

4.15 A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (2)	71
4.16 A* en yakın sınır hücre kriteri	72
4.17 Dijkstra en yakın sınır hücre kriteri	72
4.18 A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (3)	73
4.19 A* market sınır hücre kriteri	74
4.20 Dijkstra market sınır hücre kriteri	74
4.21 Dijkstra yol planlarının hedef seçimi üzerindeki etkisi	75
5.1 Çok Robotlu Sistem Mimarisi	105
5.2 Çok Robotlu Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)	107
5.3 Harita birleştirme aşamaları (1)	111
5.4 Harita birleştirme aşamaları (2)	112
5.5 Harita birleştirme aşamaları (3)	113
5.6 Harita birleştirme aşamaları (4)	114
5.7 Harita birleştirme aşamaları (5)	115
5.8 Çok Robot Metrik ve Topolojik Harita Örnekleri	118
5.9 A* en çok kenar hücresine sahip sınır hücre kriteri	121
5.10 Dijkstra en çok kenar hücresine sahip sınır hücre kriteri	121
5.11 Robotların $hMenzil$ dışında olduklarında hedef seçimi	122
5.12 Robotların $tHarita$ parametresi dolmadan hedef seçimi	123
5.13 A* en yakın sınır hücre kriteri	124
5.14 Dijkstra en yakın sınır hücre kriteri	124
5.15 Robotların $hMenzil$ içinde olduklarında hedef seçimi	125
5.16 Robotların $hMenzil$ dışında olduklarında hedef seçimi	126
5.17 A* market sınır hücre kriteri	127
5.18 Dijkstra market sınır hücre kriteri	127
6.1 Üç anlamsal sınıf: Oda, Koridor ve Kapı (Mozos vd. 2005).	129
6.2 Oda, Koridor ve Kapı sınıflarından lazer verilerine örnekler (Mozos vd. 2005).	139
6.3 Aynı konumda farklı yönelimde iki robot lazer okuması örnekleri	140
6.4 Örnek 1 ve Örnek 2 için yerel ve indekse göre lazer verileri	141
6.5 Yönelim Normalizasyonu	142
6.6 K-Ortalama yöntemi için küme merkezi örnekleri	144
6.7 ÖVN yöntemi için küme merkezi örnekleri	145
6.8 Yatay kapıda farklı yönelimlerle lazer ölçümleri	147
6.9 Dikey kapıda farklı yönelimlerle lazer ölçümleri	148
6.10 Kapı bulma için tanımlanan parametreler	149
6.11 Yatay ve dikey kapı durumları	149
6.12 Anlamsal Tabanlı Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)	153

6.13 Çok Robotlu Anlamsal Tabanlı Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)	154
6.14 Freiburg 79 gerçek referans verisi	155
6.15 ÖVN Görsel Sonuç	157
6.16 Düğüm Ağı	157
6.17 ÖVN Görsel Sonuç	159
6.18 ESOGÜ gerçek referans verisi	159
6.19 KO K =1000, EO için görsel sonuç	161
6.20 ÖVN K =1000 için görsel sonuç	161
6.21 W parametresinin etkisi	162
6.22 T_H parametresinin etkisi	163
6.23 T_E parametresinin etkisi	163
6.24 Freiburg 79 $W = 4, T_H = 1, T_E = 1.5$ için görsel sonuç	165
6.25 ESOGU $W = 15, T_H = 1.2, T_E = 2$ için görsel sonuç	166
6.26 KO K =40, V için görsel sonuç	166
6.27 ÖVN K =1000 için görsel sonuç	167
6.28 Anlamsal sınıflandırma bilgisini hedef seçme kriteri Konum (1)	169
6.29 Anlamsal sınıflandırma bilgisini hedef seçme kriteri Konum (2)	169
6.30 Konum 1 R1 ve R2 oda öncelikli kriteri	171
6.31 Konum 1 R1 koridor ve R2 oda öncelikli kriteri	172
6.32 Konum 2 R1 ve R2 oda öncelikli kriteri	173
6.33 Konum 2 R1 oda ve R2 koridor öncelikli kriteri	173
6.34 Konum 3 R1 ve R2 oda öncelikli kriteri	174
6.35 Konum 3 R1 koridor ve R2 oda öncelikli kriteri	175
7.1 Sarı, turuncu ve kırmızı test alanları	179
7.2 Robotun bakış açısına göre rampaların sınıflandırılması	181
7.3 Afet sonrası test ortamı	184
7.4 Afet ortamı anlamsal sınıflandırma bilgisi hedef seçme kriteri Konum (1)	186
7.5 Afet ortamı anlamsal sınıflandırma bilgisi hedef seçme kriteri Konum (2)	186

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1 Tek Robot A* Sonuçları	43
4.1 Tek Robot A* ve Dijkstra Sonuçları	68
5.1 İki Robot A* ve Dijkstra Sonuçları	120
5.2 Çarpışma Önlem Protokolü A* ve Dijkstra Sonuçları	128
6.1 Test Verisi Sonuçları	156
6.2 Yörünge Verisi Sonuçlar	158
6.3 ESOGÜ Yörünge Verisi Sonuçları	160
6.4 Kapı Bulma Algoritması Freiburg 79 Sonuçları	164
6.5 Kapı Bulma Algoritması ESOGÜ Sonuçları	164
6.6 ESOGÜ Yörünge Verisi Kapı Bulma ve Anlamsal Sınıflandırma Sonuçları . . .	167
6.7 Tek Robot Anlamsal Sınıf Bilgisi Sonuçları	168
6.8 İki Robot Anlamsal Sınıf Bilgisi Sonuçları	170
6.9 Çarpışma Önlem Protokolü Anlamsal Sınıf Bilgisi Sonuçları	176
7.1 Afet Ortamında Tek Robot Sonuçları	185
7.2 İki Robot Sonuçları	187

1. GİRİŞ

Keşif problemi, bilinmeyen bir ortamın içine doğru hareket ederken aynı zamanda daha sonraki seyrüsefer hareketleri için kullanılacak bir harita yapmaktır (Yamauchi, 1997). Bu noktada, geçmiş çalışmalarda genellikle metrik ve topolojik haritalar hedef seçimlerinde kullanılmıştır. Bununla beraber, son yıllarda robotun etrafını tanımasını, bulunduğu konumu başka konumlardan anlamsal olarak ayırmasını amaçlayan çalışmalar yapılmaya başlanmıştır. Robotun bu anlamsal konumlardan hangisinde olduğuna karar vermesi keşif probleminin çözümünde verimliliği artırabilir.

Bu çalışmada, bina içi ortamlarda, çok robotlu bir sistem için anlamsal tabanlı keşif algoritmaları geliştirmek amaçlanmıştır. Bina içi ortamlar, zeminin düz olduğu okul, hastane ve iş merkezi gibi yerler olabilir. Bununla birlikte, bu ortamlar, zehirli madde yayılımı, sel baskını ve yangın gibi afetlerden sonra rampa benzeri robotun seyrüseferini zorlaştıran engeller içeren; haberleşmenin kısıtlı olduğu ortamlara dönüşebilmektedir. Bu tez çalışması kapsamında önerilen keşif algoritmalarında her iki tip ortamın özellikleri de göz önünde bulundurulmuştur.

Bu noktada, bu tez çalışması kapsamında kullanılan afet sonrası bina içi ortamların tanımlarını yapmak ve sahip olduğu özellikleri belirtmek faydalı olacaktır. Jacoff ve diğerleri geliştirilen yöntemlerin standart bir değerlendirmeden geçmesini sağlamak amacıyla zorluk dereceleri farklı üç test ortamı belirlenmiştir (Jacoff vd. 2003). Sarı test alanı, yer robotlarının seyrüseferi için en basit olan alandır. Bu alanda, 10 derecelik rampalar, engeller ve afetzedeler bulunmaktadır. Ortamın büyük bir kısmı düzgün zeminden ya da koridoru tamamen kaplayan rampalardan oluşmaktadır. Bu ortam, geliştirilen keşif, haritalama, koordinasyon ve algılama problemlerinin hareket yeteneği çok gelişmiş robotlara ihtiyaç duymadan test edilmesi için uygundur. Şekil 1.1a ve 1.1b'de sarı test alanı için örnekler verilmiştir.

Turuncu test alanı hareket ve algılama yeteneği daha gelişmiş robotlar için tasarlanmıştır. Zemin farklı tipte malzemelerden oluşmaktadır. Bununla birlikte, test alanında rampa ya da merdivenle ulaşılabilen yükseltilmiş bir kısım da bulunmaktadır. Bu yükseltilmiş kısımda bulunan boşluk robotun hareket yeteneklerini zorlamaktadır. Bu test alanında rampalar 15 derece eğime sahiptir. Şekil 1.2a ve 1.2b'de turuncu test alanı için örnekler verilmiştir.



(a) Sarı test alanı örnek 1



(b) Sarı test alanı örnek 2

Şekil 1.1: Sarı test alanı örnekleri (NISTTestAlanları, 2016)



(a) Turuncu test alanı örnek 1



(b) Turuncu test alanı örnek 2

Şekil 1.2: Turuncu test alanı örnekleri (NISTTestAlanları, 2016)

Kırmızı test alanı ise hareket ve algılama yeteneği açısından en zorlu alandır. Bu alan esasında boru, plastik poşet, çakıl ve çelik kablolardan oluşan birikintinin oluşturduğu yükseltilerden oluşmaktadır. Bu durum robotun hareket ve algılama yeteneklerini oldukça zorlamaktadır. Ayrıca, ağır bir robotun üstünden geçmesi durumunda çökebilecek durumda olan kararsız zemin bölgeleri bu alanda yer almaktadır. Bununla birlikte, 20cm yüksekliğinde toplam 5 tane merdiven bulunmaktadır. Şekil 1.3a ve 1.3b'de kırmızı test alanı için örnekler verilmiştir.

Bu tez çalışması kapsamında, sarı test alanı kullanılmıştır. Bu ortamın kullanılmasının sebebi ise, geliştirilen keşif ve koordinasyon problemlerinin hareket yeteneği çok gelişmiş robotlara ihtiyaç duymadan test edilmesi için uygun olmasıdır. Tez kapsamında, özellikle keşif problemi ele alındığı için hareket yeteneği ile ilgili durumlar



(a) Kırmızı test alanı örnek 1



(b) Kırmızı test alanı örnek 2

Şekil 1.3: Kırmızı test alanı örnekleri (NISTTestAlanları, 2016)

incelenmemiştir. Problemin tanımı yapılırken kullanılan diğer varsayımlar aşağıda verilmiştir:

- Robotlar tektüredir ve aynı kaynaklara sahiptirler,
- Robotlar arası haberleşme belirlenen bir menzil içinde mevcuttur,
- Robotlar 360 derece tarama yapabilen lazer mesafe algılayıcı, renk ve derinlik (Kinect) algılayıcıya sahiptirler,
- Robotlar koordinasyon, görev paylaşımı, haritalama ve keşif gibi görevleri yapabilecek hesaplama gücüne sahiptirler,
- Robotlar eğimli düzlemlerde seyrüsefer yapabilmektedir,
- Keşif faaliyeti zaman ya da alana bağlı olmadan robotlar bütün ortamın keşfini tamamlayana sürdürülmektedir.

Bu varsayımlar altında, robotlar hareket edebilmek için gerekli olan bilgiyi, derinlik algılayıcılarından edindikleri veriyi kullanarak, birbirleriyle etkileşim içinde baştan oluşturdukları bir haritadan çıkaracaktır. Bunun başarılabilmesi için, çok robotlu sistemlerde keşif, seyrüsefer, koordinasyon ve haberleşme alt görevlerinin verimli yaklaşımlarla gerçekleşmesi ve bütün bu alt görevlerin haritalama görevi içinde birleştirilmesi gerekmektedir. Bununla birlikte, zeminin düzgün olmaması ve haberleşmenin kısıtlı olması gibi etkenler algılayıcı verilerindeki gürültünün artması neden olacaktır. Bu durumların da göz önünde bulundurulması faydalı olacaktır.

Çok robotlu sistemlerde keşif için genellikle ızgara (grid) tabanlı yaklaşımlar kullanılmaktadır. Bu yaklaşımlarda, olası hedeflerin belirlenmesi aşamasında genellikle sınır hücre (frontier) yöntemi kullanılmıştır. Olası hedefler arasından seçim yapılırken en yakın, en çok bilgi içeren ve hem hedefin yakınlığını hem de kazanılacak olan bilgiyi göz önüne alan kriterler kullanılmaktadır. Bu tez çalışması kapsamında, hedefin bulunduğu noktanın anlamsal sınıfını (oda, koridor ve kapı gibi) kullanarak hedef seçimi yapılması sağlanmaktadır. Robotun bulunduğu konumun anlamsal sınıfını belirlemek amacıyla, 360 derece tarama yapabilen lazer mesafe algılayıcı ölçümleri K-ortalama ve Öğrenen Vektör Nicemleme (ÖVN) (Learning Vector Quantization) gibi eğitim yaklaşımlarına girdi olarak verilmektedir. Daha sonra, bu bilgi Markov model ile tekrar değerlendirilerek hedefin anlamsal sınıfına karar verilmektedir.

Robotlar belirledikleri hedeflere gitmek amacıyla A* ve Dijkstra algoritmalarını kullanarak yol planı yapmışlardır. Seyrüsefer sırasında robotun eğimli düzlemlere çarparak devrilmesini önlemek amacıyla renk ve derinlik algılayıcıdan (Kinect) faydalanarak yerel ara noktalar belirlenmiştir. Robotların ortamdaki engellerden sakınmaları için Vektör Alan Histogramı (Vector Field Histogram) yaklaşımı kullanılmıştır. Robotların birbirlerine çarpmalarını önlemek için ise bir protokol gerçekleştirilmiştir. Robotlar ortamı keşfederken hem metrik hem de topolojik harita üretmişlerdir. Ayrıca, ortamla ilgili anlamsal bilgiler de topolojik haritanın düğümlerinde yer almıştır.

Robot takımının bu görevleri koordinasyon içinde yapması verimliliği artıracaktır. Robotların koordinasyonunu sağlamak için sürekli bir arada olmaları bir çözüm yolu olabilir. Ancak bu durumda, ortama yayılamadıkları için keşif süresi uzayabilir. Robotların birbirleriyle hiç haberleşmeden keşfi yapmaları da çok robotlu sistemlerin sağladığı avantajları azaltabilir. Bu durumda, en uygun çözüm robotların belirli anlarda buluşarak haberleşmeleri olabilir. Ancak, robotların keşif yapmayı bırakarak diğer robotlarla haberleşmek için bir konuma gitmesi de keşif süresini uzatacaktır. Bu tez çalışması kapsamında, robotlar belirli zamanlarda bir araya gelmektedirler ancak bu durum robotlar keşfi sürdürürken gerçekleşmektedir. Robotlar arası koordinasyonun sağlanması için market ve müzayede tabanlı gelişmiş yöntemler kullanılmaktadır. Robotların yerel haritalarının birleştirilmesinde geçmiş çalışmalarda yer alan anlaşma/uyuşmazlık fonksiyonlarına benzer bir yaklaşım önerilmiştir. Topolojik haritaların birleştirilmesi sırasında uygulanan yaklaşım ile topolojik haritanın daha az düğümlerle ortamı daha iyi temsil etmesi sağlanmıştır.

Bu tez çalışmasının temel katkısı, hedeflerin seçiminde anlamsal sınıfların kullanılmasıdır. Bu sayede, robot takımındaki üyelere oda öncelikli, koridor öncelikli gibi roller verilebilmekte ve robotların verimliliği önemli ölçüde artırılmaktadır. Bununla birlikte, keşif sırasında robotların izleyecekleri keşif stratejisini de robotların rollerini

seçerek belirleme şansı elde edilmektedir. Bu, özellikle afet sonrası bina içi ortamlarda kritik olmaktadır. Robot takımının binayı keşfe başlarken aldığı durum (bütün robotlar aynı yerden ya da ayrı ayrı yerlerden), binanın geometrik yapısı, afetin olduğu saat gibi kriterlerin göz önüne alınması için anlamsal sınıflandırma ile keşif yapmak uygun olabilir. Robotlar bu kriterler göz önüne alınarak belirlenen roller ile istenilen noktalara yönlendirilebilir.

Tezin ilerleyen bölümlerinde şu konular yer almaktadır: Bölüm 2'de afet sonrası bina içi ortamlarda keşif problemi için geçmiş çalışmalarda önerilen yaklaşımlar verilmektedir. Bölüm 3 ve Bölüm 4'te sırasıyla tek robotlu metrik ve topolojik keşif yöntemleri anlatılmaktadır. Bölüm 5'te çok robotlu keşif yöntemi verilmektedir. Bölüm 6'da anlamsal bilgi tabanlı keşif yöntemi yer almaktadır. Bölüm 7'de afet ortamında keşif anlatılmaktadır. Bölüm 8'de ise yapılan tez çalışması ile ilgili sonuç ve öneriler verilmektedir.

2. LİTERATÜR ARAŞTIRMASI

Normal ortamlarda keşif yaklaşımları ile ilgili geçmiş çalışmalar her bölümün başında verilecektir. Bu bölümde sadece geçmiş çalışmalarda yer alan afet ortamındaki keşif ve koordinasyon yöntemleri ele alınacaktır.

Afet ortamlarında robotların kullanılması fikri ilk olarak 2000'li yılların başında ortaya atılmıştır. Casper ve diğerleri, zehirli madde yayılımı, sel baskını, binaların yıkılması ya da binada bomba olması gibi durumları afet olarak tanımlamışlardır. Afet sonrası bina içi ortamlarda gezgin robotların hangi görevleri yapabileceğini ortaya koymuşlardır. Bu görevlerde robotun hareket kabiliyeti, suya dayanıklılığı, boyutları ve haberleşme gibi kısıtların olduğunu belirtmişlerdir. Bu ortamlarda, robotların algılayıcı olarak kamera kullanmalarının uygun olmadığını, lazer mesafe algılayıcısı ve ısı algılayıcıları gibi algılayıcıların kullanılması gerektiğini gerekçeleri ile açıklamışlardır (Casper vd. 2000).

Araştırmacılar robot takımlarının otonomluğunu artırmaya yönelik yöntemler geliştirmektedirler. Aynı zamanda, afet sonrası bina içi ortamlarda keşif, haritalama ve afetzedelerin bulunmasında kolaylıklar sağlaması beklenen yeni algılayıcılar da piyasada yer almaya başlamıştır. Ancak geliştirilen algoritmaların ya da fiziksel teknolojilerin tecrübe edilmesi gerekmektedir. Bununla birlikte, bahsedilen afet sonrası bina içi ortamlar ile sıklıkla karşılaşmamaktadır. Bu ortamların laboratuvarında oluşturulması da hem maliyetli hem de zor bir iştir. Bu duruma bir çözüm getirmek için RoboCup organizasyonu RoboCup Rescue projesini geliştirmiştir. Bu proje ile çok erkinli sistemlerin çeşitli seviyelerde koordinasyonunu sağlamak, arama ve kurtarma görevleri için özelleşmiş robot takımları, kişisel dijital asistanlar, standart bir benzetim ortamı ve karar destek üniteleri geliştirmek amaçlanmaktadır. RoboCup Rescue ligleri kapsamında, arama kurtarma stratejilerinin ve robotik sistemlerin belirlenmiş standartlar ile değerlendirilmesi, aynı zamanda, bahsedilen sistemler gibi çok farklı alanlarda uzmanlık gerektiren çalışmaları gerçekleştiren araştırmacıları bir araya getirerek oluşturulan sistemlerin birleştirilmesini kolaylaştırmak istenmektedir (Kitano ve Tadokoro, 2001). Bu amaçla her sene yarışmalar düzenlenmektedir. Düzenlenen yarışmalarla ilgili bilgiler aşağıda verilmektedir (RoboCupRescue, 2016).

- **Rescue Robot Yarışması**

Yarışma gerçek robotlar kullanılarak gerçekleştirilmektedir. Robotların sürüş sistemleri, algılama, planlama ve haritalama yetenekleri afet bölgelerinde test edilmektedir (RoboCupRescueRobotLeague, 2016).

- **Rescue Simulation Yarışmaları**

Bu yarışma iki amaca hizmet etmektedir. İlki, arama kurtarma sistemlerindeki robotların ve akıllı sistemlerin geliştirilmesidir. İkincisi ise, afet ortamlarını bütün detayları ile yansıtabilen standart bir benzetim ortamı elde etmektir.

- **Agent Simulation Yarışması**

Agents Simulation projesinin temel amacı afet ortamlarında arama yapabilen ve altyapı problemleri ile ilgilenen erkinler geliştirmektir. Yarışmada başarılı olabilmek için afet bölgelerinde, polis güçleri, itfaiye erleri ve sağlık ekiplerinin üstleneceği görevler erkinler tarafından yapılmalıdır. Ayrıca bu yarışma kapsamında benzetim platformuna da katkılar yapılmaktadır (RoboCupRescueSimulationLeague, 2016).

- **Virtual Robots Yarışması**

The Virtual Robots yarışmasının amacı erkin yarışmaları ve Robocup Rescue Robots ligindeki yarışmalar arasındaki bağlantıyı sağlamaktır. Bu yarışma USARSim adı verilen benzetim ortamında yapılmaktadır. Bu benzetim ortamında çok çeşitli robot sistemleri, algılayıcılar ve eyleyiciler gerçek parametrelerine çok yakın şekilde tanımlanabilmektedir. Bununla birlikte, karmaşık deprem sonrası bina içi ortamlar bu benzetim ortamında yapılabilmektedir. Robotlar, algılayıcıları ile birlikte bu benzetim ortamında yaratılabilmekte ve 3 boyutlu oyun motoru sayesinde robotlar istenilen gürültüler verilerek çalıştırılabilmektedir. Geliştirilen yaklaşımlar ve sistemler bu benzetim ortamı aracılığıyla afet ortamlarında test edilmekte ve gelecek çalışmalara yön vermektedir (Carpin vd. 2007).

Bu noktada, geçmişte yapılan Virtual Robots yarışmalarını incelemek gelecekte yapılacak olan çalışmalara yön verici olacaktır. Bu yüzden, bu yarışmaların değerlendirme raporları ve ilgili yayınlara kısaca yer verilecektir. Bu kapsamda yarışma ilk olarak 2006 yılında Almanya'da yapılmıştır. Bu yarışmanın detaylı raporu (Balakirsky vd. 2007b)'da verilmektedir. Afetzedelerin bulunması ve onların sağlık durumlarının belirlenmesi bu yarışmanın en temel amacıdır. Bu kapsamda afetzedelerin yerlerinin ve durumlarının tespit edilmesi ve keşfedilen yerlerin detaylı haritasının çıkarılması görevleri göz önüne alınarak değerlendirme ölçütleri belirlenmiştir. Yarışmanın birincisi olan Rescue Robots Freiburg

robot platformu olarak Zerg kullanmaktadır. Bu robot platformu ayrıca RFID çipleri, etiket okuyucu yazıcıları ve atalet ölçme ünitesinden oluşmaktadır. Bu takımın temel amacı hesaplama ve haberleşme yükünü artırmadan haritalama ve keşif yapmaktır. Bunu başarmak amacıyla iki aşamalı bir yaklaşım geliştirmişlerdir. İlk olarak, robotun kendi küçük çevresinde ızgara tabanlı keşif ve planlama yapması sağlanmıştır. Küresel koordinasyon, seyrüsefer ve konumlandırma ise RFID çipleri sayesinde gerçekleştirilmektedir. İkinci olan Virtual IUB robot platformu olarak P3-AT kullanmaktadır. Bu robot platformu ayrıca lazer uzaklık ölçer, kamera ve atalet ölçme ünitesinden oluşmaktadır. Virtual IUB takımı afet bölgesindeki zaman kısıtlarından dolayı kurtarma görevleri boyunca robotlar arasında herhangi bir koordinasyon yapmaya çalışmamaktadır. Bu takımın temel amacı yeni algoritmalar geliştirmek yerine daha önce geliştirilmiş yöntemleri tek ve çalışan bir sistem içerisinde birleştirmektir.

2007 yılında Atlanta'da yapılan Virtual Robots yarışmasının temel amacı 20 dakika olan robot pil ömrü tükenmeden önce mümkün olduğunca çok afetzede bulmak ve afetzedelerin olmadığı alanları mümkün olduğunca geniş bir şekilde belirlemektir (Balakirsky vd. 2007a). 2008 yılında Suzhou'da yapılan Virtual Robots yarışması afet bölgelerinde haritalama görevini öne çıkarmıştır. Yarışmaya katılan tüm takımların amacı hem robotlar hem de ilk yardımcıların kullanabileceği bir harita oluşturmaktır. Robotlar için en önemli husus hareket edebilmek için gerekli olan bilgiyi haritadan elde etmektir. İlk yardımcıları ise afetzedelere kendilerini riske atmadan hızlı bir şekilde ulaşmak için harita kullanmaktadırlar. Dolayısıyla, ilk yardımcıları için robotlar tarafından üretilmiş ızgara tabanlı haritalar çok kullanışlı olmamaktadır. Onlar afetzede siyah arabanın yanında gibi ayırt edici işaretlere ya da afetzede soldaki koridorda üçüncü kapının yanında gibi topolojik bilgilere ihtiyaç duymaktadırlar. Dolayısıyla bu değişik istekleri karşılamak amacıyla çok katmanlı haritalar oluşturulurken afetzedelerin konumları ve robotların izledikleri yollar gibi ek bilgiler de tutulmaktadır. Takımların ürettiği haritaların bazı kriterleri sağlıyor olması gerekmektedir. Bunlardan biri ortamdaki duvarların, engellerin ve boş alanların doğru şekilde haritada yer almasıdır. Ayrıca, takımlar keşfedilmiş alanları gri, afetzedelerden arınmış bölgeleri yeşil ve afetzede konumlarını kırmızı çarpı ile göstermek zorundadırlar. Afetzelerden arınmış bölge, belirtilen bölgede bütün afetzedelerin tespit edilmiş olması olarak tanımlanmaktadır. Son olarak, takımlar haritalardaki özellikleri çıkarabilmelidirler. Örneğin, bir duvar kümesi birleştirilerek oda olarak haritada gösterilmelidir (Balaguer vd. 2009).

2009 yılında Graz'da yapılan Virtual Robots yarışmasında geçmiş yıllardan farklı olarak otonom olarak üretilmiş harita kalitesi ve haberleşme alanlarında temel testler yapılmıştır. Geçmiş yıllarda yarışma birden fazla arama kurtarma senaryosunun yer aldığı turlardan oluşmaktaydı. Takımlar çok erkinli sistemlerde koordinasyon, insan arayüzü,

haritalama ve afetzedelerin bulunması gibi yeteneklerden puan alarak sıralanmaktaydı. Bu değerlendirme yöntemi yarışmanın kazananını belirlemede yararlı olsa dahi bir takımın neden kazandığı konusunda detaylı bilgi vermemektedir. 2009 senesinde yapılan yarışmada en iyi haritalama, en iyi insan arayüzü, en iyi haberleşme stratejisi ve tüm bu yeteneklerde en iyi olan takımların belirlenmesi amaçlanmıştır. Haritalama temel testinde, ilk yardımcıları binaya girmeden önce robotların pil ömrü olarak kabul edilen 20 dakikalık sürede ortamın haritalanması gerekmektedir. Takımlar en fazla 4 robot kullanabilmektedirler. Haberleşme temel testinde, robotlara birer sinyal güçlendirici takılarak kesintisiz haberleşme sağlayan en fazla alan elde edilmeye çalışılmaktadır (Balakirsky vd. 2009).

2011 yılında İstanbul'da yapılan Virtual Robots yarışmasında MRL takımı eş zamanlı konumlandırma ve haritalama problemi için kendi geliştirdikleri bir tarama eşleştirme yöntemi kullanmışlardır. Ortamı göstermek amacıyla 3 katmanlı bir metrik harita kullanmışlardır. Bu haritanın ilk katmanı lazer mesafe algılayıcıdan gelen bilgiler ile oluşturulurken kalan iki katmanı ise görsel ve diğer algılayıcılardan gelen bilgiler ile meydana gelmektedir. Elde edilen bu metrik harita yol planı ve öğrenme gibi birçok amaç için kullanılmıştır. Pasargad takımı hem metrik hem de topolojik harita üretmiştir. Topolojik haritanın elde edilmesi için metrik haritada bulunan köşe özniteliklerine sahip tüm olası konumlar sanal kapı olarak kabul edilmiştir. Daha sonra, sanal kapılar ayırıtılarak kabul edilerek düğümler belirlenmiş ve Genetik algoritma kullanılarak düğümler birleştirilmiştir. Keşif yapılırken robotlar üç durumda bulunmaktadır: otonom, yarı otonom ve operatör kontrolü. Robotlar keşfe başlarken operatör ortama yayılmaları için gerekli olan hedefleri atamakta ve otonom durumuna getirmektedir. Robotlar keşif yaparken, operatör robotlardan gelen görsel verileri izlemekte ve yakında bir afetzede varsa kontrolü ele almaktadır. SEU_RedSun yol planı olarak duvarı izleme yaklaşımını gerçekleştirmiştir. YILDIZ üniversitesinin takımı Yıldız, ilk olarak, tek robot için konumlandırma, haritalama, seyrüsefer ve afetzede tanıma görevlerini tek robotun başarılı bir şekilde yapmasını amaçlamışlardır. Daha sonra bu yaklaşımları çok robot için genişletmeyi planlamışlardır. Bir yer robotu ve bir hava aracı kullanmışlardır. Üç katmanlı bir harita üretmişlerdir. Metrik, afetzede ve seyrüsefer katmanları bulunmaktadır. Lazer algılayıcı bilgileri hedef seçiminde kullanılmaktadır (VirtualRobotCompetition, 2016).

2012 yılında Mexico City'de yapılan yarışmada Yıldız takımı ikinci olmuştur. Robotların kullanıcı tarafından belirlenen hedeflere gidebilmesi için sağ ve sol tekerlere uygulanacak olan hızlar için basit bir kontrolör kullanılmaktadır. Robotların haberleşmesini sağlamak amacıyla bütün robotlara mesajlar yayınlanmaktadır. Robot gelen mesajın kendine gelip gelmediğini kontrol etmektedir. Eğer başka robota ait bir mesaj geldiyse mesajı alan robotta bütün robotlara mesajlar yayınlanmaktadır. PoARet takımı yarışmada birinci olmuştur. P3-AT yer robotlarını 360 derece tarama yapabilen lazer mesafe

algılayıcıları ve bir kamerası ile yarışmada kullanmışlardır. Sınır hücre tabanlı yaklaşım hedeflerin belirlenmesinde kullanılmıştır. Bununla birlikte, otonomluğu artırmak amacıyla anlamsal bilgilerden de yararlanılmıştır. Kapı ve oda olmak üzere iki sınıf tanımlanmıştır. Bir sonraki hedefin belirlenmesinde, hedefe olan mesafe, kazanılacak bilgi, haberleşmenin kopma olasılığı, enerji seviyesi ve anlamsal bilgi gibi çok kriterli bir karar verme süreci işletilmektedir. Robotların yapmakla yükümlü olduğu üç farklı görev bulunmaktadır. İlk olarak keşif görevi, lazer algılayıcılar ile yapılmaktadır. Afettede arama görevi keşfedilmiş bölgede yapılmaktadır. Robot bir bölgeyi lazer ile keşfettiğinde kameranın görüş alanı yüzünden bu alandaki her noktaya bakmak gerekmektedir. Son olarak, yarışmanın kuralları gereği, verilen süre dolduğunda, her robotun bulunan afetzedenin yanında olması gerekmektedir. Bu üç farklı görev için market tabanlı bir yaklaşım ve üç farklı müzayede yapılmaktadır. Kaveh takımı keşif sırasında robotların belirlenen noktalarda belirli zamanlarda buluşup harita ve diğer bilgileri paylaşmalarını sağlamaktadır. Böylece robotlar sürekli haberleşir durumda kalmaktadır. Robotların yerel haritaları sürekli olarak küresel bir haritada birleştirilmektedir. Robotlar bir hedefi seçmeden önce o hedefe daha yakın olan bir robotun varlığını kontrol etmektedir. Her robot gideceği hedefi diğer robotlara duyurmaktadır. MRL takımının keşif yaklaşımında robotlar aynı noktadan keşfe başlamaktadırlar. Robotlar ortama gruplar halinde dağılmaktadırlar. Her bir grupta, bir robot haberleşmeyi sağlamak için hareket etmekte, diğeri ise keşif yapmaktadır. Bir hedefe ulaşmak ve aynı zamanda haberleşmenin kopmaması için oluşturulacak gruptaki robot sayısı o hedefin maliyeti olarak düşünülmektedir (VirtualRobotCompetition, 2016).

2013 yılında Eindhoven'da yapılan yarışmada PoAReT takımı 2012 yılında kullandıkları sistemin üzerine bazı eklemeler yapmıştır. Bunlardan ilki, yol planlaması yaparken birden çok başlangıç-hedef çifti için yörünge üretmek için Olasılıksal Yol Haritaları yöntemini kullanmışlardır. Bu yöntem ilk olarak bir çizge oluşturmaktadır. Bu çizge zengin bir çarpışma içermeyen yörüngeler kümesi sunmaktadır. Daha sonra istenilen başlangıç noktasından istenilen hedef noktaya en uygun yörünge bu kümenin içinden seçilmektedir. Son olarak, sınır hücrelerin belirlenmesinde sadece yeni elde edilen lazer okumaları kullanılarak bu sürecin hızlandırılması amaçlanmıştır. MRL takımı otonom keşif için RRT-connect yol planı üretme yaklaşımını kullanmıştır. Aynı zamanda, engellerden sakınmak için takip etme modu, engelden kaçınma modu, kendi etrafında dönme modu gibi seçenekler arasından uygun modu öğrenerek seçen bir yöntem seçmişlerdir. Çok robotlu keşif için 2012 yılında kullandıkları yöntemi kullanmışlardır. Yıldız takımı ise bu sene otonom seyrüsefer konusu üzerine yoğunlaşmıştır. 2012 yılında basit bir kontrolör ile seyrüsefer yapmışlardı. 2013 yılında ise tamamen otonom ve sadece robota gideceği hedef noktasının verilmesinin yeterli olacağı bir yöntem kullanmışlardır (VirtualRobotCompetition, 2016).

2014 yılında Joo Pessoa'da yapılan yarışmada eROBOTICAVR takımı da diğer takımlar gibi haberleşmeyi 3 seviyede ele almışlardır: Komuta merkezi ile direk haberleşme, diğer bir robot üzerinden haberleşme ve haberleşememe. Bununla beraber, robotların haberleşme, algılama ve hareket etme gibi davranışları ile enerji harcadıkları ve kullanılacak olan yaklaşımların enerji seviyesini göz önünde bulundurarak keşif boyunca robotun aktif kalmasını sağlamaları gerekmektedir. MRL takımı çok robotlu keşif için 3 tane strateji belirlemişlerdir. İlk olarak, fırsatçı keşifte robotlar arasında herhangi bir koordinasyon yoktur. Robotlar kendi kararlarını yerel haritalarına göre vermektedir. Bununla birlikte, komuta merkezi ile haberleşmelerinin kopmasını istememektedirler. Diğer bir yaklaşımda, robotlar komuta merkezi ile haberleşebilecekleri menzili aşarak keşif yapmakta ve belirli zaman aralıklarında bu haberleşme menziline geri dönmektedirler. Merkezi keşif yönteminde robotlar buldukları hedefleri komuta merkezine göndermekte ve komuta merkezi robotlara görevleri atamaktadır. MRL takımı aynı zamanda keşif yaparken anlamsal bilgileri de kullanmaktadır. Engellerden sakınmak için ise vektör alan histogramı yaklaşımını gerçekleştirmişlerdir. Yıldız takımı ise engellerden sakınmak için ise vektör alan histogramı yaklaşımını kullanmışlardır. Hedeflerin belirlenmesinde ise hem mesafe hem de bilgi kazancını hesaba katan bir fonksiyon kullanmışlardır (VirtualRobotCompetition, 2016).

2016 yılında Leipzig'de yapılan yarışmada Yıldız takımı birinci olmuştur. Bu seneki yarışmada geçen senelerden farklı olarak Gazebo benzetim ortamı kullanılmıştır. Yıldız takımı da bu yeni benzetim ortamına alışmak ve çok robotlu eş zamanlı konumlandırma ve haritalama problemine yoğunlaşmıştır. MRL takımı da benzer şekilde yeni kurulum için zaman harcamıştır. Bununla birlikte, çok robotlu sistemler ile keşif için iki strateji gerçekleştirmişlerdir.

Yapılan yarışmalar ile ilgili raporlar incelendiğinde afet sonrası bina içi ortamlarda robotlardan yapılması beklenen görevler şu şekilde özetlenebilir:

1. Arama kurtarma görevlerinde geniş ve bilinmeyen ortamlar kısa bir süre içerisinde robotlar tarafından keşfedilmek zorundadır.
2. Keşif görevinin verimliliği robotlar arasındaki koordinasyona ve haberleşmenin güvenilirliğine bağlıdır.
3. Robotlar ortamı doğru olarak gösteren haritalar üretmek zorundadır.
4. Robotlar düzgün olmayan zeminlerin farkında olmalı ve bu koşullar altında seyrüsefer yapabilmelidir.

5. Robotların konumlarını doğru olarak bilmesi ilk yardımcılara verecekleri bilgilerin doğruluğu açısından hayatidir.
6. Afetzedelerin konumlarının bulunması ve haritada gösterilmesi gerekmektedir.

Bu tez çalışması kapsamında ilk dört madde ele alınmıştır. Keşifte hedef seçmek ve keşif stratejisini belirlemek amacıyla robotun bulunduğu konumların anlamsal bilgisi elde edilmiş ve kullanılmıştır. Robotlar arası koordinasyonun sağlanması için market ve müzayede tabanlı gelişmiş yöntemler gerçekleştirilmiştir. Robotlar, verimli bir şekilde haberleşebilmek için belirli zamanlarda bir araya gelmektedirler ancak bu durum robotlar keşfi sürdürürken gerçekleşmektedir. Robotlar ortamı keşfederken hem metrik hem de topolojik harita üretmişlerdir. Ayrıca, ortamla ilgili anlamsal bilgiler ise topolojik haritanın düğümlerinde yer almıştır. Robotların yerel haritalarının birleştirilmesinde geçmiş çalışmalarda yer alan anlaşma/uyuşmazlık fonksiyonlarına benzer bir yaklaşım önerilmiştir. Topolojik haritaların birleştirilmesi sırasında uygulanan yaklaşım ile topolojik haritanın daha az düğümlerle ortamı daha iyi temsil etmesi sağlanmıştır. Robotlar belirledikleri hedeflere gitmek amacıyla A* ve Dijkstra algoritmalarını kullanarak yol planı yapmışlardır. Seyrüsefer sırasında robotun eğimli düzlemlere çarparak devrilmesini önlemek amacıyla renk ve derinlik algılayıcıdan (Kinect) faydalanarak yerel ara noktalar belirlenmiştir. Robotların ortamdaki engellerden sakınmaları için Vektör Alan Histogramı (Vector Field Histogram) yaklaşımı kullanılmıştır. Robotların birbirlerine çarpmalarını önlemek için ise bir protokol gerçekleştirilmiştir.

3. TEK ROBOTLU KEŞİF YÖNTEMİ

Keşif problemi, robotun bilinmeyen bir hedefe hareket ederken topladığı bilgileri saklaması olarak tanımlanmış ve çalışılmaya başlanmıştır. Bu problemin ilk resmi tanımı ise 1997 yılında Brain Yamauchi tarafından yapılmıştır. Bu tanıma göre keşif, bilinmeyen bir ortamın içine doğru hareket ederken aynı zamanda daha sonraki seyrüsefer hareketleri için kullanılacak bir harita yapmaktır. İyi bir keşif stratejisi, ortamın tamamının ya da tamamına yakın bir kısmının kabul edilebilir bir sürede haritasının çıkarılmasını sağlamalıdır (Yamauchi, 1997).

Keşif problemi üzerine yapılan çalışmalar genel olarak oluşturulan haritanın gösterimi ve sonraki hedeflerin seçimi konularında farklılıklar göstermektedirler. İlk olarak, bu farklılıkların olumlu ve olumsuz yönlerini öne çıkaran ve afet ortamında kullanılacak fikirler sunan tek robotlu keşif yöntemleri kısaca incelenecektir. Daha sonra, gerçekleştirilen tek robotlu keşif algoritması detaylı olarak anlatılacaktır.

3.1 Geçmiş Çalışmalar

Geçmiş çalışmalarda ortamlar genellikle metrik ve topolojik haritalar ile gösterilmişlerdir. Metrik gösterimlerde ortam ızgara hücreleriyle ifade edilmektedir. Bu gösterimin temel avantajları, kolay oluşturulabilmesi ve bakış açısından bağımsız olmalarıdır. Ancak bu gösterimlerde veri, hafıza ve zaman karmaşıklığı açısından verimli bir şekilde işlenememektedir. Öte yandan, topolojik gösterimlerde ortam düğümler ve bu düğümleri birbirine bağlayan ayrıtlardan oluşmaktadır. Topolojik gösterimler düşük hafıza ve zaman karmaşıklığı sağlamakta ve robotun kesin konumuna ihtiyaç duymamaktadırlar. Bununla birlikte oluşturmasının zorluğu, görüş açısına hassaslık gibi dezavantajları bulunmaktadır (Thrun, 1998).

Metrik harita gösterimini kullanan ilk keşif çalışmalarından biri Zelinsky tarafından önerilmiştir (Zelinsky, 1992). Bu çalışmada, bilinmeyen ve engeller içeren bir ortamda bir başlangıç noktasından bir hedef noktasına gitmek amaçlanmaktadır. Hedef noktası bilinmeyen kısımda olduğu için bu seyrüsefer görevi tek robotlu keşif görevi olarak kabul edilebilmektedir. Önerilen yaklaşımda yol planı mesafe dönüşümü (distance transform) yöntemi ile yapılmaktadır. Bu yöntemde, ortam dörtlü ağaç ile ifade edilmekte ve hedef hücrenin çevresindeki sekiz engel içermeyen hücreye bir (1) değeri atanmaktadır. Bir değeri içeren hücrelerin çevrelerindeki sekiz engel içermeyen hücreye iki (2) değeri atanmakta ve

bütün ortam bu şekilde gösterilmektedir. Bu gösterim sayesinde hücreler ile ilgili durumu (boş, dolu ya da bilinmiyor), durumuna güveni (confidence) gibi ek bilgiler yöntemin verimliliğini artırmak için kullanılmaktadır. Seyrüsefer görevinde başlangıç noktasından bitiş noktasına bir tepeden aşağı iner gibi hücre değerleri azaltılarak ilerlenmektedir. Ayrıca geçilen ve algılanan hücrelerin durumları ve durumlarına güvenleri de güncellenmektedir. Robot ilerlerken ya hedef noktasına ulaşmakta ya da bir engel ile karşılaşmaktadır. Hedefe ulaştığında başarılı bir seyrüsefer gerçekleştirildiği için yol üzerindeki hücrelerin durumları kesin olarak belirlenerek (keşif yapılarak) görev tamamlanmaktadır. Engelle karşılaşıldığında robotun konumu, yönelimi ve algılayıcı bilgileri ile birlikte dörtlü ağaç yapısındaki hücre güven durumları güncelleştirilmekte ve yeni bir plan yapılarak bu plan uygulanmaktadır. Eğer aşağı doğru bir yol yoksa bu hedefe ulaşılamamaktadır. Bu döngü robot hedefe ulaşana kadar ya da hedefi ulaşamaz olarak belirleyene kadar devam etmektedir.

Moorehead ve diğerleri (Moorehead vd. 2001) yaptıkları çalışmada bilinmeyen alanların keşfi gibi bir bilgi kaynağına ulaşmak yerine çoklu bilgi kaynaklarını ele alabilecek bir yaklaşım önermişlerdir. Önerilen yaklaşımda ortam, boyutları aynı olan bir doluluk ızgarası ile gösterilmekte ve bu ızgaradaki her hücre A ve G vektörlerine sahip olmaktadır. A vektörü hücre özellik vektörü olarak adlandırılmakta ve keşifin bu hücre hakkında bildiklerini içermektedir. G vektörü ise keşif tarafından göz önüne alınan her bir bilgi kaynağı için bir eleman içermekte ve 0 ile 1 değerleri ile bu algılayıcı okuması sonucunda elde edilecek bilgiyi göstermektedir. Önerilen yaklaşım dolaşılabilirlik haritaları (traversability maps) çıkarılarak test edilmiştir. A vektörünün içinde ulaşılabilirlik ve dolaşılabilirlik durumunu gösteren iki değişken bulunmaktadır. Bu değişkenler 0 ile 1 arasında değerler almakta ve 1 ulaşılabilirlik ya da dolaşılabilirliği 0 ise tam tersini göstermektedir. G vektörünün içinde komşu hücre bilgisi, dolaşılabilirliğin güvenilirliğini artırma bilgisi ve ulaşılabilirlik bilgisi yer almaktadır. Bu bilgilerin tamamı bir hücreden yapılan algılayıcı okumalarına bağlı olmakta ve bu hücreden görünen tüm hücreler bilgi kazancında kullanılmaktadır.

Topolojik harita gösterimini kullanan ilk keşif yaklaşımlarından biri ise Dudek ve diğerleri tarafında geliştirilmiştir (Dudek vd. 1991). Bu yaklaşım diğer yaklaşımların aksine uzaklık metrik sistem yerine işaret tabanlı bir yöntem kullanmakta ve robot bir düğümden başka bir düğüme ayrıt üzerinden ulaşmaktadır. Robot bir düğümden bir işareti bırakıp bir diğer işareti alabilmektedir. Ayrıca, düğümler arasındaki ayrıtların tanımlanmasında farklı bir gösterim kullanılmaktadır. Keşif algoritmasının temelini keşfedilmiş bir çizge elde etmek oluşturmaktadır. Yeni düğümler ile karşılaşıldığında bu düğümler keşfedilmiş çizgeye eklenmekte ve bu düğüme bağlı ayrıtlarda özel olarak tanımlanmaktadır. Bu ayrıtların diğer uçları mevcut keşfedilmiş çizgenin içinde değilse bu

ayrıntlar keşif için kullanılacaktır. Ayrıntların keşfedilmiş çizgenin içinde olup olmadığına karar vermek için işaretler kullanılmaktadır.

Keşif yaklaşımlarının en temel amacı robotun kısa bir sürede daha çok bilgi elde etmesini sağlamaktır. Bunu başarabilmek için sonraki hedeflerin nasıl belirleneceği konusu kritik bir rol almaktadır. Bir sonraki en iyi görüş (next-best-view) yöntemi geçmiş çalışmalarda sonraki hedefin seçiminde kullanılmaktadır. Banos ve Latombe (Gonzalez-Banos ve Latombe, 2002) yaptıkları çalışmada algılamanın çok iyi olmadığı durumlarda güvenli ve verimli bir haritalama yöntemini tek robotlu sistemler için önermişlerdir. Yazarlar, bu haritalama probleminin anlık konumlandırma ve haritalama (SLAM) problemine çok benzediğini ancak SLAM probleminde robotun bir sonraki adımda nereye gideceğinin ele alınmadığını belirtmektedirler. Bu noktada, önerilen yaklaşım robotun bir sonraki adımda gideceği iyi konumları belirlemektedir. Bu konumlarda, robotun elde edeceği bilgi hem miktar hem de kalite olarak daha çok olmaktadır. Bu yaklaşım bilgisayarda görü alanındaki bir sonraki en iyi görüş (next-best-view) yöntemine benzemektedir. Ancak, robotik alanında robotların kısıtlı algılama ve ortam bilgisini kullanarak güvenli bir şekilde seyrüsefer yapmaları gerekmektedir. Ayrıca mevcut harita ile gidilecek iyi konumdaki görüş büyük oranda üst üste gelmeli ve pozisyon hatasını da azaltmalıdır. Bunları başarmak için güvenli bölge kavramı ortaya atılmıştır. Algılayıcı kısıtlamalarını göz önüne alarak en uzak algılayıcı okumasının yapılabileceği en büyük engel içermeyen bölge güvenli olarak tanımlanmaktadır. Güvenli bölge kavramını kullanan bir sonraki en iyi görüş algoritması bu çalışmada önerilmiş ve anlık konumlandırma ve haritalama sistemiyle birlikte kullanılmıştır.

Keşif probleminin çözümü için önerilen yaklaşımların çoğunda bir sonraki hedefin belirlenmesinde sınır hücre (frontier) yöntemi kullanılmaktadır (Yamauchi, 1997). Sınır hücre tabanlı keşif yaklaşımındaki temel düşünce ortamla ilgili en çok bilgiyi elde edebilmek için açık ve haritası yapılmamış alanların sınırına hareket etmektir. Bu sınırların birleşimi ile oluşan bölgeye sınır hücre alanları (frontier regions) adı verilmektedir. Bir robot bir sınır hücre alanına doğru hareket ettiğinde kendini keşfedilmemiş bir alanın içinde bulacaktır ve bu yeni bilgileri haritaya ekleyecektir. Sonuç olarak haritalanmış bölge büyüyecek, bilinen ve bilinmeyen alanlar arasındaki sınır azalacaktır. Birbirini takip eden sınır hücre alanlarına hareket eden robot sürekli ortam hakkında bilgisini genişletecektir. Tüm ortam keşfedilene kadar sürekli bir sınır hücre alanı mevcut olduğu için bu yaklaşım tam (complete) bir yaklaşımdır ve sınır hücre tabanlı keşif yaklaşımı olarak adlandırılmaktadır.

Sınır hücre tabanlı keşif yaklaşımı ortamın gösterimi için kanıt ızgaraları (evidence grids) kullanılmaktadır (Moravec ve Elfes, 1985). Bu ızgaralar hücrelerden oluşan kartezyen

bir yapıdadırlar. Her hücre doluluk durumuna göre bir olasılık değeri tutmaktadır. İlk başta, bu olasılık değerlerine 0.5 atanmaktadır ve bunlara ilk olasılık değeri adı verilmektedir. Robotun algılayıcılarından ortamla ilgili yeni bilgiler geldikçe bu olasılık değerleri de güncellenmektedir. Bu kanıt ızgarası oluşturulduktan sonra her hücre için ilk olasılık ve doluluk olasılık değerleri karşılaştırılarak bir sınıflandırma yapılmaktadır. Her hücre aşağıda verilen sınıflardan birine aittir:

- **Boş Hücre:** Doluluk olasılığı $<$ İlk olasılık
- **Bilinmeyen Hücre:** Doluluk olasılığı $=$ İlk olasılık
- **Dolu Hücre:** Doluluk olasılığı $>$ İlk olasılık

Boş bir hücreye komşu olan herhangi bir bilinmeyen hücre, sınır hücresi (frontier edge cell) olarak adlandırılmaktadır. Bütün sınır hücreleri bir araya getirilerek sınır hücre alanları (frontier regions) oluşturulmaktadır. Belirlenen bir eşik değerin (genellikle robotun boyutları) üzerindeki her sınır hücre alanı sınır hücre (frontier) olarak kabul edilmektedir. Sınır hücreler tanımlandıktan sonra robot en yakınındaki sınıra doğru hareket etmektedir. Hareket planı derinlik öncelikli arama (depth-first search) algoritması ile yapılmaktadır ve engelsiz bir yol üretilmektedir. Robot sınıra ulaştığında bu sınır hücreyi ziyaret edilmiş olarak kayıt altına almakta ortam ile ilgili bilgileri algılayıcıları ile edindikten sonra kanıt ızgarasını ve haritasını güncellemekte ve en yakındaki sınır hücreye doğru hareket planını hazırlamaya başlamaktadır.

Bir sonraki alt bölümde yukarıda ana fikri verilen sınır hücre tabanlı keşif yaklaşımı detaylı olarak anlatılacaktır. Bu yöntem olası hedeflerin belirlenmesinde kullanılmaktadır. Bu noktada, olası hedefler arasından hangisinin seçileceği, bu hedefe nasıl gidileceği ve ortam haritasının nasıl oluşturulup güncelleneceği gibi sorular ortaya çıkmaktadır. Bu problemlere çözüm bulmak amacıyla dört erkenden oluşan bir sistem tasarlanmış ve gerçekleştirilmiştir. Bu sistemin mimarisi ve detayları da bir sonraki alt bölümde incelenecektir.

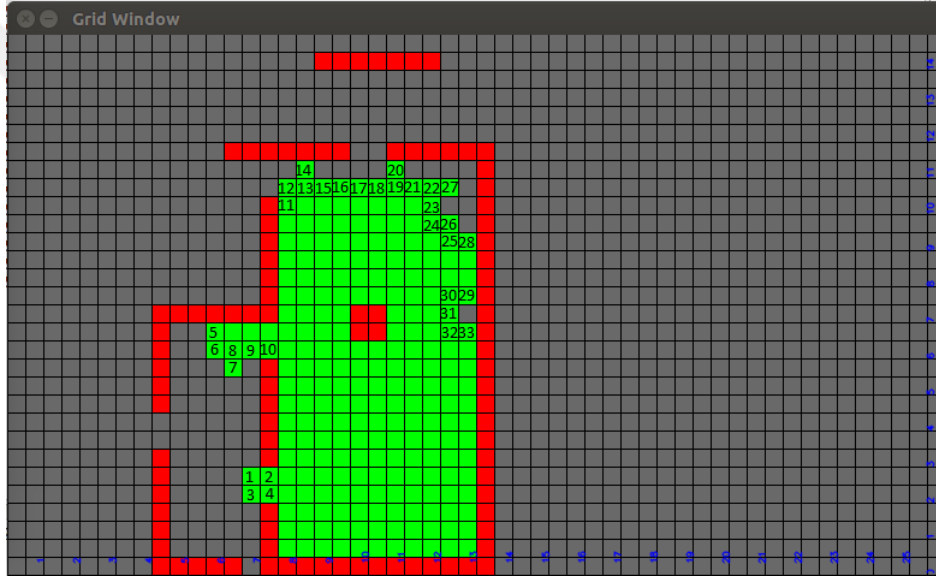
3.2 Gerçeklenen Yaklaşım

Gerçeklenen Yaklaşım alt bölümünde ilk olarak sınır hücre tabanlı keşif yaklaşımı incelenecektir. Daha sonra, sistem mimarisi verilerek bu mimari de yer alan erkinlerin üstlendikleri görevler detaylı olarak açıklanacaktır.

3.2.1 Sınır hücre tabanlı keşif yaklaşımı

Yamauchi'nin 1997 yılındaki (Yamauchi, 1997) yayınında önerdiği keşif algoritması iki fazdan oluşmaktadır. İlk fazda sınır hücreleri (frontier) aşağıdaki algorithmanda verildiği gibi belirlenmektedir.

Algoritmanın ilk adımında bilinmeyen hücreye komşu olan boş hücreler belirlenmektedir. Bu hücreler sınır kenar hücreleri (frontier edge cell) olarak adlandırılmaktadır. Sınır kenar hücrelerini tespit etmek için boş hücrelerin 8 komşuluğundaki bilinmeyen hücreler aranmaktadır. Örnek bir durum Şekil 3.1'de verilmiştir. Bu örnek için 33 adet sınır kenar hücresi belirlenmiştir.



Şekil 3.1: Sınır Kenar Hücrelerin Belirlenmesi

İkinci adımda ise birbirine komşu olan sınır kenar hücreleri sıg öncelikli arama kullanılarak gruplanmakta ve sınır hücre alanlarını oluşturmaktadır. Eğer bir hücre alanı belirlenen bir eşik değerin (genellikle robotun boyutları) üzerinde sınır kenar hücre sayısına sahipse bu hücre alanının merkezi sınır hücresi olarak kabul edilmektedir. Şekil 3.2'de oluşturulan 4 sınır hücre alanı 4 farklı renk ile gösterilmiştir. Sınır hücre alanlarının ortasındaki siyah daireler ise belirlenen sınır hücrelerini temsil etmektedirler.

İkinci fazda sınır hücreleri belirlendikten sonra robota en yakın, ulaşılabilir ve ziyaret edilmemiş sınır hücresi en iyi sınır hücresi olarak seçilmekte ve robot o hücreye doğru hareket etmektedir. Robot hedefine ulaştığında bu sınır hücresini ziyaret edilmiş olarak kayıt altına almakta, ortam ile ilgili bilgileri algılayıcıları ile edindikten sonra işaret

Algoritma 1: İlk Faz: Sınır Hücrelerini Belirleme

Girdi: Ortamın durumunu gösteren dizi *array*

Çıktı: Sınır hücrelerini tutan liste *frontier*

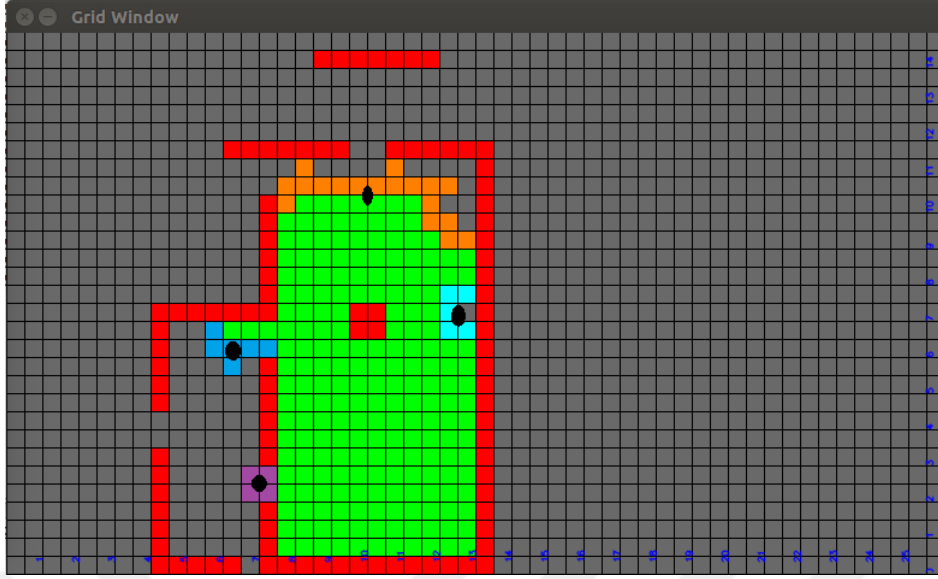
Veri: sınır kenar hücresi listesi *frontierEdgeCell*, sınır hücre alanları listesi *frontierRegion*,
sınır hücresi listesi *frontier*

```

frontierEdgeCell ← bos;
frontierRegion ← bos;
frontier ← bos;
// İlk Adım: Sınır kenar hücreleri bulunarak listeye atılıyor
foreach array hücresi do
  if hücresinin değeri == 1 then
    while count < 8 ve hücre listeye atılmadıysa do
      Hücrenin 8 komşusundan birine bak;
      if komşu hücresinin değeri == 0.5 then
        | frontierEdgeCell ← hücre;
      else
        | count++;
      end
    end
  else
    | hücreyi atla;
  end
end
// İkinci Adım: Sınır hücre alanları oluşturuluyor
foreach frontierEdgeCell elemanı do
  Sığ öncelikli arama (Breadth first search) kullanarak birbirine komşu olan sınır kenar
  hücrelerini bul;
  // BFS tamamlandığında sınır hücre alanlarından biri oluşturuluyor
  frontierRegion ← frontierEdgeCell elemanı;
end
// Son Adım: Sınır hücreleri belirleniyor
foreach frontierRegion elemanı do
  if sınır hücre alanındaki hücre sayısı > threshold then
    | sınır hücre alanının merkezini bul;
    | frontier ← sınır bölgesinin merkezi;
  else
    | sınır hücre alanını geç;
  end
end

```

ızgarasını ve haritasını güncellemekte ve bir sonraki sınır hücrelerini belirlemek için yukarıda anlatılan prosedürü tekrar işletmektedir. Yaklaşımın ikinci fazı Algoritma 2'de verilmiştir.



Şekil 3.2: Sınır Hücrelerin Belirlenmesi

Algoritma 2: En İyi Sınır Hücresinin Belirlenmesi

Girdi: Sınır hücresi listesi *frontier*

Çıktı: En iyi sınır hücresi *bestFrontier*

Veri: En yakın sınır hücresi *nearestFrontier*, en iyi sınır hücresi *bestFrontier*

// En yakın sınır hücresini bul

foreach *frontier* hücresi **do**

 | hücre = en yakın sınır hücresini bul;

 | *nearestFrontier* \leftarrow hücre;

end

// En yakın sınır hücresi ulaşılabilir mi

if *nearestFrontier* ulaşılabilir **ise then**

 | // En yakın sınır hücresi ziyaret edilmiş mi

 | **if** *nearestFrontier* daha önce ziyaret edilmemiş **ise then**

 | *bestFrontier* \leftarrow *nearestFrontier*;

 | **else**

 | *nearestFrontier* *frontier* listesinden sil;

 | başa dön;

 | **end**

else

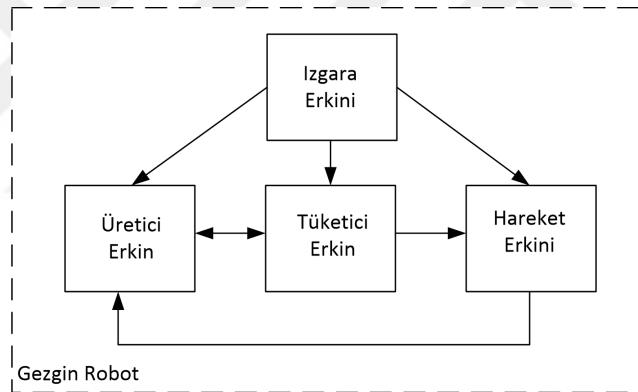
 | *nearestFrontier* *frontier* listesinden sil;

 | başa dön;

end

3.2.2 Sistem mimarisi

Bu tez çalışmasında, sınır hücre keşif algoritmasını gerçekleştirmek için dört erkinden oluşan bir sistem mimarisi tasarlanmıştır. Bu erkinlerden ızgara erkini ortamın gösterimi için kullanılan doluluk ızgarasını oluşturmak, güncellemek, istenildiğinde diğer erkinlerle paylaşmak ve mevcut ızgarayı görsel arayüz yardımı ile ekrana getirmekten sorumlu olmaktadır. Bir diğer erkin olan üretici erkin belirli aralıklarla yapılması gereken işleri belirlemektedir. Bu süreçte sınır hücre keşif algoritmasının ilk fazı gerçekleşmekte ve aday sınır hücrelerine karar verilmektedir. İkinci fazda ise çeşitli kriterler kullanılarak bu görevlerin yapılması durumunda elde edilecek fayda belirlenmekte ve en faydalı görev seçilmektedir. Tüketici erkin ise belirlenen sınır hücrelerine gitmenin maliyetini hesaplamaktadır. Hareket erkini ise, gidilecek sınır hücresi belirlendikten sonra bu hücreye gitmek için yol planı yapmaktan ve güvenli bir şekilde bu hücreye gitmekten sorumludur. Şekil 3.3'te sistem mimarisi yer almaktadır. Bundan sonraki alt bölümlerde bahsedilen dört erkin detaylı olarak anlatılacaktır.



Şekil 3.3: Sistem Mimarisi

3.2.2.1 Izgara erkini

Bu tez çalışmasında, ortamın metrik haritası doluluk ızgarası ile ifade edilmektedir. Izgara erkini temel görevi doluluk ızgarasını oluşturmak, güncellemek ve diğer erkinler tarafından istendiği takdirde güncel ızgarayı onlara göndermektir. Bununla birlikte, gezgin robot bir görevi gerçekleştirirken robotun izlediği yolun, yaptığı ölçümlerin görsel bir arayüz aracılığı ile ekrana getirilmesi görevin doğruluğunun analiz edilmesinde yardımcı olmaktadır. Bu görsel arayüzü oluşturma görevini de ızgara erkini üstlenmektedir.

Izgara hücrelerinin üç durumda olabileceği kabul edilmektedir: Bilinmeyen, boş ve dolu. Izgardadaki her hücre ilk başta bilinmeyen olarak işaretlenmekte daha sonra robotun algılayıcıları ile hücrelerin durumları güncellenmektedir. Hücrelerin durumunun

belirlenmesinde lazer ölçümleri kullanılmaktadır. Lazer ölçümleri, lazer algılayıcısının robot üzerindeki küresel konumundan (p_r) lazer ışınının uç konumuna (p_l) kadar olan (menziline ulaştığında ya da bir engelle karşılaştığında sonlanan) doğru parçaları olarak tanımlanabilmektedir. Lazer ışınının mesafe (d) ve açı (ϕ) bilgisi kullanılarak lazer ışınının uç noktasının yerel koordinatları hesaplanmaktadır. Elde edilen koordinatlar robotun yönelim açısına (θ) göre döndürülerek ve p_r kadar kaydırılarak küresel koordinatlara çevrilmiştir. Bu işlemleri ifade eden genel Denklem 3.1'de gösterilmiştir.

$$p_l = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d \cos \phi \\ d \sin \phi \\ 1 \end{bmatrix} + p_r \quad (3.1)$$

Denklem 3.1'de verilen hesaplama robotun yuvarlanma ve yunuslama açılarının sıfır olduğu ve lazer ölçümlerinin xy düzlemine paralel olduğu durumlarda geçerli olmaktadır. Bununla birlikte, robotun düzgün olmayan zemin koşullarında (rampa, moloz) yuvarlanma ve yunuslama açıları sıfırdan farklı olacaktır ve bu açıların değerlerinin de hesaba katılması gerekmektedir. Bunun için Euler açıları sapma (θ), yunuslama (β) ve yuvarlanma (γ) kartezyen koordinat sistemine Denklem 3.2'de gösterildiği üzere dönüştürülmüştür (Craig, 2005). Bu denklemde $c\theta$, $\cos \theta$ 'nin ve benzer şekilde $s\theta$, $\sin \theta$ 'nin kısaltılmışı olarak kullanılmaktadır. Denklem 3.1, Denklem 3.3 ve 3.4 yunuslama ve yuvarlanma açılarının sıfırdan farklı olduğu durumlara göre genelleştirilmiştir. Bununla birlikte, bu denklemler, sadece bahsedilen açıların belirlenen bir eşik değerinden küçük olduğunda kullanılmaktadır.

$$R_{XYZ}(\gamma, \beta, \theta) = \begin{bmatrix} c\theta c\beta & c\theta s\beta s\gamma - s\theta c\gamma & c\theta s\beta c\gamma + s\theta s\gamma \\ s\theta c\beta & s\theta s\beta s\gamma + c\theta c\gamma & s\theta s\beta c\gamma - c\theta s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (3.2)$$

$$p_l = \begin{bmatrix} d \cos \phi \\ d \sin \phi \\ 1 \end{bmatrix} R_{XYZ}(\gamma, \beta, \theta) + p_r \quad (3.3)$$

$$p_l = d \begin{bmatrix} c\theta c\beta c\phi + c\theta s\beta s\gamma s\phi - s\theta c\gamma s\phi \\ s\theta c\beta c\phi + s\theta s\beta s\gamma s\phi + c\theta c\gamma s\phi \\ -s\beta c\phi + c\beta s\gamma s\phi \end{bmatrix} + p_r \quad (3.4)$$

Bunlara ek olarak, bahsedilen açılar nedeniyle xy düzlemine paralelliği bozulan ışınların, bu düzleme izdüşümlerinin alınması gerekmektedir. Bunun için, yerçekimi vektörünün yönü aşağı doğru seçilmiş ve Denklem 3.5'te gösterilmiştir.

$$g = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (3.5)$$

Yerçekimi vektörü kullanılarak izdüşüm matrisi s aşağıdaki gibi formüle edilebilir:

$$s = \begin{bmatrix} 1 - g_x g_x & -g_x g_y & -g_x g_z \\ -g_y g_x & 1 - g_y g_y & -g_y g_z \\ -g_z g_x & -g_z g_y & 1 - g_z g_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

Lazer ışınının xy düzlemi üzerindeki izdüşümü $proj_{xy}(p_l)$ olarak tanımlanmakta ve Denklem 3.7'de verilmektedir.

$$proj_{xy}(p_l) = s p_l \quad (3.7)$$

Her lazer taramasında daha önceden belirlenen açılardaki lazer ışınlarına Denklem 3.7'de verilen işlem uygulanmaktadır. Eğer bir hücreden, en az önceden belirlenmiş bir sayı kadar farklı lazer ışını geçer ise, o hücre boş olarak kabul edilmektedir. Öte yandan, eğer bir ışının mesafesi, lazer ışınının gidebileceği en uzak mesafeden az ise, hücre dolu olarak tanımlanmaktadır. Dolayısıyla ızgaradaki hücrelerin durumlarını güncellemek için lazer ölçümlerinin hangi hücrelerden geçtiğine karar vermek yeterli olmaktadır. Sonuç olarak, hücrelerin durumlarının güncellenmesinde ızgara-doğru kesişim (grid-line intersection) algoritmalarından birinin kullanılması uygun olacaktır.

Bresenham algoritması (Bresenham, 1965) ızgara-doğru kesişim algoritmalarının en çok bilinenleri arasında yer almaktadır. Bu algoritma ilk olarak eğimin sıfırdan küçük olduğu ve mutlak değerinin birden küçük olduğu durumlar için ortaya atılmıştır. Ayrıca bu algoritma girdilerinin tamsayı olduğu varsayımına dayanmaktadır. Algoritmada, her adımda x eksenini boyunca adım boyu kadar ilerlemekte daha sonra doğrunun x eksenini kestiği yere göre bir sonraki aday noktayı belirlemektedir. Algoritmanın detaylı anlatımına (Bresenham, 2016) adresinden ulaşılabilmektedir. Bresenham tabanlı bir başka algoritma olan Dijital

Diferansiyel Analizer (DDA) ise hem tamsayı hem de gerçel sayılar için kullanılabilir. Ayrıca bu algoritmada eğimin sıfırdan büyük olduğu varsayımı yapılmaktadır (DDA, 2016). Doluluk ızgarasının oluşturulmasında, Bresenham algoritması (Bresenham, 1965) temelli bir algoritma olan DDA algoritması genişletilerek kullanılmaktadır. Gerçeklenen yaklaşımda eğimin sıfırdan büyük olduğu durumlar da hesaba katılmıştır. Görsel arayüzün gerçekleşmesi noktasında, oluşan doluluk ızgarası, robotun konumu ve lazer ölçümleri openCV yardımıyla ekrana getirilmiştir.

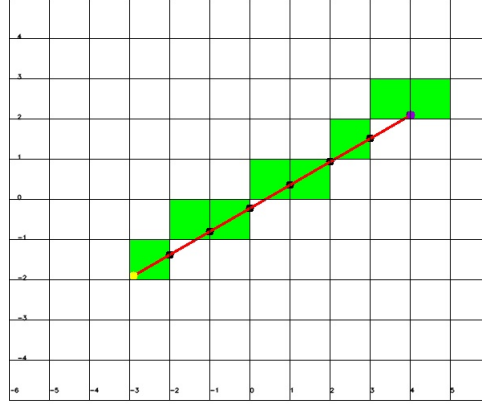
DDA algoritması

Bu algoritmada ilk olarak doğrunun eğimi hesaplanmaktadır. Eğer eğim birden küçük veya eşitse her adımda x eksenini boyunca adım boyu kadar ilerlemekte ve x_{k+1} bulunmaktadır. Daha sonra doğrunun x_{k+1} eksenini kestiği yer hesaplanmakta ve bu yer y_{k+1} olarak adlandırılmaktadır. Denklem 3.8'de x_{k+1} ve y_{k+1} değerlerini bulmak için kullanılan denklemler verilmiştir.

$$\begin{aligned}
 y_{k+1} &= y_k + m \\
 y_k &= y_{k+1} \\
 x_{k+1} &= x_k + stepSize \\
 x_k &= x_{k+1}
 \end{aligned}
 \tag{3.8}$$

Doğru parçasının sonlandığı noktanın x koordinatı x_{end} olarak tanımlanırsa Denklem 3.8'de verilen işlemler $x_k < x_{end}$ karşılaştırması doğru olduğu sürece tekrarlamalı olarak devam etmektedir. Bu süreç sonunda eğimin birden küçük olduğu durumda DDA algoritmasının sonucu Şekil 3.4'te verilmiştir. Şekilde sarı ve mor ile gösterilen daireler sırasıyla doğrunun başlangıç ve bitiş noktalarıdır. Siyah daireler her iterasyonda elde edilen x_{k+1} ve y_{k+1} koordinatlarının oluşturduğu noktalardır. İçi yeşil ile doldurulmuş hücreler ise boş hücre olarak tanımlanmaktadır. Şekilde görüldüğü gibi doğrunun içinden geçtiği ancak boş olarak işaretlenmemiş hücreler mevcuttur.

Eğimin birden büyük olduğu durumlarda ise y eksenini boyunca ilerlenmekte ve y_{k+1} bulunmaktadır. Daha sonra doğrunun y_{k+1} eksenini kestiği yer hesaplanmakta ve bu yer

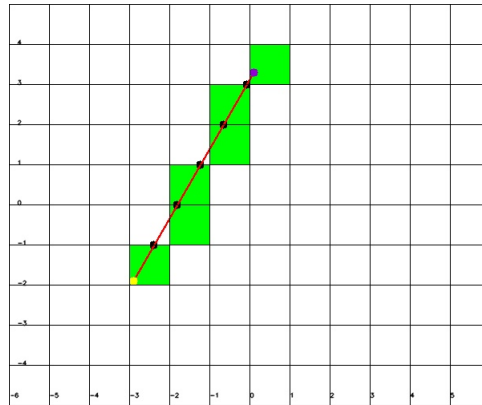


Şekil 3.4: Eğimin birden küçük olduğu durumda DDA algoritmasının sonucu

x_{k+1} olarak adlandırılmaktadır. Denklem 3.9'da x_{k+1} ve y_{k+1} değerlerinin bulunmasında kullanılan denklemler verilmiştir.

$$\begin{aligned}
 x_{k+1} &= x_k + \frac{1}{m} \\
 x_k &= x_{k+1} \\
 y_{k+1} &= y_k + stepSize \\
 y_k &= y_{k+1}
 \end{aligned}
 \tag{3.9}$$

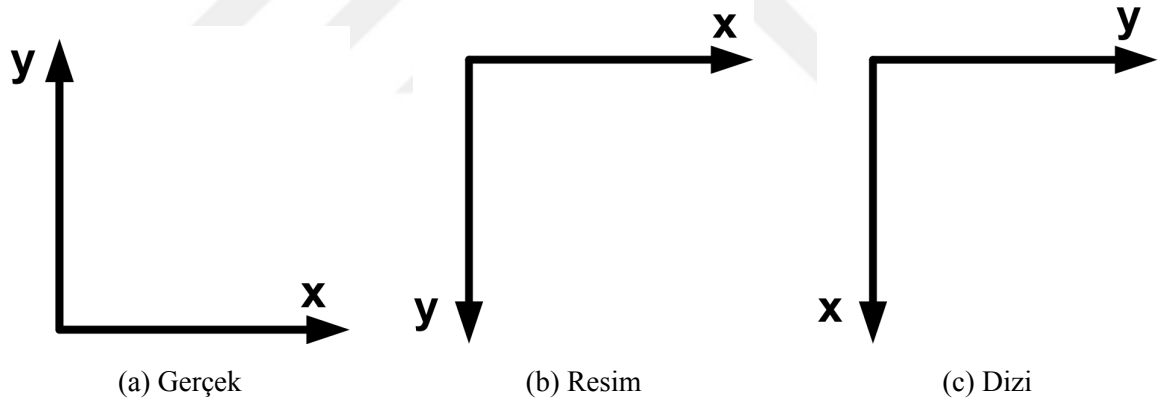
Doğru parçasının sonlandığı noktanın y koordinatı y_{end} olarak tanımlanırsa Denklem 3.9'da verilen işlemler $y_k < y_{end}$ karşılaştırması doğru olduğu sürece tekrarlamalı olarak devam etmektedir. Bu süreç sonunda eğimin birden büyük olduğu durumda DDA algoritmasının sonucu Şekil 3.5'te verilmiştir. Şekilde görüldüğü gibi doğrunun içinden geçtiği ancak boş olarak işaretlenmemiş hücreler mevcuttur.



Şekil 3.5: Eğimin birden büyük olduğu durumda DDA algoritmasının sonucu

Genişletilmiş DDA Algoritması

Gerçeklenen yaklaşımda yapılmak istenen işlerden biri robottan ya da benzetim ortamından alınan gerçek koordinatları kullanarak robotun konumunu ve algılayıcı ölçümlerinin etkilerini ortamın ızgara olarak tanımlandığı bir resim üzerinde göstermektir. Aynı zamanda ızgara ile tanımlanan ortamdaki her hücrenin durumunu bir dizide tutarak bu diziyi keşif algoritmasının girdisi olarak kullanmak amaçlanmaktadır. Dolayısıyla, gerçek koordinatların resim üzerinde nasıl gösterileceğine, dizide hangi elemana karşılık geleceğine karar vermek gerekmektedir. Bunu yapabilmek için gerçek, resim ve dizi koordinat sistemlerinin bilinmesi önemlidir. Şekil 3.6'da bu üç farklı koordinat sistemi verilmiştir. Şekilde görüldüğü gibi gerçek koordinat sisteminde x ve y eksenleri bilindik kartezyen koordinat sisteminin aynıdır. Ayrıca bu koordinat sistemi sürekli sayılar içermektedir. Resim koordinat sistemi ise sol üst köşeyi başlangıç noktası olarak kabul etmekte, x eksenini sağa, y eksenini aşağı doğru artmaktadır. Bununla birlikte, dizi koordinat sistemi resim koordinat sisteminde x eksenini ile y ekseninin yer değiştirmiş halidir. Ayrıca bu iki koordinat sistemi ayrık sayılar içermektedir.



Şekil 3.6: Koordinat Sistemleri

Bu noktada, koordinat sistemleri arasındaki geçiş matematiksel olarak gösterilecektir. Öncelikle gerçek koordinattan dizi koordinatına dönüşümü anlatmak faydalı olacaktır. Gerçek koordinatların x ekseninde $-a$ ve a , y ekseninde $-b$ ve b değerleri arasında var olduğunu varsayalım. Bu durumda dizi koordinatları x ekseninde 0 ve $2a$, y ekseninde 0 ve $2b$ değerleri arasında olacaktır. Bununla birlikte, dizi koordinat sistemi (x_D ve y_D) gerçek koordinat sisteminin (x_G ve y_G) 90 derece saat yönünde çevrilmiş halidir. Bu döndürme işleminden sonra, x ekseninde a ve y ekseninde b değerleri kadar ötelemek yeterli olacaktır. Son olarak daha önce belirtildiği gibi gerçek koordinat sistemi sürekli ve dizi koordinat sistemi ayrık olduğu için bu dönüşümü yapmadan önce gerçek koordinatların

tamsayıya yuvarlanması gerekmektedir. Gerçek koordinatlardan dizi koordinatlarına dönüşüm Denklem 3.10'da matematiksel olarak verilmiştir.

$$\begin{bmatrix} x_D \\ y_D \end{bmatrix} = \text{round} \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right) \begin{bmatrix} x_G \\ y_G \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} a + \begin{bmatrix} 0 \\ 1 \end{bmatrix} b \quad (3.10)$$

Dizi koordinatları elde edildikten sonra bu koordinatlar kullanılarak resim koordinatları elde edilebilir. Resim koordinat sisteminde x_k ile x_{k+1} ya da y_k ile y_{k+1} arasında $hBoyutu$ olarak adlandırılan sayıda piksel yer almaktadır. Bununla birlikte resim koordinat sistemindeki x_R dizi koordinat sistemindeki y_D ile aynı doğrultuda ve aynı yerden başlamaktadır. Benzer şekilde, resim koordinat sistemindeki y_R dizi koordinat sistemindeki x_D ile aynı doğrultuda ve aynı yerden başlamaktadır. Dolayısıyla dizi koordinat sisteminden resim koordinat sistemini elde etmek için eksenleri yer değiştirip bu değerleri $hBoyutu$ ile çarpmak yeterli olacaktır. Dizi koordinatlarından resim koordinatlarına dönüşüm Denklem 3.11'de matematiksel olarak verilmiştir.

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_D \\ y_D \end{bmatrix} hBoyutu \quad (3.11)$$

Gerçeklenen algoritmayı anlatmaya başlamadan önce son olarak bazı tanımlamaların yeniden yapılması gerekmektedir. Denklem 3.12'de bu tanımlamalar yer almaktadır.

$$\begin{aligned} m &= \frac{x_2 - x_1}{y_2 - y_1} \\ x &= my + b \\ b &= x - my \\ y &= \frac{x - b}{m} \end{aligned} \quad (3.12)$$

Bu noktadan sonra ızgara erkinini gerçeklemek için kullanılan algoritmalar kısaca anlatılacaktır. Hücre Bul algoritması gerçek koordinatları verilen bir noktanın bulunduğu hücrenin sağ üst köşesini bulmak için kullanılmaktadır. Bu köşe hücrenin dizi koordinatlarını göstermektedir.

Algoritma 3: Hücre Bul

Girdi: Robotun gerçek koordinatları x_c ve y_c
Çıktı: Hücrenin dizi koordinatları x_{D_k} ve y_{D_k}
Veri: Ortamın boyutları x eksen için $2a$ ve y eksen için $2b$

$$x_k \leftarrow \text{ceil}(x_c);$$

$$b \leftarrow x_c - m * y_c;$$

$$y_k \leftarrow \frac{x_k - b}{m};$$

$$x_{D_k} \leftarrow y_k + a;$$

$$y_{D_k} \leftarrow -x_k + b;$$

Hücre Güncelle algoritması ise dizi koordinatları verilen hücrenin *array* dizisindeki durumunu güncellemektedir. Her hücre için bir indeks tutulmaktadır. Bu indekste bir tarama sırasında bu hücreden kaç ışının geçtiğini tutmaktadır. Bu değer belirlenen bir eşik değerinden büyükse hücreye 1 atamaktadır. Eğer ölçümlerden biri menzilden daha az bir değerdeyse bu hücreye 0 atanmaktadır. Aksi takdirde hücre 0.5 değerini korumaktadır. Bu değerler kullanılarak görsel arayüz güncellenmektedir. Hücrenin 0 olduğu durumda kırmızı, 1 olduğu durumda yeşil ve 0.5 olduğu durumda ise gri bu hücrenin durumunu göstermek için kullanılmaktadır.

Algoritma 4: Hücre Güncelle

Girdi: Hücrenin dizi koordinatları x_{D_k} ve y_{D_k}
Çıktı: Hücre bilgilerini tutan dizi *array* ve güncellenmiş resim
Veri: Hücre Boyutu her iki eksen için *hBoyutu*

$$\text{array}[x_{D_k}][y_{D_k}].\text{index} \leftarrow \text{array}[x_{D_k}][y_{D_k}].\text{index} + 1;$$

if $\text{array}[x_{D_k}][y_{D_k}].\text{index} > \text{threshold}$ **then**
| $\text{array}[x_{D_k}][y_{D_k}].\text{state} \leftarrow 1$

else if $\text{array}[x_{D_k}][y_{D_k}].\text{value} < \text{range}$ **then**
| $\text{array}[x_{D_k}][y_{D_k}].\text{state} \leftarrow 0$

else
| $\text{array}[x_{D_k}][y_{D_k}].\text{state} \leftarrow 0.5$

$$\text{upperRight} \leftarrow (x_{D_k} * h\text{Boyutu}, y_{D_k} * h\text{Boyutu});$$

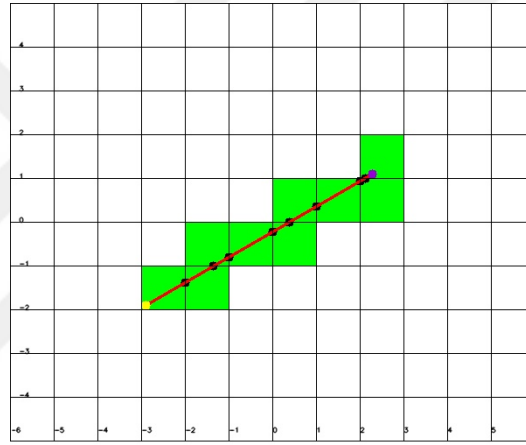
$$\text{lowerLeft} \leftarrow ((x_{D_k} - 1) * h\text{Boyutu}, (y_{D_k} + 1) * h\text{Boyutu});$$

if $\text{array}[x_{D_k}][y_{D_k}].\text{state} == 0$ **then**
| $\text{fill}(\text{upperRight}, \text{lowerLeft}, \text{red})$

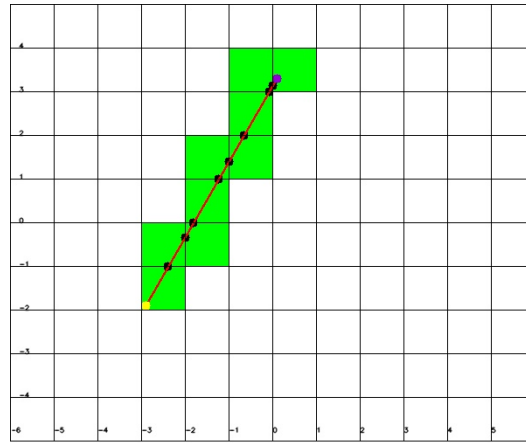
else if $\text{array}[x_{D_k}][y_{D_k}].\text{state} == 1$ **then**
| $\text{fill}(\text{upperRight}, \text{lowerLeft}, \text{green})$

else if $\text{array}[x_{D_k}][y_{D_k}].\text{state} == 0.5$ **then**
| $\text{fill}(\text{upperRight}, \text{lowerLeft}, \text{gray})$

Izgara erkini algoritması verilen ortam boyutlarına uygun olarak bir resim oluşturmakta; bu resmin üzerine $hBoyutu$ ile belirlenen aralıklarda çizgiler çizerek ızgara tabanlı ortamın gösterimini elde etmektedir. Daha sonra lazer ölçümlerinden ışının son noktasını yukarıda anlatılan yunuslama ve yuvarlanma açılarını da hesaba katarak bulmaktadır. Daha sonra bulduğu uç noktayı xy düzlemine yansıtmaktadır. Gerçek koordinatlar dizi koordinatlarına çevrildikten sonra eğim hesaplanmakta ve ölçümün başlangıç bitiş noktaları ile eğim göz önünde bulundurularak her iki eksen içinde hareket edilecek doğrultular belirlenmektedir. Algoritma 5'te ızgara erkini ile ilgili detaylar verilmiştir. Izgara erkini algoritmasında DDA'nın tek eksende yaptığı tarama her iki eksen de yapılmıştır. Böylece boş olarak işaretlenmesi gereken ancak DDA ile işaretlenmeyen hücreler de doğru bir şekilde güncellenmiştir (Şekil 3.7).



(a) Eğim birden küçük



(b) Eğim birden büyük

Şekil 3.7: Izgara erkini algoritmasının sonuçları

Algoritma 5: Izgara Erkini

Girdi: Robotun gerçek koordinatları (x_c, y_c, th_c) ve lazer ölçümleri $d(i), i = 0, \dots, 180$

Çıktı: Ortamı ızgara halinde gösteren resim ve hücre bilgilerini tutan dizi *array*

Resmi yarat, Izgara çizgilerini çiz ve numaralandır;

forall the d ölçümleri do

$x_{end} \leftarrow x_c + d(i) * \cos(th_c + i - 90), y_{end} \leftarrow y_c + d(i) * \sin(th_c + i - 90);$

Hücre Bul(x_{end}, y_{end}), $m \leftarrow x_{end} - x_c / y_{end} - y_c;$

if $m > 0$ then

if $x_c < x_{end}$ then

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $x_{D_k} > x_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Yukarı doğru x eksenini taranıyor

$x_{D_k} \leftarrow x_{D_k} - 1, y_{D_k}$ hesapla;

end

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $y_{D_k} < y_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Sağa doğru y eksenini taranıyor

$y_{D_k} \leftarrow y_{D_k} + 1, x_{D_k}$ hesapla;

end

else

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $x_{D_k} < x_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Aşağı doğru x eksenini taranıyor

$x_{D_k} \leftarrow x_{D_k} + 1, y_{D_k}$ hesapla;

end

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $y_{D_k} > y_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Sola doğru y eksenini taranıyor

$y_{D_k} \leftarrow y_{D_k} - 1, x_{D_k}$ hesapla;

end

end

else

if $x_c < x_{end}$ then

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $x_{D_k} > x_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Yukarı doğru x eksenini taranıyor

$x_{D_k} \leftarrow x_{D_k} - 1, y_{D_k}$ hesapla;

end

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $y_{D_k} > y_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Sola doğru y eksenini taranıyor

$y_{D_k} \leftarrow y_{D_k} - 1, x_{D_k}$ hesapla;

end

else

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $x_{D_k} < x_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Aşağı doğru x eksenini taranıyor

$x_{D_k} \leftarrow x_{D_k} + 1, y_{D_k}$ hesapla;

end

$(x_{D_k}, y_{D_k}) \leftarrow$ Hücre Bul(x_c, y_c);

while $y_{D_k} < y_{D_end}$ do

Hücre Güncelle(x_{D_k}, y_{D_k}); // Sağa doğru y eksenini taranıyor

$y_{D_k} \leftarrow y_{D_k} + 1, x_{D_k}$ hesapla;

end

end

end

Lazer ölçümlerini göster;

end

Resmi göster, array ekrana getir;

3.2.2.2 Üretici erkin

Üretici erkinin temel görevi belirli aralıklarla yapılması gereken işleri belirlemektir. Bunu dört aşamada gerçekleştirmektedir. İlk aşama görev sunma olarak adlandırılabilir. Bu aşamada, ızgara erkininden gelen diziyi kullanarak sınır hücre keşif algoritması ile aday sınır hücreleri belirlenmektedir. Belirlenen aday sınır hücreleri için temel parametrelere ve fiyatlara karar verilmektedir. Fiyatların belirlenmesinde üç kriter kullanılmıştır:

- **En yakın sınır hücre kriteri:** Bu kriter için sabit ve çok yüksek bir tavan fiyat belirlenmektedir. Böylece maliyeti en düşük olan yani en yakın aday sınır hücrenin seçilmesi sağlanmaktadır.
- **En çok sınır kenar hücreye sahip sınır hücre kriteri:** Bu kriter için fiyat, sınır kenar hücre sayısının büyük bir sayı ile çarpılması ile hesaplanmıştır. Dolayısıyla çok sayıda sınır kenar hücreye sahip adaylar için fiyatlar daha da artmakta ve en çok sınır kenar hücreye sahip adayın seçilmesi garanti altına alınmaktadır.
- **Market sınır hücre kriteri:** Bu kriter için fiyat, sınır kenar hücre sayısı olarak belirlenmiştir.

Aday sınır hücrelerinin fiyatlarına karar verildikten sonra, üretici erkin bu görevleri tüketici erkinine bildirmektedir. İkinci aşamada, daha önceden belirlenmiş olan bir süre boyunca tüketici erkininden gelecek teklifleri beklemekte ve gelen teklifleri daha sonra değerlendirmek üzere kayıt altına almaktadır. Bekleme süresi dolduğunda, üretici erkin teklif değerlendirme olarak adlandırılan aşamaya geçmektedir. Bu aşamada Macar algoritmasını kullanarak hangi görevin hangi robota atanacağına karar verilmektedir.

Atama problemi ve Macar algoritması

Atama problemi, yöneylem araştırmasında en çok bilinen kombinatoriyal eniyileme problemlerinden birisidir. Bu problemin tanımı, her görev-kaynak atamasının maliyetinin bilindiği varsayılarak n görev ve n kaynağın atamalarından en iyi olanının seçilmesi olarak yapılabilir. Bir en iyi atama, toplam atama maliyetini en küçüklemektedir (Munkres, 1957). Atama probleminin matematiksel modeli aşağıda verilmiştir (Wolsey, 1998).

İndeksler

i : Görev indisi, j : Kaynak indisi

Kümeler

v_i : Görevler kümesi, u_j : Kaynaklar kümesi

Parametreler

n : Görev sayısı, m : Robot sayısı, p_{ij} : v_i 'nin u_j 'ye atanmasının kârı.

Karar Değişkenleri

$$x_{ij} = \begin{cases} 1 & \text{Eğer } v_i \text{ görevi } u_j \text{ kaynağına atanırsa} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

Model

$$enbz = \sum_{i=1}^n \sum_{j=1}^m p_{ij} * x_{ij} \quad (3.13)$$

kısıtlar altında

$$\sum_{i=1}^j x_{ij} = 1, j = 1, \dots, m \quad (3.14)$$

$$\sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n \quad (3.15)$$

$$x_{ij} \in 0, 1, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (3.16)$$

3.13 numaralı denklem modelin amaç fonksiyonudur ve toplam atama kârını en büyükmektedir. 3.14 denklemi her kaynağın tek bir göreve atanmasını garanti etmektedir. Benzer şekilde 3.15 denklemi her görevin bir kaynağa atanmasını garanti etmektedir. Denklem 3.16 karar değişkenlerinin 0-1 ve tamsayı olmasını garanti etmektedir.

Macar algoritması

Macar ya da diğer adıyla Kuhn-Munkres algoritması (Kuhn, 1955) ilk olarak 1955 yılında Kuhn tarafından önerilmiş ve 1957 yılında Munkres tarafından düzenlenmiştir (Munkres, 1957). Macar algoritması atama problemini $O(n^3)$ zamanda çözer. Burada n iki parçalı (bi-partite) çizgenin bir parçasının boyutunu göstermektedir. Macar algoritması, $G = (V, U, E)$ iki parçalı çizgesinin mevcut olduğu varsayımını yapmaktadır. Burada V ve U çizgenin her parçasındaki düğümler kümesini, E ise ayrıtlar kümesini göstermektedir. Macar algoritmasının adımları Algoritma 6'da verilmiştir (Korsah vd. 2007).

Algoritma 6: Macar Algoritması

Girdi: Sınır hücresi listesi İki parçalı çizge, $G = (V, U, E)$
Çıktı: En küçük maliyetli görev-kaynak eşleşmesi, M
Veri: $|V| = |U| = n$, $n \times n$ boyutlu kenar maliyetlerinin matrisi C
 Adım 1: Matristeki her satırdan, satırın en küçük değerini satırdaki bütün elemanlardan çıkar;
 Adım 2: İlk adımda elde edilen matristeki her sütundan, sütunun en küçük değerini sütundaki bütün elemanlardan çıkar;
 Adım 3: İkinci adımda elde edilen matristeki bütün sıfırları kapsayacak en az sayıda dikey ve dikey çizgiler çiz;
 // k çizgi sayısını gösterir
if $k < n$ **ise then**
 // s en küçük kapsanmamış matris elemanı olsun
 Her kapsanmamış matris elemanından s 'i çıkar;
 Her iki çizgiyle kapsanmış matris elemanına s ekle;
 Adım 3'ün başına geri dön;
end
if $k == n$ **ise then**
 Adım 4'e geç;
end
 Adım 4: En üstteki satırdan başlayarak, her satır için atama yap. Atama yapıldığında satır ve sütunu C matrisinden sil.

Görev atanmasında Macar algoritması

Bu çalışmada, Macar algoritması robotlara anlık görev atamalarının en büyük faydayı sağlayacak şekilde belirlenmesinde kullanılmıştır. Görev atamasında, geleneksel atama probleminden farklı olarak amaç fonksiyonu en büyükleme formundadır. Girdi

matrisi maliyetler yerine kârları göstermektedir. Bu matriste, pozitif değerler v_i işinin u_j robotuna atanması durumunda elde edilecek kârı göstermektedir. İlk olarak, tüm pozitif değerler girdi matrisinin en büyük değerinden çıkartılarak problem en küçükleme formuna dönüştürülmektedir. İkinci olarak, istenmeyen atamaların önlenmesi gerekmektedir (Kaleci, 2010).

Son aşamada, üretici erkin hangi görevi hangi robotun yapacağını belirledikten sonra, görevi yapmaya hak kazanan robotun tüketici erkinine bu durumu bildirmekte ve robotun hareket erkininden görevin bittiğine dair haber gelmesini beklemektedir. Robotun hareket erkini görevini bitirdiğini duyurduğunda tekrar yeni görevler üretilmekte ve bu süreç gidilecek herhangi bir sınır hücresi kalmayana kadar devam etmektedir (Kaleci, 2010).

3.2.2.3 Tüketici erkin

Tüketici erkin, üretici erkin tarafından belirlenen görevlere servis sağlamak amacıyla tasarlanmıştır. İki temel aşaması bulunmaktadır. İlk aşamada görevler için üretici erkin ile görüşmeleri kapsamaktadır. İkinci aşama ise görevi yapmaya hak kazanmış ise o görevi icra etmesi için hareket erkinine haber vermektedir. Tüketici erkin göreve üretici erkinden gelecek olan görev anonslarını bekleyerek başlamaktadır. Görev anonsu geldiğinde, robot görev yapma durumunda değilse bu görev anonsunu kabul etmektedir. Daha sonra, görevin maliyetine karar vermektedir. Bu yaklaşımda görevlerin maliyetleri robotun göreve olan uzaklığı olarak belirlenmiştir. Tüketici erkin görevin fiyatından kendi bulduğu maliyeti çıkararak bu görevden ne kadar kâr edeceğini belirlemekte, bu kârı üretici erkine göndermekte ve ondan gelecek cevabı beklemeye başlamaktadır. Robot görevi yapmaya hak kazanmış ise, hareket erkini aracılığı ile hedefe doğru hareket etmektedir (Kaleci, 2010).

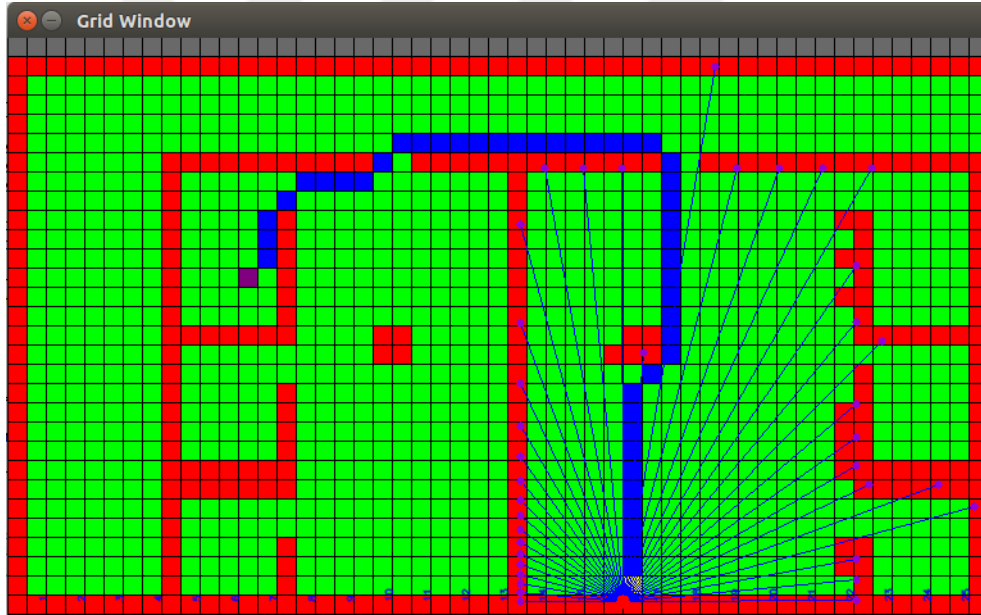
3.2.2.4 Hareket erkini

Hareket erkini gidilecek hedef belli olduktan sonra bu hedefe gitmek için gerekli olan yol planını hazırlamaktadır. Hazırlanan yol planı ortamın gösterimi ile yakından ilişkilidir. Örneğin ortam metrik harita ile ifade edildiğinde sığ öncelikli ya da derine öncelikli arama yöntemleri ile yol planı hazırlanabilir. Benzer şekilde A* metrik gösterim için sıklıkla kullanılan yol planı yöntemlerinden biridir. Diğer taraftan ortam topolojik harita ile gösterildiğinde Dijkstra ve Floyd-Warshall gibi en kısa yol bulma algoritmaları kullanılabilir. Hareket erkini yol planını hazırladıktan sonra robotun güvenli bir şekilde (sabit ya da hareket eden engellere çarpmadan) hedefine varmasını sağlamak ile sorumludur. Robot hedefine vardığında hareket erkini, üretici ve tüketici erkinlere hedefe vardığını ve yeni görev için hazır olduğunu bildirmektedir.

Robotun seyrüsefer görevi için metrik gösterim ile A* (Hart vd. 1968) kullanılarak yol planı yapılmıştır. Bu noktadan sonra metrik harita üzerinde çalıştırılan geleneksel A* algoritmasının dezavantajları ve genişletilmiş A* algoritması anlatılacaktır. Yol planları elde edildikten sonra, robotun yol planını izlerken engellerden sakınması için Vektör Alan Histogramı (VAH) kullanılmıştır.

A* Yol Planlama Algoritması

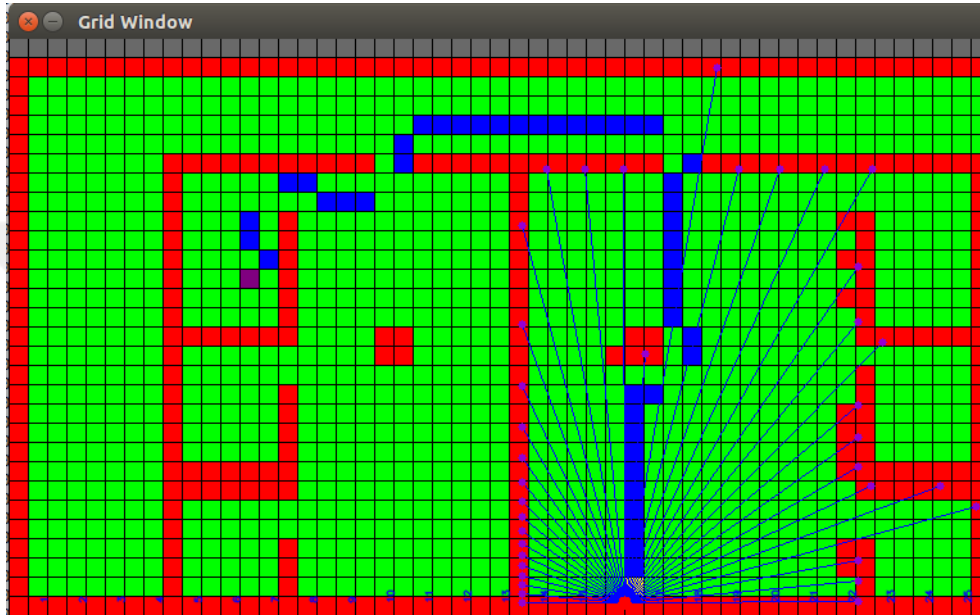
A* yol planlama algoritması ilk olarak Peter Hart, Nils Nilsson ve Bertram Raphael tarafından 1968 yılında önerilmiştir (Hart vd. 1968). A* algoritması başlangıç noktasından hedef noktaya en az maliyetli yolu bulabilmek için çizge arama yöntemlerini kullanmaktadır. Bu arama sırasında birbiriyle ilişkisi olan bir hücre kümesi elde edilmektedir. Bu hücrelerin komşuları bir öncelik kuyruğunda tutularak sonraki hücrelerin bulunması sağlanmaktadır.



Şekil 3.8: Standart A* Algoritması

Bu çalışmada robottan alınan konum bilgisi başlangıç noktası olarak kullanılmaktadır. Daha önceden belirlenmiş hedef noktaları da sırasıyla hedef olarak kullanılmaktadır. Izgara erkini kısmında nasıl oluşturulduğu detaylı bir şekilde anlatılan metrik haritadaki boş hücreler kullanılarak A* algoritması çalıştırıldığında bir hücre listesi yol planı olarak elde edilmektedir. Elde edilen yol planı Şekil 3.8'de koyu mavi renkle gösterilmiştir. Ayrıca engellere karşılık gelen dolu hücreler kırmızı renkle, boş hücreler ise yeşil renkle gösterilmiştir. Hedef sol üst odada mor renkteki hücrede yer alırken, robot ise

sağ alttaki geniş odanın en altındaki sarı hücrede bulunmaktadır. Şekil incelendiğinde, A* hedef ile başlangıç arasındaki en kısa yolu bulmayı amaçladığından, engellere çok yakın hücreler yol planı içerisinde yer almaktadır. Bu durum robotun sıradaki hücreye gitme isteğiyle engelden sakınma davranışı arasında kalarak kararsız hareket etmesine neden olmaktadır. Bunun üstesinden gelebilmek için, Algoritma 7'de verilen yöntem kullanılmıştır. Elde ettiğimiz yol planı Şekil 3.9'da verilmiştir.



Şekil 3.9: Genişletilmiş A* Algoritması

Vektör Alan Histogramı (VAH) Bina içi ortamlarda görev yapan gezgin robotlar için engelden sakınma durumu kaçınılmaz olarak ele alınması gereken bir konudur. Üretilen yol planlarının engel içermemesine karşın robot herhangi bilinmeyen bir engelle karşılaşabilir. Bu durumda görevine devam edebilmek için bu engelden güvenli bir şekilde sakınması gerekmektedir. Geçmiş çalışmalarda bu problemin üstesinden gelmek için ilk olarak potansiyel alanlar yaklaşımı kullanılmıştır (Khatib, 1985). Bu yöntemde hedef çekici, engeller ise itici kuvvetler oluşturmaktadır. Bu kuvvetlerin birleşimi robotun uygulayacağı hareketi belirlemektedir. Bu yöntemin hesaplama maliyetinin düşük olması en büyük avantajıdır. Diğer yandan robot hedefine varmadan zaman zaman uygulanacak kuvvet sıfır olabilmekte ve robotun o noktada durup beklemesi de yöntemin en büyük dezavantajı olarak kabul edilmektedir. Dinamik Pencere Yaklaşımı (Fox vd. 1997) ise robotun uygulayabileceği hızlanma ve yavaşlama ivmelerini göz önüne almaktadır. Bu yaklaşımda robotun gitmek istediği hedef ve bir sonraki engele olan uzaklık kullanılarak bir amaç fonksiyonu belirlenmekte ve bu amaç fonksiyonunun değerini en büyüleyecek açışal

Algoritma 7: Standart A* algoritmasına eklenen genişleme

Girdi: A* ile elde edilen hücre listesi *yol*

Çıktı: Gözden geçirilmiş hücre listesi *degisen_yol*

```

for i ← 1 to yol_boyutu do
  yol(i + 1)yon ← yol(i) ve yol(i + 1)
  if yol(i + 1)yon == sag ya da yol(i + 1)yon == sol then
    if yol(i)ust == engel then
      | degisen_yol(i) ← yol(i)alt
    else if yol(i)alt == engel then
      | degisen_yol(i) ← yol(i)ust
    end
  else if yol(i + 1)yon == ust ya da yol(i + 1)yon == alt then
    if yol(i)sag == engel then
      | degisen_yol(i) ← yol(i)sol
    else if yol(i)sol == engel then
      | degisen_yol(i) ← yol(i)sag
    end
  else if yol(i + 1)yon == sol_ust then
    if yol(i)ust == engel then
      | degisen_yol(i) ← yol(i)alt
    else if yol(i)sol == engel then
      | degisen_yol(i) ← yol(i)sag
    end
  else if yol(i + 1)yon == sag_ust then
    if yol(i)ust == engel then
      | degisen_yol(i) ← yol(i)alt
    else if yol(i)sag == engel then
      | degisen_yol(i) ← yol(i)sol
    end
  else if yol(i + 1)yon == sol_alt then
    if yol(i)sol == engel then
      | degisen_yol(i) ← yol(i)sag
    else if yol(i)alt == engel then
      | degisen_yol(i) ← yol(i)ust
    end
  else if yol(i + 1)yon == sag_alt then
    if yol(i)sag == engel then
      | degisen_yol(i) ← yol(i)sol
    else if yol(i)alt == engel then
      | degisen_yol(i) ← yol(i)ust
    end
  else
    | degisen_yol(i) ← yol(i)
  end
end

```

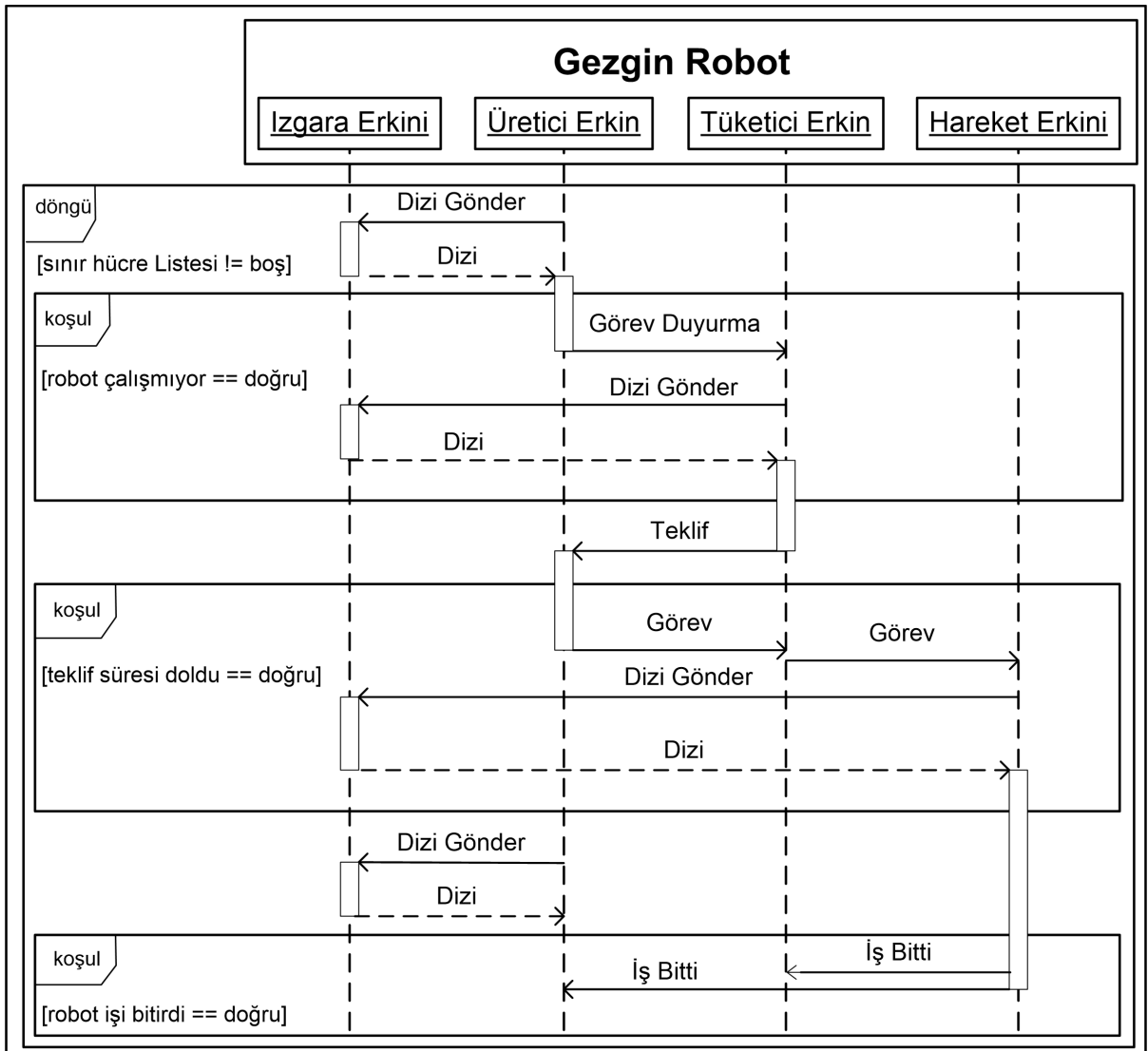
ve doğrusal hızlar uygulanmak üzere seçilmektedir. Bu yöntem yüksek hızlarda güvenli seyrüsefer yapılmasını sağlarken yüksek işlem gücüne ihtiyaç duymaktadır.

VAH ise yüksek hızlı robotlar için hızlı ve güvenilir bir yöntem sunmaktadır. Bu yöntem, iki boyutlu kartezyen bir histogram ızgarası kullanılmaktadır. Bu ızgara sürekli olarak robotun üzerindeki algılayıcı bilgileri kullanılarak güncellenmektedir. VAH yöntemi robotun istenilen açısal ve doğrusal hızlara karar verebilmek için iki aşamalı bir veri indirgemesi gerçekleştirmektedir. İlk aşamada yöntem, iki boyutlu histogram ızgarasını robotun anlık konumunun etrafında oluşturduğu tek boyutlu polar histograma çevirmektedir. Bu polar histogram daha önceden belirlenen sayıda sektörler bölünmekte ve sektörler içlerindeki engellerin oranına göre bir değere sahip olmaktadır. İkinci aşamada ise düşük engel değerine sahip sektörler arasından en uygun olanı seçilmekte ve bu sektörün yönü robotun yönü olarak seçilmektedir (Borenstein ve Koren, 1991), (Ulrich ve Borenstein, 1998). Bu raporda sunulan çalışmada Akçakoca'nın (Akçakoca vd. 2014) yaptığı çalışma lazer mesafe verisi kullanılarak gerçekleştirilmiştir. VAH yönteminin başarısını önemli ölçüde seçilen parametreler belirlemektedir. Örneğin, iki boyutlu kartezyen histogram ızgarasının boyutları ve çözünürlüğü robotun engelden sakınma ve hedefe gitme davranışlarını dengeli yapabilmesi için kritiktir. Benzer şekilde, seçilen maksimum açısal ve doğrusal hızlar da bu davranışları önemli ölçüde etkilemektedir. Ayrıca hedef ve engellerin üretilecek olan hızlara hangi oranda katkı sağlayacağı ya da polar histogramın açısal çözünürlüğünün belirlenmesi önemlidir. Bu parametreler tek tek denenerek ve etkileri gözlemlenerek robotun yapması istenen harekete göre ayarlanmıştır.

3.2.2.5 Erkinler arası iletişim

Sistem mimarisinde bulunan dört erkin detaylı olarak anlatıldıktan sonra bu erkinler arasındaki iletişime kısaca göz atmak faydalı olacaktır. Süreç, tüketici erkinin ızgara erkininden ortamın güncel durumunu ifade eden doluluk ızgarasını tutan diziyi istemesiyle başlamaktadır. Izgara erkini diziyi gönderdiğinde tüketici erkin sınır hücre yaklaşımını kullanarak aday sınır hücrelerini belirlemekte, seçilen fiyatlandırma kriterine göre adayların fiyatlarını hesaplamakta ve bu görevleri tüketici erkine göndererek bu görevler için gelecek teklifleri beklemeye başlamaktadır. Eğer robot herhangi bir görevi gerçekleştirme durumunda değilse üreticiden gelen görevlerin maliyetini hesaplayabilmek için ızgara düğümünden güncel dizi istenmektedir. Dizi geldikten sonra A* yol planlama algoritması kullanılarak robotun bulunduğu yerden aday görevlere gidilmesi durumunda ortaya çıkacak olan maliyetler belirlenmektedir. Tüketici erkin her görev için görevin fiyatından maliyetini çıkararak robotun bu görevi yaptığında ne kadar kâr elde edeceğini hesaplamakta, üretici erkine tekliflerini göndermekte ve ondan gelecek cevabı beklemeye başlamaktadır. Üretici erkin teklifleri beklerken gelen teklifleri kaydetmektedir. Bekleme süresi dolduğunda

robotun kârını en büyükleyecek şekilde robotun bir sonraki görevini belirlemekte ve tüketici erkine göndermektedir. Tüketici erkin görev bekleme süresi içinde üreticiden cevap alırsa bu görevi hareket erkinine iletmektedir. Bekleme süresi dolduğunda herhangi bir cevap gelmezse süreci en baştan tekrar başlatmak üzere üretici düğüm ızgara düğümünden dizi istemektedir. Eğer hareket düğümü görevi aldıysa yol planı yapmak için diziyi istemekte, yol planını yapmakta ve VAH yardımıyla hedefine doğru yola çıkmaktadır. Görev bittiğinde ise bu durum tüketici ve üretici düğümlere duyurularak yeni görevler için hazır olunduğu belirtilmektedir. Şekil 3.10'da erkinler arası iletişimi gösteren sıralama diyagramı verilmiştir.



Şekil 3.10: Erkinler Arası İletişim (Sıralama Diyagramı)

3.3 Sonular

Bu b3l3mde ilk olarak gereklenen yaklařımın verimlilięini 3lmek iin kullanılan benzetim ortamı ve robot kontrol aray3z3 detaylı olarak anlatılacaktır. Daha sonra deneysel kurulum b3l3m3nde deney ortamı erkinleri alıřtırırken kullanılan parametreler gibi detaylar yer alacaktır. Son olarak farklı hedef seme kriterleri iin g3rsel ve sayısal sonular verilerek verimlilikleri tartıřılacaktır.

3.3.1 Robot kontrol aray3z3

Robotların otonom olarak kendilerine verilen g3revi yapabilmeleri iin ortamdan aldıkları algılayıcı verilerine g3re bazı kararlar vermesi ve uygulaması gerekmektedir. Bu uygulamalar sonucunda ortamın durumu deęiřecek ve tekrar yeni kararlar ile d3ng3 bu şekilde devam edecektir. Bu noktada, ortamın algılanması ve kararlar verilebilmesi iin bir robot kontrol aray3z3ne ihtiya duyulmaktadır. Mobility Open Architecture Simulation and Tools (MOAST), Player ve onun uzantısı Robot Operating System (ROS) ve řu anda kullanılmayan ve desteklenmeyen Pyro gibi eřitli robot kontrol aray3zleri mevcuttur. Bu robot kontrol aray3zleri arasında desteęi hala s3ren ve pop3lerlięi her geen g3n artan ROS dięerlerinden bir adım 3ne ıkmaktadır.

ROS, robot yazılımlarını geliřtirmek iin esnek bir robot kontrol aray3z3d3r. G3n3m3zde sayısı ve eřitlilięi ok artan robot platformları iin karmařık ve g3rb3z robot davranıřlarını yaratmayı kolaylařtırmak amacıyla k3t3phaneler ve aralar barındırmaktadır. Bunun sebebi, her robot platformunda alıřan g3rb3z robot yazılımları geliřtirilmesinin olduka zor olmasıdır. İnsanlar iin ok kolay gibi g3r3nen problemler eřitli ortam ve g3revlerde robotlar iin olduka zorlayıcı olabilmektedir. Bunların 3stesinden gelmek tek bir laboratuvar ya da tek bir enstit3n3n tek bařına bařarabileceęi bir iř deęildir. Sonu olarak ROS iřbirliki robotik yazılımlarını geliřtirmeyi cesaretlendirmektedir. 3rneęin, bir laboratuvar i ortamlarda haritalamada uzman olabilir ve d3nya apında bir sistemde haritalama g3revine katkıda bulunabilir. Dięer bir grup haritaları kullanarak seyr3sefer yapmakta uzman olabilir ya da bařka bir grup bilgisayarda g3r3 konusunda uzman olabilir. ROS bu şekilde bir alanda uzman olan grupların iřbirlięini saęlamak amacıyla tasarlanmıřtır (ROS, 2016).

ROS'un alıřma prensibini anlamak iin 3ncelikle bazı tanımlamaların yapılması gerekmektedir. ROS'da alıřma alanı (workspace) adı verilen ana bir klas3r bulunmaktadır. Bu klas3r3n altında bulunan alt klas3rlere ise paket (package) adı verilmektedir. Her pakette CMakeList.txt adı verilen ve paketin derlenmesi iin gerekli olan verileri ieren bir dosya bulunmaktadır. Benzer şekilde package.xml dosyası ise paketin derleme ve alıřma

zamanı bağımlılıklarını içermektedir. Her bir paket bir görev için geliştirilmiştir. Paketin görevini yapabilmesi için düğüm (node) adı verilen yürütücülere ihtiyaç duyulmaktadır. Bu düğümler diğer paketlerdeki ya da kendi paketinin içindeki diğer yürütücüler ile haberleşebilmek için kullanılmaktadır. Yürütücüler birbirleriyle konular (topics) üzerinden haberleşmektedirler. Düğümler konulara mesaj yayınlayabilmekte ve konudan mesaj alabilmektedirler (Nodes, 2016; Topics, 2016).

3.3.2 Benzetim ortamı

USARSim, kentsel arama ve kurtarma robotları ve ortamları için çok robotlu sistemlerin koordinasyonu ve insan robot iletişimi çalışmalarında kullanılmak üzere doğruluğu yüksek bir araştırma platformu olarak tasarlanmıştır. USARSim benzetim ortamı otoyol robotları, DARPA yarışmasındaki robotlar, futbol robotları, su altı robotları, insansı robotlar ve uçan robotlar gibi çok çeşitli robotların çalıştığı ortamları destekleyebilmektedir. USARSim, National Institute of Standards' and Technology (NIST) ile birlikte kentsel arama ve kurtarma görevleri için tasarlanmış ve zorluk seviyesi sarı, turuncu ve kırmızı olan fiziksel afet ortamlarına göre standartlaştırılmıştır. USARSim, kullanıcı arayüzü elemanlarını doğru bir şekilde yorumlayarak insan robot iletişimini desteklemektedir. Bunu yapabilmek için robotların davranışlarını doğru bir şekilde benzetim ortamına yansıtarak operatörlerin robotlar hakkındaki farkındalığını artırmaktadır. USARSim benzetim ortamının sağladığı bu avantajlar oyun motoru üzerinde çalışmasından kaynaklanmaktadır. Benzetim ortamının son sürümü çeşitli ortam modelleri, ticari ve deneysel robot platformları ve algılayıcı modellerini içermektedir (Usarsim, 2016).

USARSim benzetim ortamı robotların TCP/IP soketleri aracılığı ile doğrudan kontrol edilmesine izin vermektedir. Bu şekilde, araştırmacılar robotun üzerinde bulunan lazer, Kinect, kamera, ses üstü gibi algılayıcılardan doğruluğu yüksek olan verileri elde edebilmekte ve aynı zamanda robotların motorlarına uygulanacak tork ya da hız gibi komutları da gene TCP/IP soketleri aracılığı ile benzetim ortamına gönderebilmektedirler (Carpin vd. 2007).

Bununla birlikte, USARSim benzetim ortamının Windows işletim sistemi ile ve oyun motoru üzerinden çalışması, benzetim ortamında istenilen ortamların yaratılmasını, bu ortamların değiştirilmesini zorlaştırmaktadır. Aynı zamanda ROS'un linux işletim sistemi ile çalışması benzetim ortamının kullanılması için iki bilgisayar gerekliliğini getirmektedir. İki bilgisayar arasında TCP/IP soketleri aracılığı ile çalışan sistemin hızı da oldukça düşük olmaktadır. Bu noktada, Gazebo benzetim ortamı bir seçenek olarak ortaya çıkmıştır.

Gazebo, bir robotun karşılaşılabileceği değişken çevre şartlarını türetebilmek için tasarlanmıştır. Benzetilen bütün nesnelere, itildiğinde, çekildiğinde, devrildiğinde veya taşındığında gerçekçi davranış sergilemesini sağlayan kütle, hız, sürtünme gibi çok sayıda özelliğe sahiptir. Gazebo, katı gövdeler ve çeşitli eklem tipleri de dahil olmak üzere, dinamik benzetimler için Açık Dinamik Motoru kullanmaktadır. Yaratılan dünyalar ve modeller, yeni ortamların hızlı bir biçimde yaratılabilmesi için XML formatındaki dosyalarda tanımlanır. Ek olarak, şimşek, gölgeler ve sis gibi karmaşık görsel etkilere de olanak tanınmaktadır (Fink vd. 2009). Gazebo'nun en temel özelliklerinden birisi, lazer algılayıcılardan, 2 ve 3 boyutlu kameralardan, Kinect tarzı algılayıcılardan, dokunmaya duyarlı algılayıcılardan ve daha pek çok bileşenden, isteğe bağlı olarak gürültülü, algılayıcı verisi üretebilmesidir. Ayrıca, PR2, Pioneer2 DX, iRobot Create, Turtlebot gibi pek çok robot platformu da Gazebo içerisine dahil edilmiştir (Gazebo, 2016). Böylece, Gazebo çok robotlu sistemlerin yeni ve ilginç şekillerde test edilmesini ve hızlı bir biçimde geliştirilmesini sağlayan zengin bir benzetim ortamı sunmaktadır (Koenig ve Howard, 2004).

Gazebo benzetim ortamı, ilki “gzserver” ve ikincisi “gzclient” olmak üzere iki farklı programı çalıştırır. “gzserver” programı fizik güncelleme döngüsünü ve algılayıcı veri üretimini çalıştırır. Bu program Gazebo'nun çekirdeğini oluşturur ve herhangi bir grafik arayüzü ile bağımsız olarak kullanılabilir. “gzclient” programı da QT tabanlı kullanıcı arayüzünü çalıştırır. Bu uygulama benzetimin güzel bir görüntüsünü ve çeşitli benzetim özellikleri üzerinde kontrol imkânını sağlar (Gazebo, 2016).

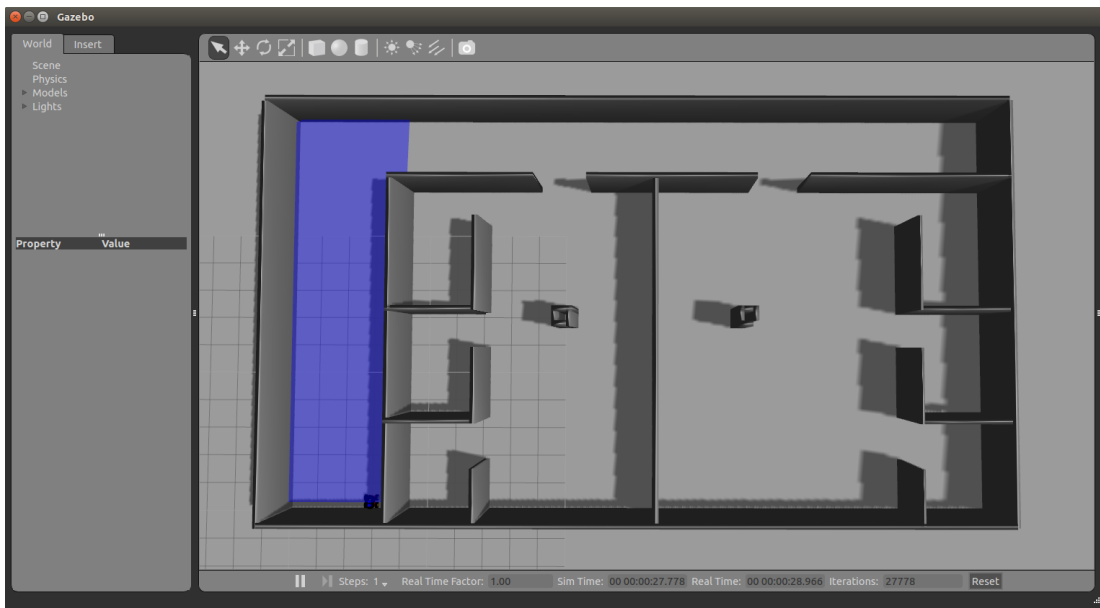
Son olarak, 2009 yılında, John Hsu, ROS'u ve PR2 robotunu Gazebo ortamına aktarmıştır. Gazebo o zamandan beri ROS topluluğunca en çok kullanılan temel araçlardan biri durumuna gelmiştir. 2012 yılında Açık Kaynaklı Robotik Vakfı (OSRF), Gazebo projesinin temel geliştiricisi durumuna gelmiştir. Yetenekli bir grup bireyin yoğun geliştirme çabaları sonucunda, OSRF Gazebo'yu, DARPA Robotik Yarışması (DARPA Robotics Challenge)'nın bir parçası olan, Sanal Robot Yarışması'nı (Virtual Robotics Challenge) işletmek için kullanmıştır.

Sonuç olarak, yukarıda bahsedilen deprem ortamlarına benzer ortamları hızlıca oluşturabilme kabiliyeti, ROS ile birlikte tek bir bilgisayar kullanarak kolaylıkla çalışması ve Sanal Robot Yarışması gibi önemli bir organizasyonda kullanılmış olması Gazebo'yu USARSim'in bir adım önüne taşımaktadır.

3.3.3 Deneysel kurulum

Önerilen yöntem, Gazebo benzetim ortamında modellenen Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Laboratuvar binasında test edilmiştir (Şekil 3.11). Ortamın boyutları 26×15 metredir ve 52×30 'luk metrik ızgara ile ifade edilmiştir. Benzetimlerde lazer mesafe algılayıcısı bulunan Pioneer P3-AT robotu kullanılmıştır. Lazer ışınlarının menzilleri 10 metre olarak seçilmiştir. Izgara hücrelerinin durumunu güncellemek için 5'er derece aralıklı 37 tane lazer ışını kullanılmıştır. Eğer bir hücreden 3 adet lazer ışını geçebiliyorsa bu hücre boş olarak kabul edilmektedir. Izgaranın güncel durumu saniyede bir ekrana getirilmektedir. Bir hücre bir boyutta 15 adet piksel ile ifade edilmektedir. Üretici erkinin ızgara erkininden diziyi aldıktan sonra sınır hücreleri belirlemek için en fazla 5 saniye süresi bulunmaktadır. Üretici erkin tüketiciden gelecek olan teklifleri ise en fazla 15 saniye beklemektedir. Tüketici erkin ise üretici erkinden gelecek cevabı 20 saniye beklemektedir. VAH'da ise iki boyutlu histogram 3×1 metre olarak seçilmiş ve 0.1 metre çözünürlükle kullanılmıştır. Azami doğrusal hız 200 mm/sn ve açısal hız 0.60 rad/sn olarak seçilmiştir. Hedef noktanın ağırlığı 0.4, engellerin ise 0.6 olarak belirlenmiştir. Polar histogramın açısal çözünürlüğü 10 derece olarak seçilmiştir. Hareket erkini saniyede bir kartezyen ve polar histogramlarını güncellemekte, uygun açısal ve doğrusal hızları hesaplamaktadır.

Robotun kontrolü için ROS kullanılmıştır. Robotun benzetim ortamında hareket edebilmesi ve odometre algılayıcıdan elde edilen konum bilgileri için p2os paketi (p2os, 2016) kullanılmaktadır.



Şekil 3.11: Test ortamı (Gazebo)

Uygulamalar gerçekleştirilirken A* yol planı ile birlikte üç farklı fiyatlandırma kriteri (en yakın sınır hücre, en çok kenar hücreesine sahip sınır hücre ve market sınır hücre) kullanılmıştır. Robot üç farklı yerden görevine başlamakta ve belirlenen fiyatlandırma kriterine göre bütün ortamın keşfi tamamlanana kadar görevini sürdürmektedir. Şekil 3.12, Şekil 3.14 ve Şekil 3.16'da sırasıyla en çok kenar hücreesine sahip sınır hücre, en yakın sınır hücre ve market sınır hücre fiyatlandırma kriterine göre robotun 24, 4, 1.57 konumundan başladığı durumdaki görsel sonuçlar verilmiştir. Şekillerde siyah çemberlerin içinde kırmızı rakamlarla belirtilen konumlar sırasıyla seçilen hedefleri göstermektedir. Bu çemberler arasında kalan kırmızı noktalar ise robotun izlediği yolları ifade etmektedir. Elde edilen sayısal sonuçlar Çizelge 3.1'de verilmiştir.

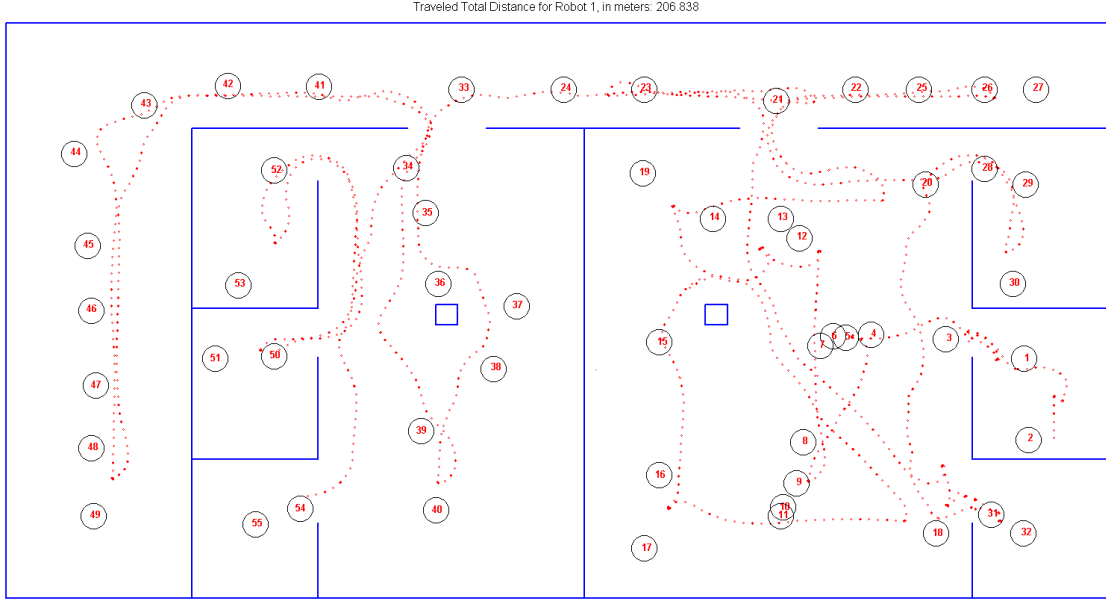
3.3.4 Deneysel sonuçlar

Çizelge 3.1: Tek Robot A* Sonuçları

Başlangıç Konumu	Fiyatlandırma Kriteri	Mesafe (Metre)	Süre (Dakika)
2.5, 0.3, 1.57	En çok kenar hücreesi	249.0206	92
	En yakın	137.7664	71
	Market	129.4584	66
9, 7, 1.57	En çok kenar hücreesi	193.9504	80
	En yakın	165.8054	77
	Market	171.0292	78
24, 4, 1.57	En çok kenar hücreesi	206.8380	83
	En yakın	156.6460	73
	Market	161.0887	71

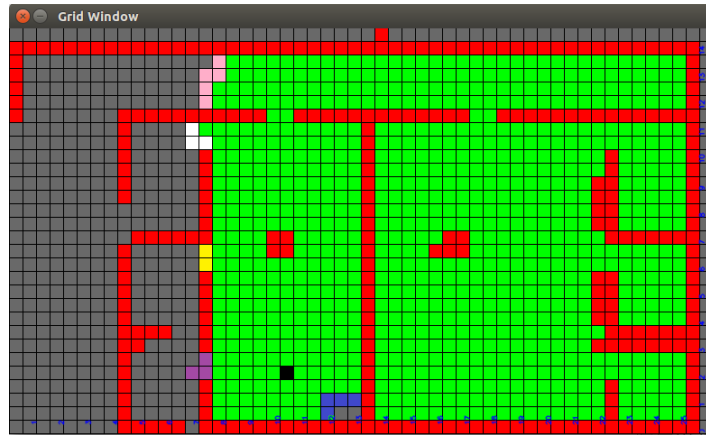
Fiyatlandırma kriterlerinden en çok kenar hücreesine sahip sınır hücre kriteri kullanıldığında, robotun üç farklı başlangıç durumunda da, market ve en yakın sınır hücre kriterlerine göre daha uzun mesafeler katettiği ve keşfin daha uzun süre aldığı açıkça görülmektedir. Bu durum, robotun sürekli olarak farklı bir yerde ortaya çıkan daha çok kenar hücreesine sahip sınır hücreesine gitme istediğinden kaynaklanmaktadır. Şekil 3.12'de en çok kenar hücreesine sahip sınır hücre kriterinin yaptığı hedef seçimleri görünmektedir.

Şekil 3.13'te ise robotun 40. hedefindeyken 41. hedefi belirlemesi esnasındaki doluluk ızgarasının durumu gösterilmektedir. Şekilde robot siyah kare ile gösterilen yerde bulunmaktadır. Robot gidebileceği aday sınır hücreleri farklı renklerle gösterilmiştir. Şeklin üst kısmında turkuvaz renklerle gösterilen aday hedef beş sınır kenar hücreesi içermekte ve robota en uzak konumda bulunmaktadır. En çok kenar hücreesine sahip sınır hücre



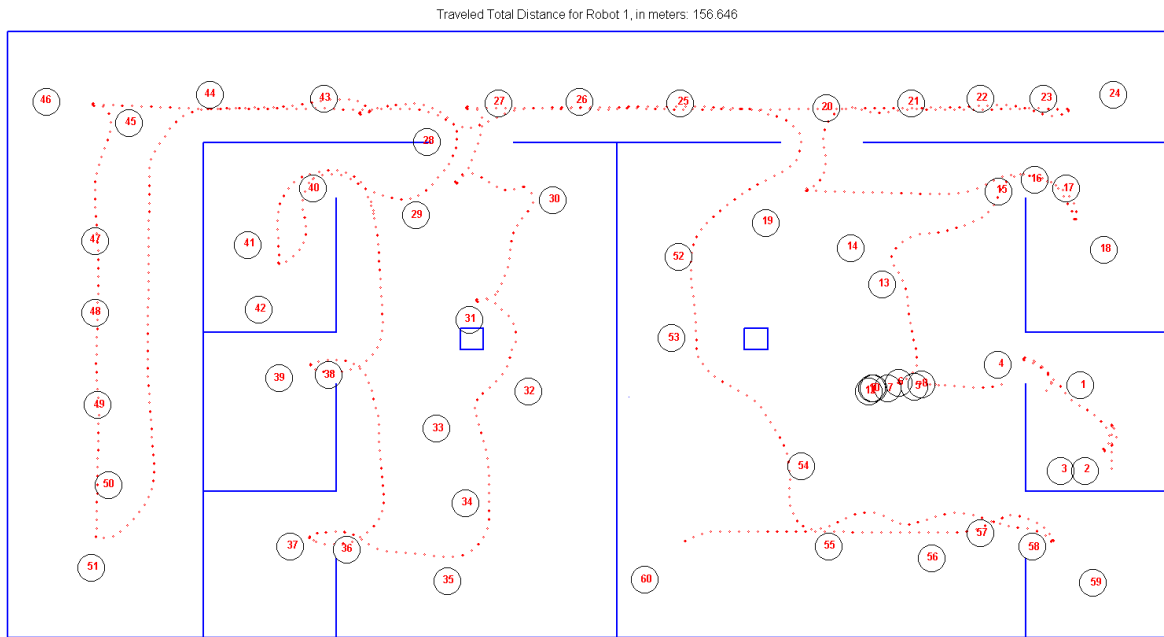
Şekil 3.12: En çok kenar hüresine sahip sınır hücre kriteri

fiyatlandırma kriteri kullanıldığında robot bu hedefi seçecektir. Bununla birlikte, robotun sağ alt tarafında lacivert renkle gösterilen hedef ise dört sınır kenar hücresi içermekte ve robota en yakın konumda bulunmaktadır. Ancak bu kriterde sadece en çok sınır kenar hücre sayısı hedef seçildiği için robot sürekli farklı konumlara gitme eğiliminde olmaktadır. Bu durum robotun daha fazla yol kat etmesine ve zaman kaybetmesine yol açmaktadır.



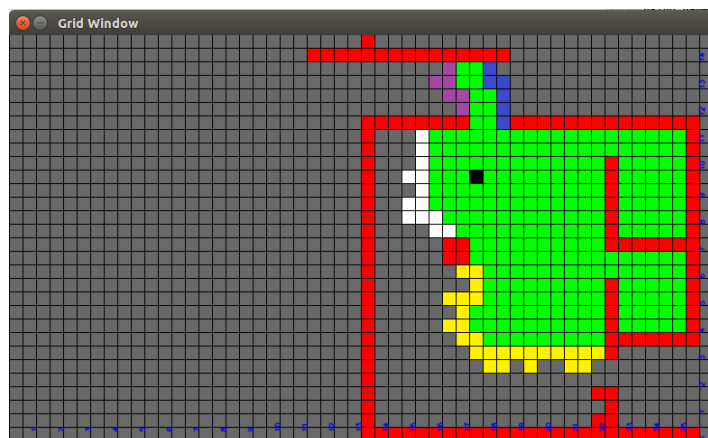
Şekil 3.13: En çok kenar hüresine sahip sınır hücre örneği

En yakın sınır hücre kriteri kullanılarak fiyatlandırma yapıldığında robot o sınır hücresine gittiğinde ne kadar yeni bilgi kazanacağını önemsemeden en yakındaki sınır hücresine hareket etmektedir. Şekil 3.14'te en yakın sınır hücre kriterinin yaptığı hedef seçimleri gösterilmiştir.



Şekil 3.14: En yakın sınır hücre kriteri

Şekil 3.15'te ise robotun 19. hedefindeyken 20. hedefi belirlemesi esnasında doluluk ızgarasının durumu gösterilmektedir. Şekilde robot siyah kare ile gösterilen yerde bulunmaktadır. Robotun gidebileceği aday sınır hücreleri farklı renklerle gösterilmiştir. Şeklin üst kısmında lacivert renklerle gösterilen aday hedef altı sınır kenar hücresi içermekte ve robota en yakın olanıdır. En yakın sınır hücre fiyatlandırma kriteri kullanıldığında robot bu hedefi seçecektir. Bununla birlikte, robotun sol tarafında beyaz renkle gösterilen hedef ise on üç sınır kenar hücresi içermektedir. Ancak bu kriterde sadece en yakın hedefi seçtiği için robot sürekli kendine yakın konumlara gitme eğilimindedir. Bu durum ortamın büyük kısımlarının daha geç keşfedilmesine neden olmaktadır.



Şekil 3.15: En yakın sınır hücre kriteri örneği

4. TEK ROBOTLU TOPOLOJİK KEŞİF YÖNTEMİ

Geçmiş çalışmalarda ortamlar genellikle metrik ve topolojik haritalar ile ifade edilmektedir. Bununla birlikte, son yıllarda her iki gösterimin iyi yanlarını öne çıkaran melez yöntemler de geliştirilmeye başlanmıştır. Metrik gösterimin temel avantajı, kolay oluşturulabilmesidir. Ancak bu gösterimde dikkat edilmesi gereken en önemli nokta ızgarada bulunan hücre sayısına nasıl karar verildiğidir. Ortam yüksek sayıda hücre ile ifade edildiğinde gösterimin hassaslığı artmaktadır. Bu durum hafıza ve zaman karmaşıklığı açısından verimliliği düşürmektedir. Diğer yandan, az sayıda hücre sayısı da gösterimin hassasiyetini düşürmektedir. Topolojik gösterimlerde ise ortam düğümler ve bu düğümleri birbirine bağlayan ayrıtlardan oluşmaktadır. Topolojik gösterimler düşük hafıza ve zaman karmaşıklığı sağlamaktadır. Ancak, topolojik haritaların oluşturulması ve güncellenmesi metrik haritalara göre daha fazla emek istemektedir (Thrun, 1998).

Bu bölümde, ilk olarak geçmiş çalışmalarda yer alan topolojik harita oluşturma, spektral kümeleme ile topolojik harita oluşturma ve tek robotlu topolojik keşif yaklaşımlarından öne çıkanlar incelenecektir. Daha sonra, gerçekleştirilen yaklaşım alt bölümünde normalleştirilmiş kesimler problemi tanıtılacak ve spektral kümeleme ile ilişkisi ortaya konulacak, spektral kümeleme kullanılarak topolojik haritanın oluşturulması detaylı olarak verilecek ve oluşan topolojik haritanın keşifte nasıl kullanıldığı anlatılacaktır. Son olarak, deneysel kurulum verilecek ve topolojik ve metrik harita kullanımının keşif üzerindeki olumlu ve olumsuz yönleri tartışılacaktır.

4.1 Geçmiş Çalışmalar

4.1.1 Topolojik harita oluşturma yaklaşımları

Topolojik harita oluşturmaya yönelik ilk çalışmalarda genellikle Voronoi Diyagramı (VD) kullanılmıştır. VD bir düzlem üzerinde daha önceden belirlenmiş bölgelerden en az iki ya da daha fazlasına eşit uzaklıkta bulunan noktalar kümesi olarak tanımlanmaktadır (Aurenhammer, 1991). Choset (Choset, 1996) yaptığı çalışmada Genelleştirilmiş Voronoi Diyagramı'nın (GVD) tanımını şöyle yapmaktadır: "GVD, VD'nin konveks engeller için genişletilmiş halidir öyleki m sayıda konveks engele m boyutta eşit uzaklıkta olan noktalar kümesidir." Bununla birlikte, eğer $m > 2$ ise oluşturulan düğümlerin bağlanmasına gerek olmadığını vurgulamıştır. Bu durumda, Choset Hiyerarşik Genelleştirilmiş Voronoi Diyagramı'nı (HGVD) bütün düğümlerin bağlı bir ağa dahil olduğunu garanti eden bir yöntem olarak tanımlamış ve robot için yol planı yapılırken bu yöntemin verimli olarak

kullanılabileceğini göstermiştir. Thrun ise topolojik harita oluştururken ilk adım olarak ortamın VD'sini çıkartmaktadır. Bu noktada, VD'ye dahil olan noktalar kümesinden kritik noktalara karar verilmektedir. Eğer, bir noktanın en yakın olduğu engellerle arasındaki mesafe o noktanın belirli bir komşuluğunda bulunan noktaların en yakın olduğu engellerle arasındaki mesafeden daha küçükse bu nokta kritik nokta olarak adlandırılmaktadır. Kritik noktalardan, onlara en yakın engellere çizilen çizgiler kritik çizgi olarak tanımlanmakta ve bu kritik çizgilerin ortamı bölgelere ayırmasıyla ortamın topolojik haritası oluşmaktadır (Thrun, 1998).

Topolojik harita oluşturmaya yönelik çalışmalarda VD dışında yöntemler de bulunmaktadır. Buschka ve Saffiotti oda benzeri alanları bölütlemektedir. Yöntemin önemli avantajı topolojik haritanın adım adım oluşturulmasıdır. Bu sayede algılayıcı gürültüsünün ve hesaplama karmaşıklığının azaltılması amaçlanmıştır. Bu yöntemde, araştırmacılar ultrasonik algılayıcılardan elde edilen verileri kullanarak yerel ve bulanık bir doluluk ızgarası oluşturmaktadır. Daha sonra bu ızgarayı gri seviyede bir resim olarak düşünmekte ve bu resimde yer alan boş bölgelerin şekli hakkında bilgi edinmek için bulanık matematiksel morfoloji yöntemini kullanmaktadırlar. Son olarak, bu bilginin topolojik yapısını çıkarabilmek için bulanık dijital topoloji yaklaşımı uygulanmıştır (Buschka ve Saffiotti, 2002).

Tapus ve Siegart yaptıkları çalışmada ortamın topolojik haritasını oluştururken parmak izi tabanlı bir gösterim kullanmışlardır. Bu gösterim sayesinde robotta bulunan bütün algılayıcılardan alınan bilgileri kullanarak, algılayıcı gürültülerini azaltmakta ve alanların farklı özelliklerini ortaya çıkarabilmektedirler. Bu çalışmada iki adet 180 derecelik lazer mesafe algılayıcı ve bir adet her yöne (omnidirectional) kamera kullanılmış; parmak izi özellikleri ise lazer verisinden elde edilen köşeler ve kameradan elde edilen dikey ayrıt bilgileri olarak seçilmiştir. Topolojik haritaya yeni bir düğüm eklerken klasik yöntemlerden (belirli zaman ya da mesafe) farklı olarak robotun çevresindeki değişiklik temel alınmaktadır. Parmak izi özelliklerinde tutulan değerler bir önce yaratılan düğümüne göre belirli bir eşik değerinin üstündeyse yeni bir düğüm yaratılmaktadır (Tapus ve Siegart, 2006).

Mozos ve Burgard ortamın topolojik haritasını çıkarmak için lazer mesafe verilerini kullanmışlardır. Bu çalışmada, bina içi ortamların genelde oda, koridor ve kapı olmak üzere üç anlamsal sınıf ile ifade edilebileceği varsayımı yapılmaktadır. Bu noktada AdaBoost adı verilen denetimli öğrenme yöntemi ile ortamı anlamsal sınıflara ayırmışlardır. Daha sonra, sınıflandırma hatalarını azaltmak amacıyla olasılıksal gevşetme etiketlemesi uygulanmıştır. Elde edilen her farklı alan için bir düğüm yaratılmış ve düğümler birbirlerine anlamsal

sınıflardan yararlanılarak bağlanmıştır. Bu şekilde ortamın topolojik haritası elde edilmiş olmaktadır (Mozos ve Burgard, 2006).

Joo ve diğerleri yaptıkları çalışmada metrik haritadan sanal kapı olarak tanımladıkları kavramı kullanarak topolojik harita üretmişlerdir. Doluluk ızgarasındaki dolu hücrelerin bazıları köşe özelliği olarak kabul edilmiş ve bu özellikler kullanılarak sanal kapılar ve odaların sınırları belirlenmiştir. Sanal kapılar ortamı topolojik olarak bölmekte kullanılmaktadır. Bu noktadan sonra genetik algoritma ile topolojik haritadaki düğümler birleştirilmekte ve ayrıtlar kaldırılarak topolojik haritaya son hali verilmektedir. Bu haritada, düğümler sanal kapılar ile ayrılırken, ayrıtlar gerçek kapılarda yer almaktadır (Joo vd. 2010).

Konolige ve diğerleri önerdikleri yöntemde metrik harita ve topolojik haritayı birlikte kullanmaktadırlar. Metrik haritadan konumlandırma ve seyrüsefer için, topolojik haritadan ise yol planı için yararlanmaktadırlar. Topolojik harita, yerel metrik haritanın robotun hareketine göre kaydırılmasıyla oluşturulmaktadır. Doluluk ızgarasında belirli bir menzil içindeki boş olan hücreler belirlenmekte, hedefe ulaşmak için en yakını seçilmekte ve ona ilerlenmektedir. O noktaya gidene kadar da doluluk ızgarası güncellenmektedir. İstenilen noktaya ulaşıldığında orada bir düğüm oluşturulup, bu süreç devam ettirilmektedir. Basit ama etkili olan bu yöntemin dezavantajı ise sadece yerel doluluk ızgarasını göz önünde bulundurmasıdır (Konolige vd. 2011).

Werner ve diğerleri topolojik haritalamanın en zor problemlerinden olan robotun odometre ölçümü alamadığı ya da alanların özgün parmak izlerine sahip olmadığı durumları ele almışlardır. Bu durumlarda, robotun geçmişte elde ettiği algılayıcı ölçümlerinden (lazer ya da resim) yararlanılmaktadır. Mevcut andan başlayarak robotun daha önceden belirlenmiş bir sayıdaki geçmişinde elde ettiği ölçümler kullanılarak topolojik harita oluşturulmaktadır. Bununla birlikte, yazarların asıl motivasyon noktası belirtilen kısıtlar altında robotun daha önce uğradığı bir düğümün farkına varmasıdır. Bunu yapabilmek için bu noktanın komşuluğunda aldığı bilgileri kullanmaktadır. Yöntem hem ultrasonik hem de görsel algılayıcı kullanılarak test edilmiş ve başarılı bir şekilde bulunduğu düğümün farkına varıldığı ve topolojik haritanın doğru bir şekilde oluşturulup güncellendiği deneysel sonuçlarla gösterilmiştir (Werner vd. 2012).

Bir diğer çalışmada ise iki ve üç boyutlu metrik haritalardan topolojik harita çıkarılmaktadır. Bu süreç beş adımda gerçekleşmektedir. İlk adımda üç boyutlu metrik harita iki boyutlu metrik haritaya indirgenmekte ve filtreleme yapılmaktadır. İkinci adımda kapılar tespit edilmektedir. Kapıların metrik haritadaki minimum ve maksimum boyutları göz önüne alınarak bu işlem gerçekleştirilmektedir. Üçüncü adımda VD üretilmektedir.

Dördüncü adımda ise VD ile kapıların yerleri kontrol edilmekte ve VD üzerindeki bütün noktalardan daireler çizilerek topolojik harita oluşturulmaktadır. Son adımda ise sanal ve gerçek duvarlar kullanılarak topolojik haritaya son hali verilmektedir. Bu çalışmanın önemli noktası, anlamsal kelime, düğümlerin geometrik tanımlaması, görsel özellikler, alan ve merkezi nokta özelliklerinin düğümlerde saklanmasıdır (Santos vd. 2013).

Topolojik harita üretmek üzere yapılan son araştırmalardan birinde lazer verisi kullanılarak doluluk ızgarası oluşturulmuş ve bu ızgara bir resim gibi düşünülüp ortamın iskeleti çıkarılmıştır. Bununla birlikte, iskeletin robotun seyrüsefer yapabilmesine uygun olması için bazı koşullar tanımlanmış ve bu koşullara uymayan pikseller kullanılmamıştır. Bu noktadan sonra 1-piksel ince (one-pixel thin) yöntemi kullanılarak iskeletin birleşme, son ve ayırıt üzerindeki noktalarına karar verilmiştir. Bu iskeletin birleşme ve son noktaları topolojik haritanın düğümleri olarak kabul edilmiştir. Düğümler birleşme ve son noktalar arasında kalan ayırıtlarla bağlanmıştır (Li ve Qiu, 2015).

4.1.2 Spektral kümeleme ile topolojik harita oluşturma yaklaşımları

Son yıllarda, spektral kümeleme yöntemi (Von Luxburg, 2007) robotik alanında da sıklıkla kullanılmaya başlanmıştır. Bunun sebebi, temel olarak diğer kümeleme yöntemlerine göre daha kolay gerçekleştirilmesi ve standart doğrusal cebir algoritmaları ya da yazılımları ile kümelemeyi yapabilmesidir. Ayrıca, k-ortalama gibi geleneksel kümeleme yöntemlerinden daha iyi kümeleme sonuçları ürettiği iddia edilmektedir.

Spektral kümeleme kullanarak topolojik harita oluşturmaya çalışan ilk yaklaşımlardan biri Blanco ve diğerleri tarafından önerilmiştir (Blanco vd. 2006). Bu çalışmada robotun algılayıcılarından gelen bilgiler kullanılarak ortak algılanan bölgeler kümelene çalışılmaktadır. Dolayısıyla, ard arda alınan algılayıcı verileri arasındaki benzerlikler kullanılarak ortak algılanan bölgeler belirlenmekte ve bu bölgeler topolojik haritanın düğümleri olarak kabul edilmektedir. Birbiri üzerine gelmiş olan kısımlar ise düğümleri bağlayan ayırıtları oluşturmaktadır. Önerilen yaklaşım lazer mesafe verileri ile test edilmiş ve elde edilen metrik ve topolojik harita sunulmuştur.

Zivkovic ve diğerleri metrik haritaları temel seviye haritalar olarak tanımlamışlar ve metrik haritaları geometrik ve görünüş tabanlı gösterimler olarak kategorize etmişlerdir. Geometrik tabanlı gösterim, ses üstü algılayıcı ya da lazer mesafe algılayıcı kullanılarak oluşturulurken, görünüş tabanlı gösterimler ise robotun kamerasından alınan resimler ile meydana gelmektedirler. Topolojik haritayı oluşturmak için sadece görünüş tabanlı temel seviyedeki haritayı spektral kümelemenin girdisi olarak kullanmışlardır. Görünüş tabanlı

temel seviyedeki haritaya uygulanabildiği için önerdikleri yöntemin VD'ye göre daha genel olduğunu iddia etmektedirler (Zivkovic vd. 2006).

Bir diğer çalışmada yazarlar ham lazer mesafe verilerini kullanarak topolojik haritayı çevrimiçi olarak oluşturmaktadır. Robot adım adım ortamda ilerlerken topolojik haritasını oluşturmakta ve bu haritayı spektral kümeleme kullanarak alt haritalara bölmektedir. Daha sonra AdaBoost denetimli öğrenme algoritması kullanılarak lazer verileri ile alt haritaların birbirinden nasıl ayırt edileceği açıklanmaktadır. Son olarak, robot daha önceden ziyaret ettiği bir yere geldiğini fark ettiğinde topolojik harita son halini almaktadır (Brunskill vd. 2007).

Spektral kümelemede kullanılacak olan özellikler her zaman robota ve robotun algılayıcılarına bağlı olmayabilir. (Vázquez-Martín vd. 2009) yaptıkları çalışmada yukarıda anlatılan üç yöntemde olduğu gibi mevcut haritayı alt haritalara bölmeyi amaçlamaktadırlar. Ancak, bu yaklaşımlardan farklı olarak ortamın özellikleri düğümlerin oluşmasında kullanılmıştır. Bunu yapabilmek için birlikte görülebilirlik çizgesini (Covisibility Graph) kullanılmaktadır. Bu çizgenin düğümleri gözlemler tarafından oluşturulurken ayrıtlar ise robotun aynı konumda bulunduğu gözlemlerde tespit edilen özellikleri ifade etmektedir. Daha sonra bu çizge kullanılarak ortam kümelerine ayrılmaktadır.

Jia ve diğerleri yaptıkları çalışmada topolojik haritayı oluşturmak için üç katmanlı bir yapı kullanılmaktadır. En alt katmanı yerel özellik katmanı olarak adlandırmaktadırlar ve bu katmanda kabul edilebilir uzay ağacı (admissible space tree) yöntemi kullanarak topolojik haritanın düğümlerini belirlemektedirler. Aynı zamanda SIFT (Scale-Invariant Feature Transform) ile farklı düğümler arasındaki ilişkiyi ortaya çıkarmakta ve elde edilen eşleşme sonuçlarını eniyilemek için RANSAC yöntemini kullanılmaktadır. İkinci katmanda ortamın VD'sini oluşturmaktadırlar. Son katmanda, ilk iki katmanda oluşturdukları düğümlerin aralarındaki benzerlikleri kullanarak spektral kümeleme ile ortamı kümelerine bölmüş ve topolojik haritayı oluşturmuşlardır (Jia vd. 2013).

Son yıllarda yapılan bir çalışmada, yarı yapılandırılmış (semi-structured) ortamların topolojik haritasının çıkarılması incelenmiştir. Yarı yapılandırılmış ortamı ise farkedilebilir büyüklükte engeller içeren geniş açık alan olarak tanımlamışlardır. Bu noktada, ortamı GVD kullanarak bölütlemişlerdir. Daha sonra, bölütlenen ortam parçalarını spektral kümeleme kullanarak kümelemiş ve ortamın topolojik haritasını oluşturmuşlardır. Son olarak, geçmiş çalışmalarda genellikle yer almayan ancak kümeleme performansını etkileyen ortamın kaç bölgeye bölünmesinin ve kümeleme yapılırken bölütler arasındaki benzerliği tanımlarken hangi mesafenin kullanılmasının doğru olacağı gibi konuları tartışmış ve deneysel sonuçlarla yapılacak seçimlerin etkilerini göstermişlerdir (Liu vd. 2014).

Yapılan son çalışmalarda topolojik haritanın oluşturulması ile elde edilen çizge kullanılarak ortamın anlamsal haritası da oluşturulmaya başlanmıştır. Luperto ve diğerlerinin önerdiği yaklaşımda daha önceki çalışmalarda olduğu gibi ilk önce ortam alt parçalara bölütlenmektedir. Daha sonra, benzerliğe göre parçalanmış alt haritalar kümelenebilir. Her alt küme bir düğüm ile gösterilmektedir. Bu süreç robot adım adım ortamda ilerlerken devam etmektedir. Bu noktada, her düğüme oda ve koridor anlamsal sınıflarından biri atanmaktadır. Daha sonra, komşu olan oda düğümleri kendi aralarında kümelenebilir bir alt küme oluştururken koridor düğümleri sayesinde bu bölütlenmiş parçalar topolojik haritaya bağlanmaktadır (Luperto vd. 2015).

Bu tez kapsamında kullanılan yaklaşıma benzer bir çalışma Choi ve diğerleri tarafından önerilmiştir. Yazarlar düşük maliyetli ultrasonik algılayıcılar kullanarak ortamı topolojik olarak ifade edecek bir yaklaşım önermişler ve önerdikleri yöntemi ev benzeri bir ortamda test etmişlerdir. Metrik haritanın oluşturulması aşamasında yazarların daha önceden önerdiği bir yaklaşım doluluk ızgarasına uygulanmıştır. Daha sonra, metrik haritaya hücre ayrıştırma yöntemi uygulanarak haritadaki boş hücreler belirlenmiştir. Bu noktada, elde edilen boş hücreler spektral kümelemede kullanılarak ortamın topolojik haritası elde edilmiştir. Önerilen yaklaşımın temel dezavantajı topolojik haritanın gerçek zamanlı değil çevrimdışı bir yöntem kullanılarak oluşturulmasıdır (Choi vd. 2011).

Liu ve diğerleri ise topolojik harita oluştururken spektral kümeleme yönteminin kullanılması durumunda ortaya çıkabilecek dezavantajları incelemişlerdir. İlk olarak, ilginlik matrisinin boyutunun büyümesi durumunda spektral kümeleme işleminin hesaplama maliyeti çok büyük olmaktadır. Öte yandan, spektral kümeleme her çalıştığında farklı sonuçlar üretebilmektedir. Son olarak, spektral kümeleme yapılırken verinin kaç kümeye bölüneceği sorusu ortaya çıkmaktadır. Bu sayının küçük ya da büyük olması durumunda kabul edilemez sonuçlar meydana gelebilmektedir. Yazarlar, bu dezavantajların üstesinden gelmek için Choi-Liu adını verdikleri ağaç yapısını kullanmışlardır. Bu ağacı, en büyük ağırlıklara sahip en küçük kapsayan ağaç olarak tanımlamışlar ve ayrıt ağırlıklarını ortak bilgi adını verdikleri bir ölçüt kullanarak hesaplamışlardır (Liu vd. 2011).

4.1.3 Tek robotlu topolojik keşif yaklaşımları

Rekleitis ve diğerleri yaptıkları çalışmada, ortamın hali hazırda bir topolojik haritasının olduğunu varsaymışlardır. Robot daha önceden uğranmış olan düğüm ya da ayrıtları daha önceden uğranmamış olanlardan ayrıt edememektedir. Bununla birlikte, bulunduğu düğümlerle ilişkili olan bazı özellikleri algılayabilmektedir. Robot bir düğümden başlayarak bir diğer düğüme gitmekte ve her seferinde sadece bir düğüm hakkındaki bilgiyi güncellemektedir. Ulaştığı düğüm için düğüme giren ayrıtı bilmekte ancak çıkan ayrıtı

bilmemektedir. Burada saat yönünün tersine ilk ayrıtı seçerek keşfine devam etmekte ve her yeni düğümü bütün düğümler bir döngü oluşturacak şekilde mevcut düğümlere eklemektedir. Bu şekilde keşfe devam ederken bir noktada döngü kapanacak ve robot bunu farkederek keşfe son verecektir (Rekleitis vd. 1999).

Bir diğer çalışmada iki katmanlı bir planlama yapılarak robotun ortamı keşfetmesi sağlanmaktadır. İlk olarak ortamın doluluk ızgarası ve topolojik haritası oluşturulmaktadır. Daha sonra, keşif problemi şu şekilde tanımlanmaktadır: 1) Metrik haritadaki bütün bilinmeyen hücreler sadece bir kez ziyaret edilmelidir. 2) Bunu gerçekleştirirken robot mümkün olduğunca az mesafe kat etmelidir. Bu tanımlama esasında gezgin satıcı problemi ile aynıdır. Dolayısıyla, gezgin satıcı probleminin çözümünü veren bir sezgisel ile küresel katmanda topolojik harita kullanılarak yol planı yapılmaktadır. Topolojik haritadaki düğümler arasındaki seyrüsefer ise yerel katman ile planlanmaktadır. Bunu yapabilmek için yazarlar iki düğüm arasında bulunan bölgedeki birbirine komşu olan hücrelerin oluşturduğu alanı tespit etmektedirler. Bu alan engeller göz önüne alınarak belirlenmektedir. Bu noktada, bu alanın iskeleti çıkarılmaktadır. Son olarak, robot, bu iskeleti çekici, iskelet dışında kalan bölgeyi ise itici güç kabul ederek gideceği düğüme potansiyel alanlar yöntemi ile ulaşmaktadır (Poncela vd. 2002).

Jia ve diğerleri de keşfin verimliliğinin artması için hem metrik hem de topolojik haritayı kullanmışlardır. Yazarlar seçilen hedef ve yol planı gibi etkenlerin verimliliği önemli ölçüde etkilediğini belirtmektedirler. Dolayısıyla, hedef seçiminin keşif verimliliğini artırması için hem topolojik hem de metrik bilgiyi hesaba katan bir fayda fonksiyonu önermişlerdir. Bu fonksiyon, hedefe ulaştığında kazanacağı bilgiyi, hedefe gitmek için gerekli olan maliyeti hesaba kattığı gibi aynı zamanda ortamın duvar, ada ve mağara gibi sınıflandırılan topolojik özelliklerini de göz önünde bulundurmaktadır. Önerdikleri yaklaşımın önemli katkılarından biri ise taşma maliyeti (cost overflow) adını verdikleri ve metrik tabanlı çalışan yol planlamasıdır (Jia vd. 2004).

Son olarak, hedef seçiminde konkav düğüm kavramını kullanan yaklaşım ele alınacaktır (Cheong vd. 2008). Yazarlar metrik haritayı elde ettikten sonra haritanın iskeletini çıkarmaktadırlar. Daha sonra bu iskeletten topolojik haritanın düğümlerini belirlemektedirler. Dört tip düğüm tanımlamışlardır: Dallenma düğümü, üç ayrıttan daha fazla ayrıta sahip olan düğümdür. Hedef düğümleri ise ilk çocuk düğümlerinden meydana gelmektedir. Konkav düğüm bir dallanma ve iki bitiş düğümlerine sahip olan düğümdür. Bitiş düğümü ise sadece bir ayrıtı olan düğüm olarak tanımlanmaktadır. Son olarak, bu düğümleri kullanarak robotun ziyaret edeceği hedefler belirlenmektedir.

Bu alt bölümde verilen geçmiş çalışmalar göz önüne alındığında topolojik harita oluşturma ve bu haritanın keşifte kullanılması esnasında çok farklı yaklaşımların olduğu açıktır. Bu tez çalışması kapsamında büyük ölçekli bir ortamın metrik haritasını kullanarak ortamın topolojik haritasını oluşturmak ve oluşan topolojik haritayı da keşifte kullanmak hedeflenmiştir. Metrik harita oluşturulduktan sonra, topolojik haritanın oluşturulması için spektral kümeleme yöntemi seçilmiştir. Bunun sebebi, temel olarak diğer kümeleme yöntemlerine göre daha kolay gerçekleştirilebilmesi ve standart doğrusal cebir algoritmaları ya da yazılımları ile kümelemeyi yapabilesidir. Ayrıca, VD üretmek için Fortune algoritması kullanıldığında zaman karmaşıklığı $O(n \log n)$ olurken spektral kümeleme Shi ve Malik'in önerdiği algoritma ile çözümlerse zaman karmaşıklığı $O(n)$ olmaktadır. Bu noktada (Liu vd. 2011)'e göre, spektral kümeleme yönteminin göz önüne alınması gereken bazı dezavantajları bulunmaktadır. Bunlardan birisi ilginlik matrisinin boyutudur. Bu dezavantajı bertaraf etmek amacıyla, önerilen yöntemde sadece belirli anlarda elde edilen boş hücreler spektral kümelemede kullanılmaktadır. Böylece, ilginlik matrisinin boyutu çok fazla büyümemekte ve topolojik harita aşamalı olarak çevrimiçi bir şekilde oluşturulmaktadır. Spektral kümelemenin ikinci dezavantajı ise küme sayısına karşı hassas olmasıdır. Bu durumda, küme sayısı, bir kümeyi oluşturmak için birleştirilebilecek yeni boş hücre sayısı göz önüne alınarak uyarlamalı olarak belirlenmektedir. Bu adaptasyonun uygulanmasındaki temel düşünce, topolojik haritanın bütün ortamı kapsayacak şekilde olması ve ortamın mümkün olduğunca az sayıda düğüm ile ifade edilmesidir. Üçüncü dezavantaj ise tekrar edilemez sonuçlardır. Bunu önlemek için k-ortalama yönteminin ilk küme merkezi atamalarında basit bir sezgisel algoritma kullanılmıştır. Kümeleme tamamlandıktan sonra, küme merkezleri düğüm olarak tanımlanmakta ve en küçük kapsayan ağaç algoritmasıyla birbirine bağlanmaktadır (Kaleci vd. 2015b).

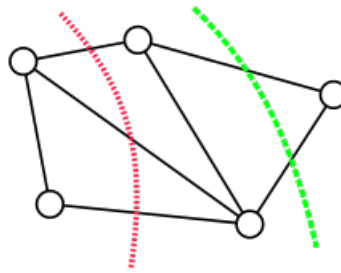
Geçmiş çalışmalarda vurgulandığı üzere hedef ve yol planı seçimi keşfin verimliliğini önemli ölçüde etkilemektedir. Topolojik keşif yönteminde metrik keşif yönteminde olduğu gibi hedefler sınır hücre yaklaşımı ile belirlenecek ve verilen üç kriter gereği en iyisi seçilecektir. Bu noktada, elbette geçmiş çalışmalarda yer alan yaklaşımlarda olduğu gibi farklı hedef belirleme yöntemleri kullanılabilir. Ancak, tezin ana katkısı anlamsal sınıfların kullanılması ile hedeflerin belirlenmesi olduğu için farklı hedef belirleme yöntemleri kapsam dışında bırakılmıştır. Diğer yandan, yol planı yapılırken hem metrik hem de topolojik haritanın kullanıldığı daha karmaşık ama verimli yöntemler seçilebilir. Benzer sebep yüzünden çizge üzerinde uygulanabilen en kısa yol bulma algoritmalarından Dijkstra kullanılarak yol planı yapılmıştır. Bu alt bölümden sonra gerçekleştirilen yaklaşım ve deneysel sonuçlar detaylı olarak verilecektir.

4.2 Gerçeklenen Yaklaşım

4.2.1 Normalleştirilmiş kesim problemi

Topolojik gösterimlerde ortamın belli bölgelerinde düğümler oluşturulmaktadır. Bu noktada, düğümlerin nasıl ve nerede oluşturulacağı sorusu ortaya çıkmaktadır. Geçmiş çalışmalar incelendiğinde, kullanılan yöntemlerde genellikle ortamı bir şekilde parçalara ayırarak bu parçalar yardımıyla düğümler oluşturulduğu görülmüştür. Ortamı belli parçalara ayırmak için metrik haritadan yararlanılabilir. Bu aşamada, metrik harita bir resim gibi düşünülüp resim bölütleme algoritmaları kullanılarak ortam parçalara ayrılabilir. Bu algoritmalar arasında en popüler olanı Shi ve Malik tarafından önerilmiştir (Shi ve Malik, 2000). Bu algoritma normalleştirilmiş kesim probleminin çözümünün ortamı parçalara ayırmakta kullanılabileceği ana fikri üzerine kurulmuştur. Bu noktada, çizge teorisinde kullanılan kesim ve en küçük kesim gibi kavramların tanımlarını vermek uygun olacaktır.

Kesim, V düğüm kümesinin, S ve $\bar{S} = V - S$ şeklinde iki parçaya bölünmesi olarak tanımlanabilir. Her bir kesim, bir uç noktası S , diğer uç noktası ise \bar{S} alt kümelerine denk gelen bir ayrıklar kümesini belirtmektedir. Bu şekilde tanımlanan bütün kesimler arasında en küçük ayrık sayısı ya da ayrık ağırlığına sahip kesim en küçük kesim olarak tanımlanmaktadır (Ahuja vd. 1993). Şekil 4.1'de kesim ve en küçük kesim için örnekler verilmiştir. Kırmızı kesikli çizginin geçtiği ayrıklar kümesi bir kesimi göstermektedir. Bununla birlikte, yeşil kesikli çizgi ise en küçük kesimin sahip olduğu ayrıklar üzerinden geçmektedir.



Şekil 4.1: Kesim ve en küçük kesim örnekleri (Cut, 2016)

$G = (V, E)$ yönsüz, ağırlıklı bir çizge olsun. Burada, V düğümler kümesini, E ağırlık değerleri negatif olmayan ayrıklar kümesini göstermektedir. w_{ij} de i 'nci ve j 'nci düğümler arasındaki ayrığın ağırlığına karşılık gelmektedir. Bu ağırlık, i 'nci ve j 'nci düğümler arasındaki benzerliğin bir fonksiyonu ile hesaplanmaktadır. W , boyutları düğümler kümesindeki eleman sayısı olan simetrik kare bir matrisi ifade etmekte ve her düğüm için w_{ij} ağırlıklarını içermektedir.

Bu noktada yapılmak istenen V düğümler kümesini $A \subset V$ ve $B \subset V$ alt kümelerine bir kesim ile ayırmaktır. Ancak bu kesim aşağıda verilen iki ölçütü mümkün olduğunca sağlamak zorundadır.

- Alt kümelerin (A ya da B) kendi içlerindeki düğümler arasındaki benzerliğin yüksek olmasıdır. Bu ölçütü hesaplamak için Denklem 4.1 kullanılmaktadır. Bu denklem, A alt kümesinin düğümlerinden V kümesinin düğümlerine olan tüm ayrıtların ağırlıklarının toplamını ifade etmektedir.

$$iliski(A, V) = \sum_{u \in A, v \in V} w_{uv} \quad (4.1)$$

- Farklı alt kümelerde bulunan düğümler arasındaki benzerliğin düşük olmasıdır. Bu ölçütü hesaplamak için Denklem 4.2 kullanılmaktadır. Bu denklem, A alt kümesinin düğümlerinden B alt kümesinin düğümlerine olan tüm ayrıtların ağırlıklarının toplamını ifade etmektedir.

$$kesim(A, B) = \sum_{u \in A, v \in B} w_{uv} \quad (4.2)$$

Sonuç olarak, alt kümelerin içindeki düğümlerin birbirlerine yüksek oranda benzediği ve iki farklı alt kümede bulunan düğümler arasındaki benzerliğin az olacağı bir kesim bulmak normalleştirilmiş kesim problemi olarak adlandırılmaktadır (Shi ve Malik, 1997). Normalleştirilmiş kesimi ($Nkesim$) hesaplamak için Denklem 4.3 kullanılmıştır.

$$Nkesim(A, B) = \frac{kesim(A, B)}{iliski(A, V)} + \frac{kesim(A, B)}{iliski(B, V)} \quad (4.3)$$

Son olarak, en küçük normalleştirilmiş kesim ($EKNkesim$),

$$EKNkesim = argenk_{A,B}(Nkesim(A, B)) \quad (4.4)$$

denklemleri ile tanımlanmaktadır.

En küçük normalleştirilmiş kesim problemi NP-Tam bir problemdir. Bununla birlikte, Shi ve Malik (Shi ve Malik, 2000) en küçük normalleştirilmiş kesim problemini

belirli koşullar altında genelleştirilmiş özdeğer problemine dönüştürmeyi başarmışlardır. Genelleştirilmiş özdeğer problemi Denklem 4.5'te verilmiştir.

$$(D - W)y = \lambda Dy \quad (4.5)$$

Denklemden, W ilginlik matrisi (affinity matrix) olarak tanımlanmakta ve düğümler arasındaki benzerliği ifade etmektedir. D , elemanları W 'nin satırlarının toplamına karşılık gelen köşegen bir matristir ve Denklem 4.6'da gösterilmiştir.

$$D(i, i) = \sum_j W(i, j). \quad (4.6)$$

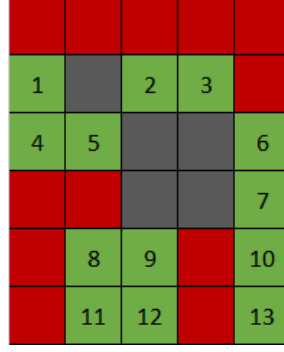
Genelleştirilmiş özdeğer problemi çözüldüğünde, en küçük özdeğere karşılık gelen özvektör sıfır olduğu için, ikinci en küçük özdeğere karşılık gelen özvektör en küçük normalleştirilmiş kesim probleminin çözümü olmaktadır. Shi ve Malik'in (Shi ve Malik, 2000) belirttiği üzere, ikinci özvektörde sadece iki farklı değer bulunmaktadır. Bu şekilde, V iki kümeye parçalanabilmektedir. Çizmeyi k parçaya ayırabilmek için, ardışık en küçük özdeğerlere denk gelen özvektörler kullanılır.

Standart bir özdeğer probleminin çözümü $O(n^3)$ zaman karmaşıklığına sahiptir. Ancak, W az sayıda düğüm birbirine bağlı olduğu için genellikle seyrek bir matristir. Ayrıca düğüm kümesini alt kümelere bölebilmek için sadece birkaç özvektörün yeterli olması bu problemin çözümünü $O(n)$, başka bir deyişle polinom, zaman karmaşıklığına indirgemektedir (Shi ve Malik, 2000).

4.2.2 Topolojik haritanın oluşturulması

Topolojik haritanın oluşturulmasında metrik haritadan yararlanılmaktadır. Metrik haritadaki boş hücreler topolojik harita oluşumunun temelini teşkil etmektedir. Örnek bir metrik harita Şekil 4.2'de verilmiştir. Şekildeki kırmızı, gri ve yeşil hücreler sırasıyla dolu, bilinmeyen ve boş hücreleri göstermektedir. Boş hücrelerin içindeki numaralar hücrelerin indekslerini belirtmektedir.

Denklem 4.5'te verilen genelleştirilmiş özdeğer problemini çözebilmek için ilk olarak ilginlik matrisinin (W) metrik haritadaki boş hücreler kullanılarak oluşturulması gerekmektedir. W , metrik haritadaki boş hücre sayısı boyutlarında kare, simetrik ve negatif



Şekil 4.2: Metrik Harita Örneği

olmayan ağırlıklar içeren bir matristir. Şekil 4.2'de verilen metrik harita örneği için oluşturulan ilginlik matrisi Şekil 4.3'te gösterilmiştir.

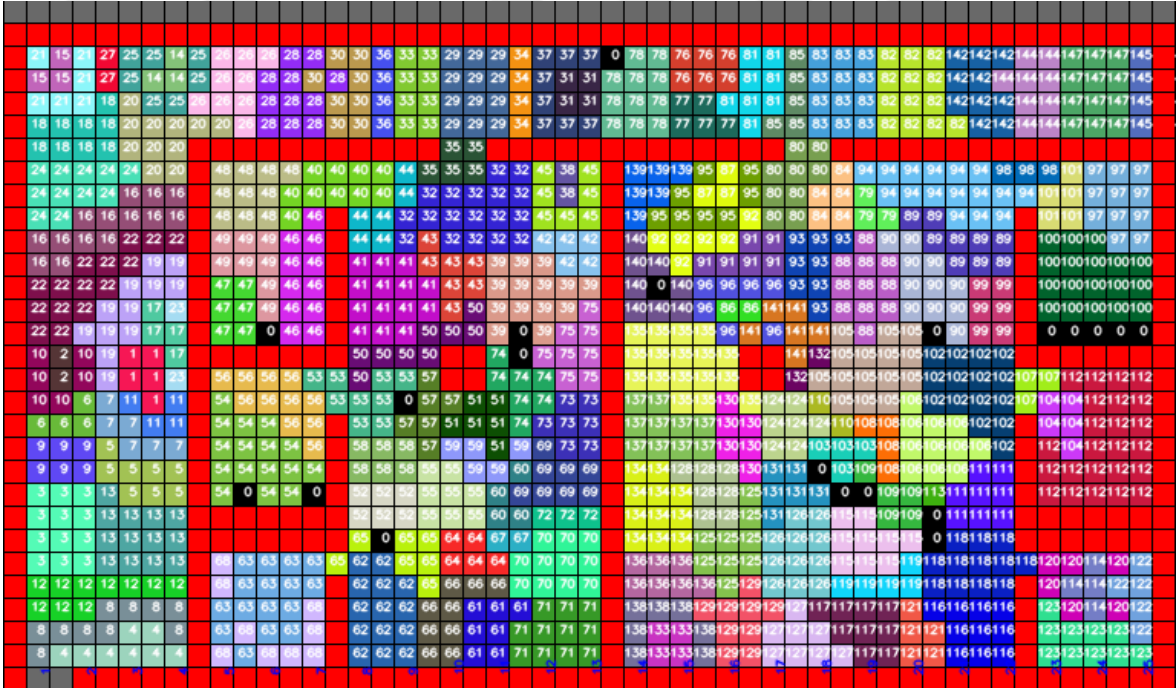
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	0	0.9	0.7	0	0	0	0	0	0	0	0
2	0	1	0.9	0	0.7	0	0	0	0	0	0	0	0
3	0	0.9	1	0	0	0.7	0	0	0	0	0	0	0
4	0.9	0	0	1	0.9	0	0	0	0	0	0	0	0
5	0.7	0.7	0	0.9	1	0	0	0	0	0	0	0	0
6	0	0	0.7	0	0	1	0.9	0	0	0	0	0	0
7	0	0	0	0	0	0.9	1	0	0	0.9	0	0	0
8	0	0	0	0	0	0	0	1	0.9	0	0.9	0.7	0
9	0	0	0	0	0	0	0	0.9	1	0	0.7	0.9	0
10	0	0	0	0	0	0	0.9	0	0	1	0	0	0.9
11	0	0	0	0	0	0	0	0.9	0.7	0	1	0.9	0
12	0	0	0	0	0	0	0	0.7	0.9	0	0.9	1	0
13	0	0	0	0	0	0	0	0	0	0.9	0	0	1

Şekil 4.3: İlgilik Matrisi Örneği

W matrisinin elemanlarının boş hücreler arasındaki benzerliği göstermesinden dolayı bu değerlerin 0 ile 1 arasında olması uygun olacaktır. Bu durumda 1 tam benzerliği, 0 ise hiç benzer olmadığı durumu belirtmektedir. Dolayısıyla W matrisinin elemanlarına karar verilirken ilk önce her hücre kendisi ile tam benzer olduğu için kendisine karşılık gelen indekse 1 değeri atanmaktadır. Daha sonra, sekiz komşuluğunda boş hücre olup olmadığı kontrol edilmektedir. Eğer dikey veya yatay komşularında boş hücre mevcutsa, komşusu olan hücrenin indeksine 0.9 değeri atanmaktadır. Öte yandan, köşegenlerindeki komşularda boş hücre mevcutsa, o komşulara da 0.7 değeri verilmektedir. Son olarak, boş hücreler birbirlerinin sekiz komşuluğunda değilse ilgili elemanlara 0 değeri atanmaktadır.

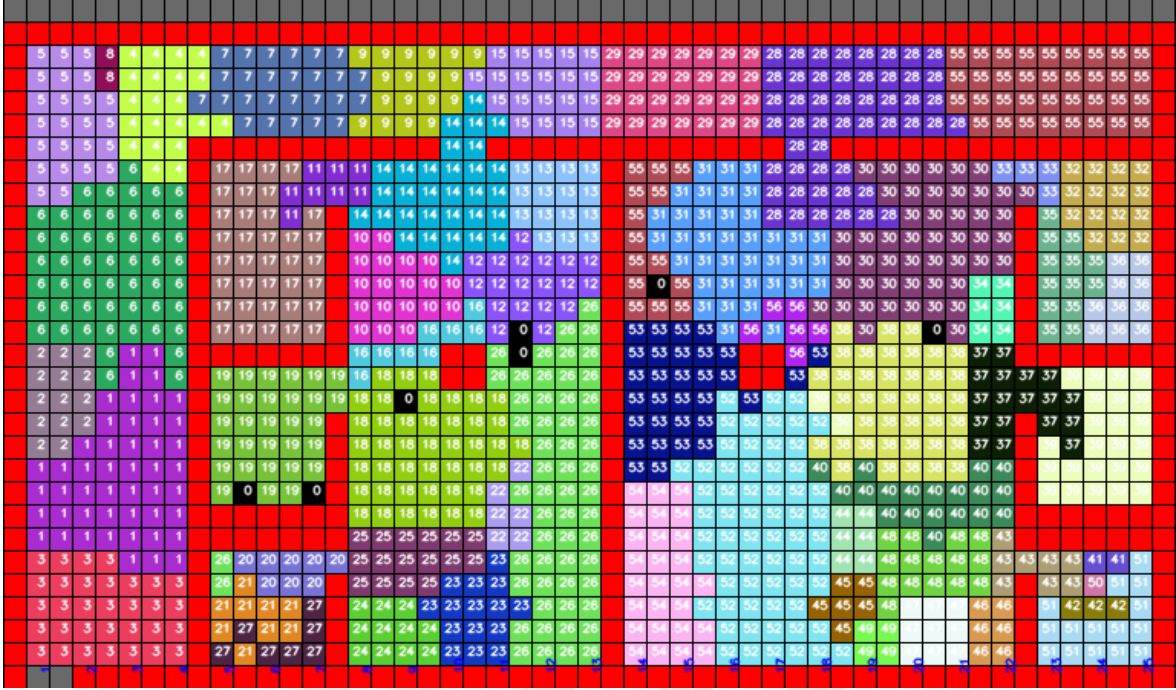
W matrisi elde edildikten sonra, D köşegen matrisi oluşturulmaktadır. Bu matristeki köşegen değerleri W matrisinde o satırda bulunan elemanların toplamı olarak hesaplanmaktadır. Denklem 4.5'te tanımlanan genelleştirilmiş özdeğer problemi, oluşturulan D ve W matrisleri kullanılarak çözülmüştür. Elde edilen özvektörler ilgili oldukları özdeğerlere göre küçükten büyüğe sıralanmıştır.

Bu aşamada, sıralanmış özvektörler kullanılarak boş hücreler kümelenecektir. Ancak bu hücrelerin kaç küme oluşturmasının uygun olacağı sorusu ortaya çıkmaktadır. Eğer az sayıdaki boş hücre bir küme oluşturursa ortam çok fazla sayıda kümeyle ve dolayısıyla düğümle ifade edilecektir (Şekil 4.4). Bu durumda topolojik haritanın sağladığı zaman ve hafıza karmaşıklığı avantajı azalacaktır.



Şekil 4.4: Çok fazla sayıda küme örneği

Diğer yandan eğer çok sayıdaki boş hücre bir küme oluşturursa ortam çok az sayıda kümeyle ve dolayısıyla düğümle ifade edilecektir (Şekil 4.5). Bu durumda ise ortamın bazı bölgeleri topolojik haritada yer bulamayabilir. Sonuç olarak, kümelerin sahip olacakları boş hücre sayısının çok fazla ya da çok az olmaması uygun olacaktır. Bu aşamada, önceden belirlenen bir sayıda boş hücrenin, bir küme oluşturacak şekilde birleştirilmesi



Şekil 4.5: Çok az sayıda küme örneği

amaçlanmaktadır. Önceden belirlenen bu sayı Beklenen Küme Sayısı (BeKS) olarak adlandırılmıştır. Bu durumda, küme sayısı (KS) aşağıdaki formül ile belirlenmektedir:

$$KS = \frac{BosHucresayisi}{BeKS} \quad (4.7)$$

Küme sayısı belirlendikten sonra boş hücreleri kümelerine ayırmak için özvektörlere çeşitli yöntemler uygulanabilmektedir. İdeal durumda ikinci en küçük özdeğere karşılık gelen özvektörde sadece iki farklı değer bulunması gerekmektedir. Dolayısıyla, bu değerlerin tespit edilip, hangi boş hücrenin hangi değere karşılık geldiği bulunarak kümeleme yapılabilir. Ancak, özvektörler her zaman sadece iki değer içeren bir yapıda olmayabilir. Bu durumda, özvektörleri kümelemek için bir ayrılma noktasına ihtiyaç duyulmaktadır. Bu ayrılma noktası 0 seçilip işarete göre kümeleme yapılacağı gibi ayrılma noktası özvektördeki elemanların ortalaması da seçilebilir. Aynı özvektör kullanılarak tekrarlamalı şekilde istenilen sayıda küme oluşturulabilir ya da üçüncü en küçük özdeğere karşılık gelen özvektör kullanılarak da kümelemeye devam edilebilmektedir (Shi ve Malik, 2000). Şekil 4.2'de verilen örnek metrik harita için özvektörler ilgili oldukları özdeğerlere göre küçükten büyüğe sıralanmış ve Şekil 4.6'da gösterilmiştir. Şekilde görüldüğü gibi ikinci en küçük özdeğere karşılık gelen özvektör sadece iki değere sahiptir ve boş hücreler mor ve yeşil olmak üzere kümelerine ayrılabilir. Bu noktadan sonra mor kümedeki hücreler 3. sütunda pozitif ve negatif değerlere göre lacivert ve pembe kümelerine bölünebilir. Son

olarak, lacivert küme benzer şekilde 4. sütunda pozitif ve negatif değerlere göre sarı ve mavi kümelere bölünebilir. Şekilde 4. sütun dikkatli incelendiğinde farklı renkte dört küme oluşturulduğu görülmektedir.

	1	2	3	4
1	0.32	0.00	0.39	-0.27
2	0.32	0.00	0.17	0.37
3	0.32	0.00	0.01	0.54
4	0.33	0.00	0.40	-0.27
5	0.36	0.00	0.38	-0.11
6	0.33	0.00	-0.19	0.39
7	0.35	0.00	-0.34	0.08
8	0.00	0.50	0.00	0.00
9	0.00	0.50	0.00	0.00
10	0.36	0.00	-0.44	-0.30
11	0.00	0.50	0.00	0.00
12	0.00	0.50	0.00	0.00
13	0.30	0.00	-0.40	-0.41

Şekil 4.6: Örnek metrik harita için özvektörler

Şekil 4.6'da gösterilen özvektörler farklı ayrılma noktaları belirlenerek de kümelere ayrılabilir. Bununla birlikte, k-ortalama gibi kümeleme yöntemleri de özvektörler üzerinde uygulanabilir. K-ortalama kümeleme yönteminde, ilk aşamada küme merkezleri rastgele belirlenmekte ve bu da her çalıştırmada farklı kümeleme sonuçlarına yol açmakta ve kümelemenin verimliliğini etkilemektedir. Bu problemi aşmak için, k-ortalamanın başlangıcında küme merkezleri verideki birbirine en uzak örnekler olarak seçilmiştir. Algoritma 8 en uzak örneklerin seçilme işlemini göstermektedir.

K-ortalama yöntemi ile oluşturulan küme merkezleri düğüm olarak kabul edilmektedir. Bu aşamada, elimizde birbiriyle herhangi bir ilişkisi bulunmayan düğümler bulunmaktadır. Bu düğümlerin bağlanabilmesi için, en küçük kapsayan ağaç algoritması kullanılmaktadır. Ancak, en küçük kapsayan ağaç algoritması düğümler arasındaki ayrıtların ağırlığını olabildiğince küçük yapmaya çalıştığından, bazı ayrıtlar dolu hücrelerden geçebilmektedir. Bunu engellemek için DDA algoritması kullanılmaktadır; eğer iki düğüm arasındaki ayrıt dolu bir hücreden geçiyorsa, ayrıtın ağırlığına çok büyük bir sayı atanarak en küçük kapsayan ağaç algoritması tekrar çalıştırılmaktadır. Bunun sonucu olarak çarpışma içermeyen bir topolojik haritanın oluşturulması sağlanmaktadır.

Algoritma 8: En uzak örneklerle k-ortalama yönteminin ilk değerlerinin belirlenmesi

```

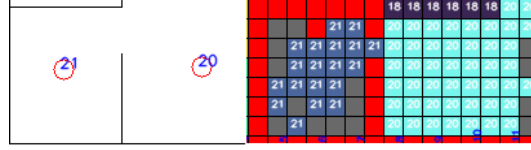
Girdi:  $l \times m$  boyutunda veri vektörü  $A$ , küme sayısı  $k$ 
Çıktı:  $k \times 1$  boyutunda en uzak örnekler vektörü  $I$ 
// Örnekler arasındaki Öklid uzaklığını hesapla ve  $Dist$  matrisine at
for  $a \leftarrow 1$  to  $l$  do
  | for  $b \leftarrow 1$  to  $l$  do
  | |  $Dist(a, b) \leftarrow a$  ve  $b$  örnekleri arasındaki uzaklık
  | end
end
 $I \leftarrow Dist$  matrisi içindeki en büyük değerin indekslerini bul
if  $k > 2$  then
  | for  $c \leftarrow 3$  to  $k$  do
  | |  $Dist \leftarrow 0$ 
  | | for  $a \leftarrow 1$  to  $l - k$  do
  | | | for  $b \leftarrow 1$  to  $c$  do
  | | | |  $Dist(a, b) \leftarrow a$  ve  $b$  örnekleri arasındaki uzaklık
  | | | end
  | | end
  | |  $Dist \leftarrow Dist$  matrisinin sütunlarını çarp
  | |  $I \leftarrow$  matrisi içindeki en büyük değerin indeksini bul
  | end
end

```

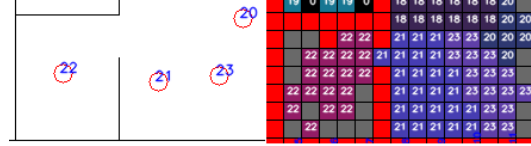
Bu aşamada, bazı bağlanmamış düğümlerin olduğu gözlenmiştir. Bu durumun oluşmasının sebebi, iki düğüm arasında en küçük kapsayan ağaç algoritmasının çarpışma içermeyen bir ayrıt oluşturamamasıdır (Şekil 4.7(a)). Bu durumu önlemek için de bazı yeni düğümlerin oluşturulması gerekmektedir. Bu yeni düğümler KS değerini ve özvektör sayısını uyarlamalı olarak değiştirerek oluşturulmaktadır (Şekil 4.7(b)). Bu süreç, bütün düğümlerin birbirleri ile, çarpışmaya neden olmayacak şekilde bağlanana kadar tekrarlanmaktadır (Şekil 4.7(c)).

Metrik harita oluşturmaktan çarpışmasız topolojik harita oluşturmaya kadar olan bütün işlemler belirli anlarda elde edilen yeni hücreler için tekrarlanmaktadır (Algoritma 9). Dolayısıyla, topolojik harita, robot yeni boş hücreler tespit ettikçe çevrimiçi bir şekilde büyümektedir. Böylece, topolojik harita oluşturmanın zaman karmaşıklığı azalmaktadır. Bu yöntemin önemli avantajlarından ilki çevrimiçi çalışmasıdır.

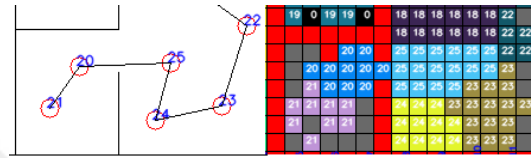
Bu alt bölümde anlatılan topolojik harita oluşturma yönteminin ikinci avantajı ise bir kümede bulunacak hücre sayısına uyarlamalı olarak karar verilmesidir. Bu yöntemin temel motivasyonu ortamı mümkün olduğunca az sayıda düğüm ile temsil ederken aynı zamanda ortamdaki her bölgeye de ulaşmaktır. Geleneksel spektral kümeleme



(a) Odanın duvarı nedeniyle bağlanamayan düğümler



(b) Odaya ulaşmak için yeni düğümlerin üretilmesi



(c) Odaya ulaşım

Şekil 4.7: Uyarlamalı düğüm oluşturma örneği

Algoritma 9: Çarpışmasız Topolojik Harita Üretme Algoritması

Girdi: Lazer Mesafe Taraması

Çıktı: Elde edilen düğümler

while bilinmeyen hücre mevcut **do**

 Metrik haritayı oluştur

repeat

 Boş hücreleri bul

W ve D matrislerini oluştur

do

 Denklem 4.5'teki genelleştirilmiş özdeğer problemini çöz

 K-ortalama ile kümelendir

 Küme merkezlerini kullanarak düğümlere karar ver

 En küçük kapsayan ağaç ve DDA algoritmalarını kullanarak çarpışmasız topolojik haritayı oluştur

if bağlanamayan düğümler mevcut **then**

 | KS ve özvektörü sayısını güncelle

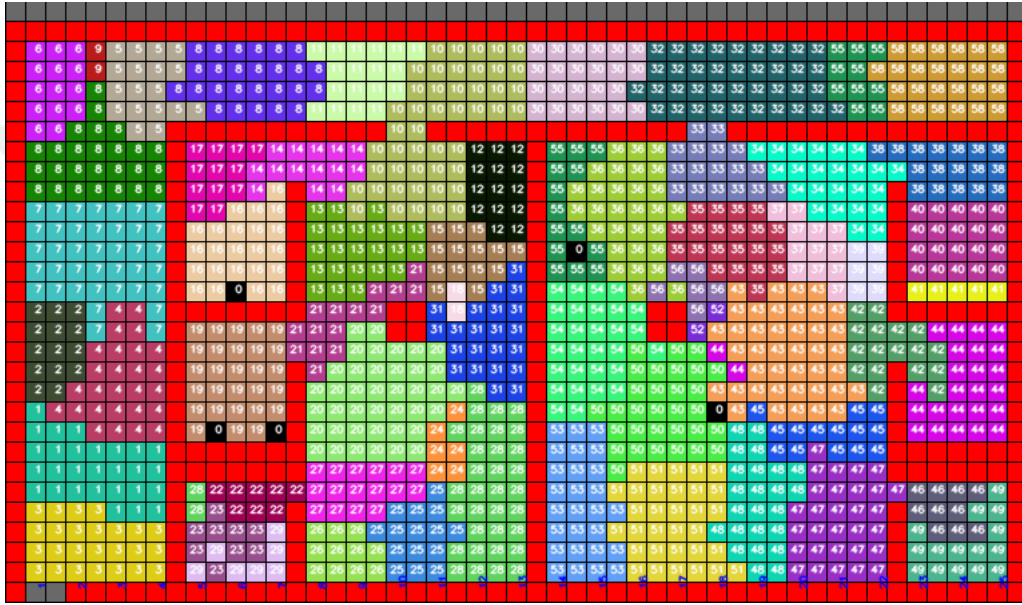
end

while bağlanamayan düğümler mevcut;

until her belirli zaman aralığında;

end

yöntemlerinde oluşturulacak kümelerde bulunacak olan hücre sayısı sabittir. Dolayısıyla bu sayının büyük olması bazı bölgelere ulaşılmamasına, küçük olması ise topolojik haritanın zaman ve hafıza karmaşıklığı avantajının azalmasına yol açmaktadır. Uyarlamalı olarak bir kümede bulunacak hücre sayısının değişmesi koridor ve büyük odalarda daha seyrek düğümlerin oluşmasına aracı olurken, küçük oda ve kapı gibi yerlerde düğümleri sıklaştırarak topolojik haritanın bağlanabilirliğini (connectivity) bozmamaktadır. Uyarlamalı yöntem ile elde edilen kümeleme sonuçları ve topolojik harita Şekil 4.8 ve Şekil 4.9'da sırasıyla verilmektedir.

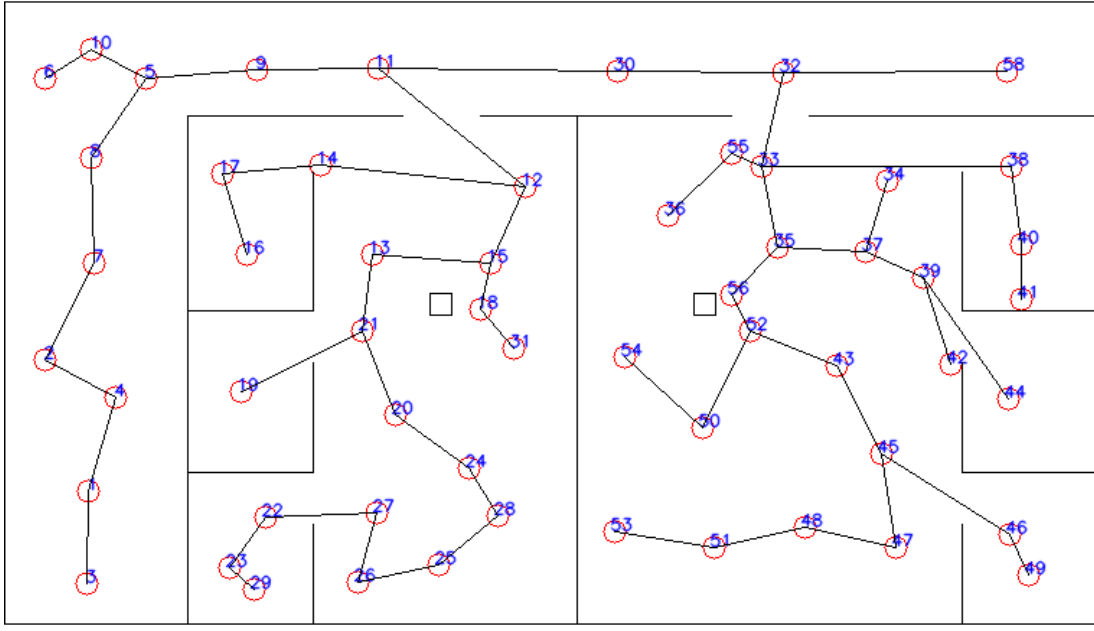


Şekil 4.8: Uyarlamalı yöntem ile elde edilen kümeleme sonuçları

4.2.3 Keşifte topolojik haritanın kullanılması

Keşifte topolojik haritanın kullanılabilmesi için sistem mimarisinde bulunan erkinlerde bazı düzenlemeler yapılmıştır. İlk olarak, ızgara erkini metrik ızgarayı oluşturmak, güncellemek, arayüzde göstermek ve istenildiğinde diğer erkinlerle paylaşmak ile sorumludur. Bunlara ek olarak, aynı sorumluluklar topolojik harita için de ızgara erkine yüklenmiştir.

Tüketici erkin, belirlenen görevler için maliyeti hesaplamaktadır. Metrik harita kullanıldığında görevlerin maliyeti robotun bulunduğu konumdan hedefine giderken üzerinden geçeceği hücre sayısı göz önüne alınarak hesaplanmaktadır. Dolayısıyla maliyetin birimi hücre sayısı olmaktadır. Bununla birlikte, topolojik harita ile keşif yapıldığında robot hedefine gitmek için bulunduğu konum ile hedefi arasındaki düğümleri kullanmakta ve maliyetin birimi mesafe olmaktadır.

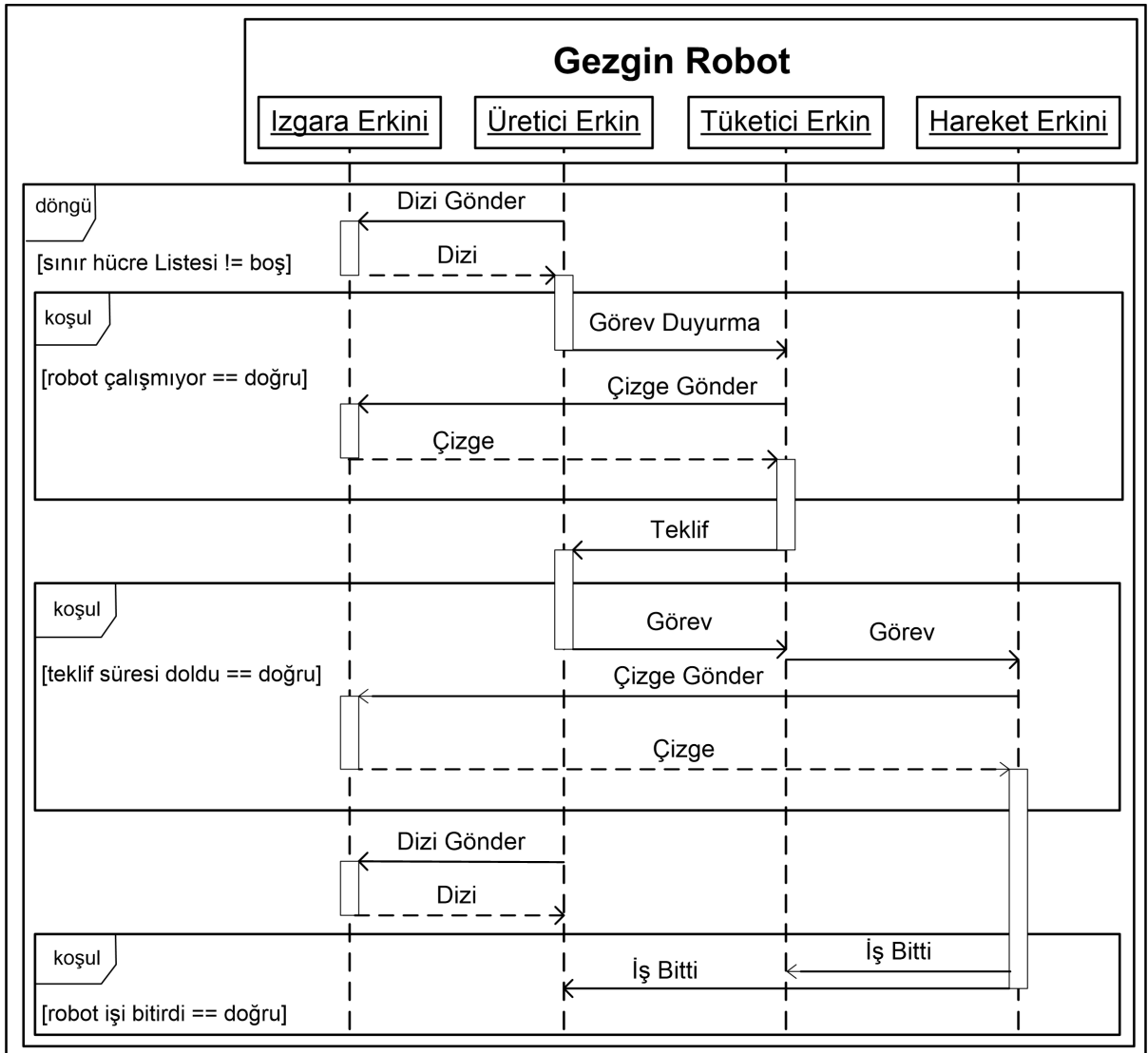


Şekil 4.9: Uyarlamalı yöntem ile elde edilen topolojik harita

Üretici erkin belirli aralıklarla yapılması gereken görevleri belirlemekte, bu görevleri fiyatlandırmakta ve robot için en faydalı görevi seçmektedir. Görevleri belirleme ve en faydalı görevi seçme süreçlerinde topolojik harita ile birlikte gelen bir değişiklik mevcut değildir. Ancak, topolojik haritanın kullanılmasıyla birlikte görev fiyatlarının biriminin değişmesi gerekmektedir. Bunun sebebi, metrik harita ile keşif yapıldığında görevleri fiyatlandırırken sınır kenar hücre sayısı kullanılmakta ve fiyatın birimi hücre sayısı olmaktadır. Benzer şekilde görevlerin maliyetleri de hücre sayısı biriminde olduğu için aynı birimdeki maliyet ve fiyat kullanılarak fayda hesaplanabilmektedir. Bununla birlikte, topolojik haritanın kullanılmasıyla birlikte görevlerin maliyetlerinin birimi mesafe olmuş ve fiyatın da mesafe cinsinden hesaplanması gerekmiştir. Bu noktada, görevi tanımlayan sınır hücre alanının çevresi fiyat olarak belirlenmiştir.

Hareket erkini iki temel görevden sorumludur. Bunların ilki, robotun bulunduğu noktadan belirlenen hedefine gidebilmesi için yol planının hazırlanmasıdır. Metrik harita kullanılarak keşif yapıldığında yol planı A* ile elde edilmektedir. Topolojik harita kullanıldığında ise robot kendisine ve hedefe en yakın düğümlere karar vermektedir. Ancak, robot ile robota en yakın ve hedef ile hedefe en yakın düğümler arasında engel olmaması gerekmektedir. Bu durum göz önüne alınarak başlangıç ve bitiş düğümleri belirlendikten sonra, Dijkstra'nın (Dijkstra, 1959) en kısa yol algoritması kullanılarak yol planı yapılmıştır. Yol planı oluştuktan sonra robotun engellerden sakınarak güvenli bir şekilde hedefine varabilmesi için VAH kullanılmıştır.

Şekil 4.10'da erkinler arası iletişimi gösteren sıralama diyagramı keşifte topolojik harita kullanıldığı sistem için verilmiştir. Şekilde görüldüğü gibi, üretici erkin fiyatları belirlemek için diziyi isterken tüketici erkini maliyeti hesaplamak için çizgeyi istemektedir. Benzer şekilde hareket erkini de yol planını yapabilmek için çizgeyi istemektedir. Izgara erkini hareket erkinin iş bitti mesajını aldığı anda yeni boş hücreleri kullanarak yeni düğümleri oluşturmakta ve topolojik haritayı güncellemektedir. Böylece, tüketici erkin yeni görevlere teklif vermek için ızgara erkinin topolojik haritayı istediğinde güncel harita gönderilebilmektedir.



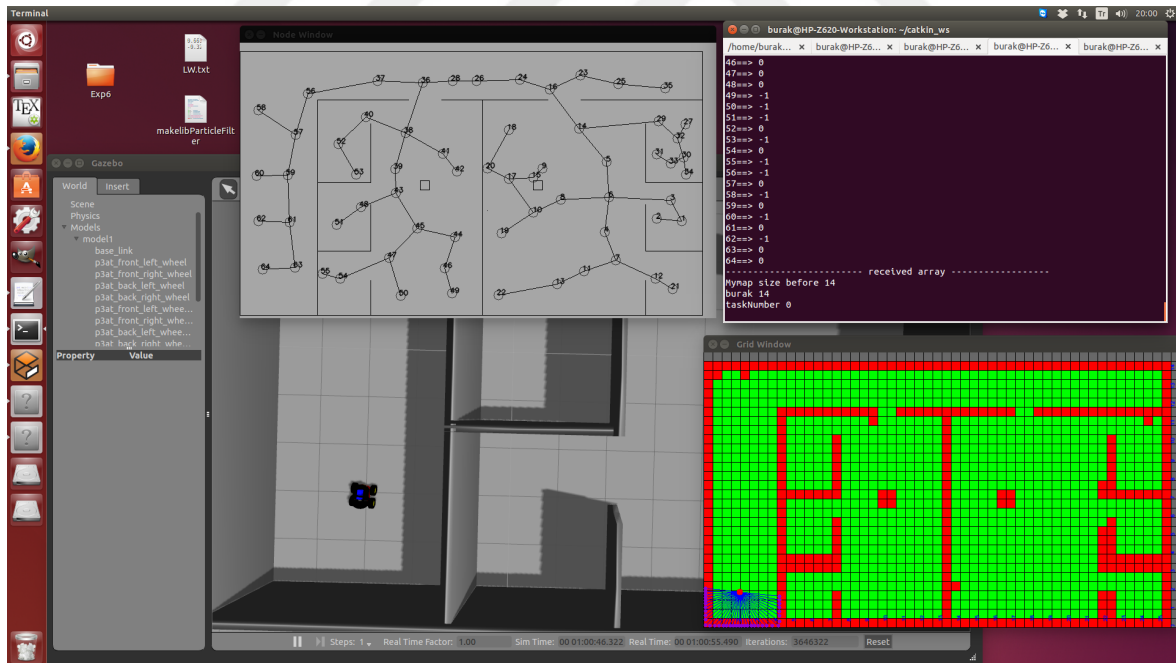
Şekil 4.10: Topolojik Harita ile Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)

4.3 Sonular

Bu alt b3l3mde ilk olarak deneysel kurulum aıklanacaktır. Deneysel ortamı, erkinleri alıřtırırken kullanılan parametreler gibi detaylar bu kısımda yer alacaktır. Daha sonra A* ve Dijkstra yol planı ile birlikte farklı fiyatlandırma kriterleri iin sayısal ve g3rsel sonular verilecek ve her bir fiyatlandırma kriteri iin yol planlarının sonular 3zerindeki etkileri irdelenecektir.

4.3.1 Deneysel kurulum

3nerilen y3ntem, Gazebo benzetim ortamında modellenen Eskiřehir Osmangazi 3niversitesi Elektrik-Elektronik Laboratuvar binasında test edilmiřtir. Ortamın boyutları 26x15 metredir ve 52x30'luk metrik ızgara ile ifade edilmiřtir. Izgaranın g3ncel durumu saniyede bir ekrana getirilmektedir. Bununla birlikte, topolojik harita, robot g3revini bitirdiėinde g3ncellenmekte ve openCV ile g3rselleřtirilmektedir. Őekil 4.11'de robot topolojik haritayı kullanarak keřfi tamamladıktan sonra elde edilen metrik ve topolojik harita 3rneėi verilmiřtir.



Őekil 4.11: Test ortamı (Gazebo), Metrik ve Topolojik Harita 3rneėi

Topolojik haritanın oluřturulması esnasında *BeKS* parametresi 25 boř h3cre olarak seilmiřtir. Ayrıca, 3zvekt3rler belirlendikten sonra en k33k 3zdeėere karřılık gelen 3zvekt3r kullanılmamaktadır. En k33k ikinci ve 33nc3 3zdeėere karřılık gelen 3zvekt3rler kullanılarak d3ė3mler oluřturulmaya alıřılmakta; engeller y3z3nden d3ė3mler

bir ağaç oluşturamazsa *BeKS* küçültülerek ve/veya kullanılan özvektör sayısı artırılarak uyarlamalı olarak tüm düğümler bağlanana kadar bu süreç devam etmektedir.

Uygulamalar gerçekleştirilirken Dijkstra yol planı ile birlikte üç farklı fiyatlandırma kriteri (en yakın sınır hücre, en çok kenar hücrelerine sahip sınır hücre ve market sınır hücre) kullanılmıştır. Robot, hem metrik harita hem de topolojik keşif için aynı olan üç farklı yerden görevine başlamakta ve belirlenen fiyatlandırma kriterine göre bütün ortamın keşfi tamamlanana kadar görevini sürdürmektedir.

4.3.2 Deneysel sonuçlar

Çizelge 4.1'de hem A* hem de Dijkstra yol planları ile üç farklı fiyatlandırma kriteri için sayısal sonuçlar verilmiştir. Sonuçlar fiyatlandırma açısından incelendiğinde, metrik harita ve A* kullanılarak yapılan keşifte olduğu gibi en çok kenar hücrelerine sahip sınır hücre kriteri kullanıldığında; robotun üç farklı başlangıç durumunda da, market ve en yakın sınır hücre kriterlerine göre daha uzun mesafeler katettiği ve keşfin daha uzun süre aldığı açıkça görülmektedir. Bununla birlikte, market ve en yakın sınır hücre kriterleri farklı hedefler seçmelerine rağmen hem mesafe hem de süre açısından benzer sonuçları üretmektedir.

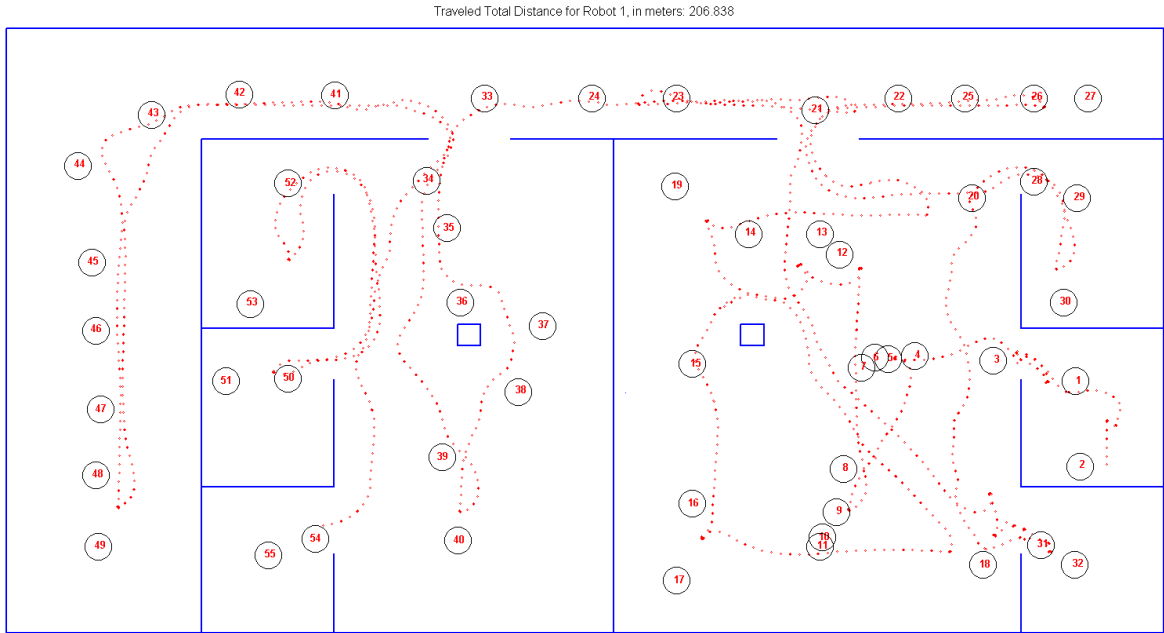
Çizelge 4.1: Tek Robot A* ve Dijkstra Sonuçları

Başlangıç Konumu	Fiyatlandırma Kriteri	A*		Dijkstra	
		Mesafe (m)	Süre (Dk)	Mesafe (m)	Süre (Dk)
2.5, 0.3, 1.57	En çok kenar hücresi	249.0206	92	225.1777	72
	En yakın	137.7664	71	188.3614	67
	Market	129.4584	66	201.4353	68
9, 7, 1.57	En çok kenar hücresi	193.9504	80	276.1921	82
	En yakın	165.8054	77	220.8951	71
	Market	171.0292	78	197.6729	66
24, 4, 1.57	En çok kenar hücresi	206.8380	83	286.2189	80
	En yakın	156.6460	73	231.8223	72
	Market	161.0887	71	241.0084	74

Dijkstra yol planı ile bütün fiyatlandırma kriterleri için robotun katettiği mesafe, A* yol planına göre daha fazladır. Bununla birlikte, her iki yol planı için de robotun keşfi tamamlama süresi birbirine çok yakındır. Bu durumu açıklamak amacıyla üç fiyatlandırma kriteri için A* ve Dijkstra ile elde edilen görsel sonuçlar verilecektir. Her fiyatlandırma kriterinde robot her iki yol planı için de aynı noktadan keşfe başlamaktadır. Görsel

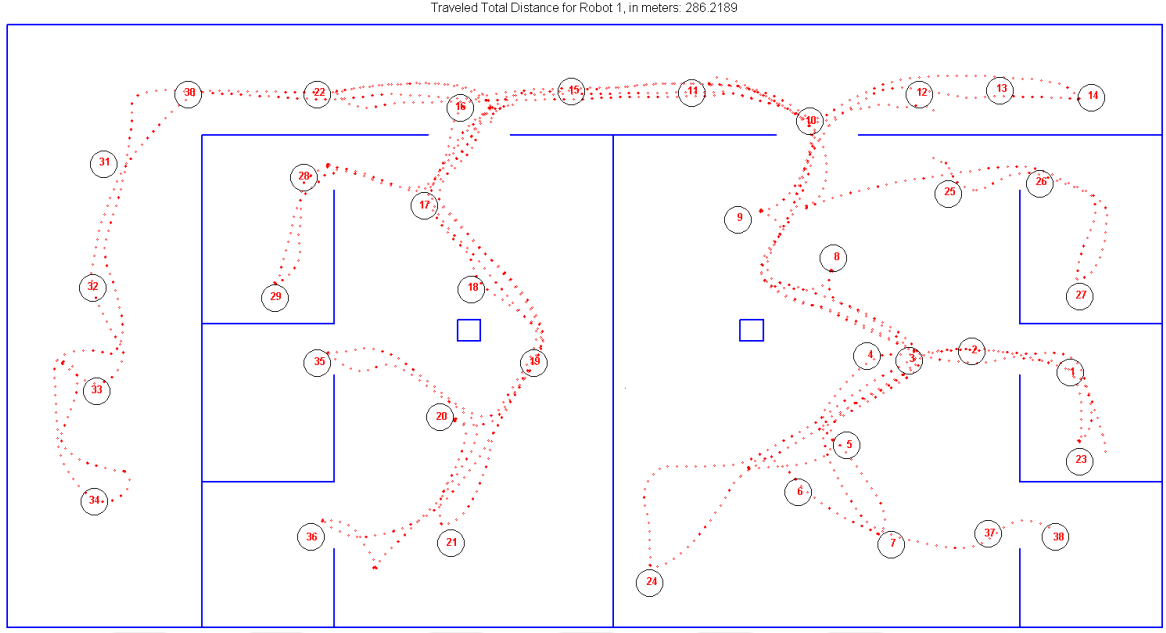
sonuçlarda siyah çemberlerin içinde kırmızı rakamlarla belirtilen konumlar sırasıyla seçilen hedefleri göstermektedir. Bu çemberler arasında kalan kırmızı noktalar ise robotun izlediği yolları ifade etmektedir.

Şekil 4.12 ve 4.13'te sırasıyla A* ve Dijkstra yol planları için robotun 24, 4, 1.57 konumundan başladığı durumdaki en çok kenar hüresine sahip sınır hücre kriterinin yaptığı hedef seçimleri gösterilmiştir. En çok kenar hüresine sahip sınır hücre kriterinde hedef seçiminde sadece kenar hücre sayısı göz önüne alınıp hedefe gitme maliyeti hesaba katılmadığı için her iki yol planı ile benzer sonuçların elde edilmesinin beklenmesi doğaldır. Bununla birlikte, verilen sonuçlarda farklı hedef seçimlerinin yapıldığı gözlemlenmektedir. Robotun farklı yol planlarını izlemesi, doluluk ızgarasının farklı güncellenmesine neden olmaktadır. Dolayısıyla, farklı doluluk ızgaraları da farklı sonuçlara neden olmaktadır.



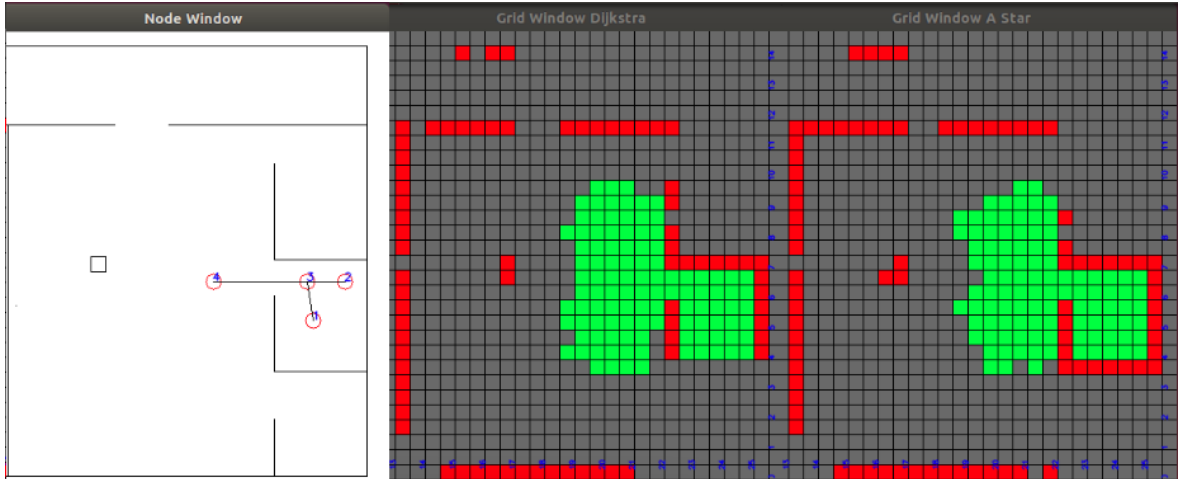
Şekil 4.12: A* en çok kenar hüresine sahip sınır hücre kriteri

Şekiller incelendiğinde, Dijkstra yol planı kullanıldığında robotun A* yol planı kullanıldığı duruma göre daha az sayıda hedef belirlediği gözlemlenebilir. Dijkstra yol planı kullanıldığında, robot kendine en yakın düğümü belirlemekte ve o düğümüne gitmektedir. Daha sonra hedefe en yakın düğümüne kadar belirlenen yolu takip etmekte ve bu düğümüne ulaştığında da hedefine yönelmektedir. Bu durumda, hedefe en yakın düğüm hedef ile robot arasındaysa önce düğümüne sonra hedefe ulaşılır. Bununla birlikte, eğer hedef robot ile hedefe en yakın düğüm arasındaysa, robot asıl hedefini geçmekte ve daha sonra geri dönmektedir. Ortam hakkında yeterli bilgi elde edildiğinde bu sadece robotun fazladan mesafe



Şekil 4.13: Dijkstra en çok kenar hücreesine sahip sınır hücre kriteri

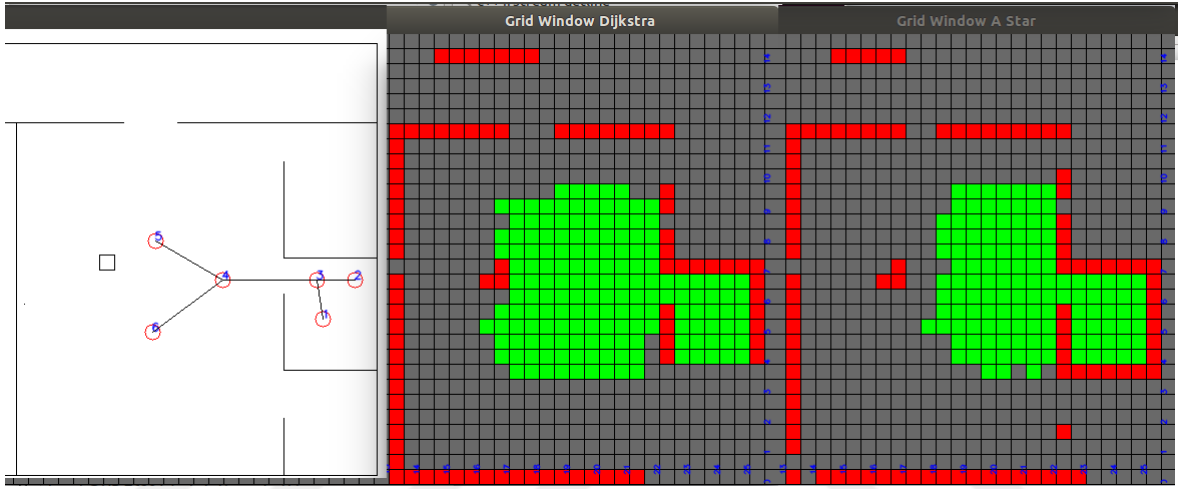
katetmesine ve zaman harcamasına neden olacaktır. Ancak, robotun keşfe yeni başladığı ve ortam hakkında bilgisinin az olduğu anlarda asıl hedefinden ileri gitmesi ona fazladan bilgi kazandıracığı için bu durum avantaj olarak görülebilir. Şekil 4.14'te bu duruma bir örnek verilmektedir.



Şekil 4.14: A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (1)

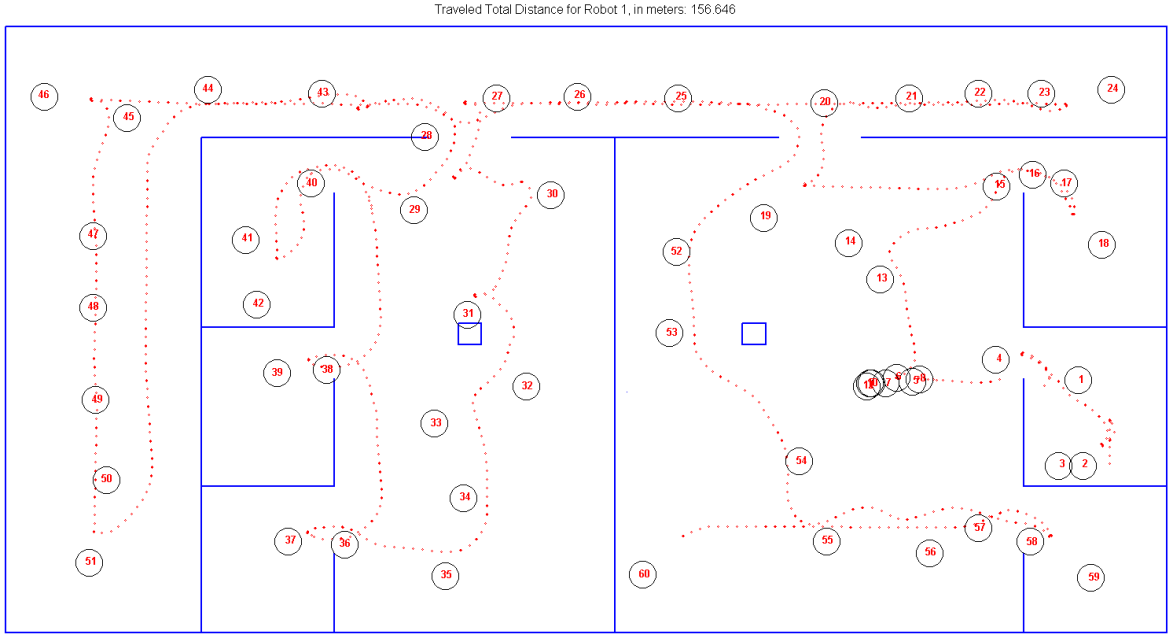
Bu örnekte, robotun A* yol planı ile 3. hedefine (Şekil 4.12) ve Dijkstra yol planı kullanarak 2. hedefine (Şekil 4.13) ulaştığı andaki doluluk ızgaraları ve topolojik harita verilmiştir. Bu anlarda, robot hemen hemen aynı konumdadır ve doluluk ızgaraları birbirine

çok benzemektedir. Dolayısıyla, robotun A* yol planındaki 4. hedefi ile Dijkstra yol planındaki 3. hedefi hemen hemen aynıdır. Bununla birlikte, Dijkstra yol planı kullanıldığında robot önce 4. düğüme gitmekte daha sonra hedefine yönelmektedir. A* ise direk olarak hedefine gitmektedir. Bu iki yol planı ile robotlar hedeflerine vardıklarında elde edilen doluluk ızgaraları Şekil 4.15'te verilmiştir. Şekilde görüldüğü üzere Dijkstra yol planı kullanıldığında elde edilen doluluk ızgarası A* yol planı kullanıldığı duruma göre çok daha fazla bilgi içermektedir. Dolayısıyla, aynı bilgiye sahip olabilmek için A* yol planı ile daha fazla hedefe gitmek gerekmektedir.

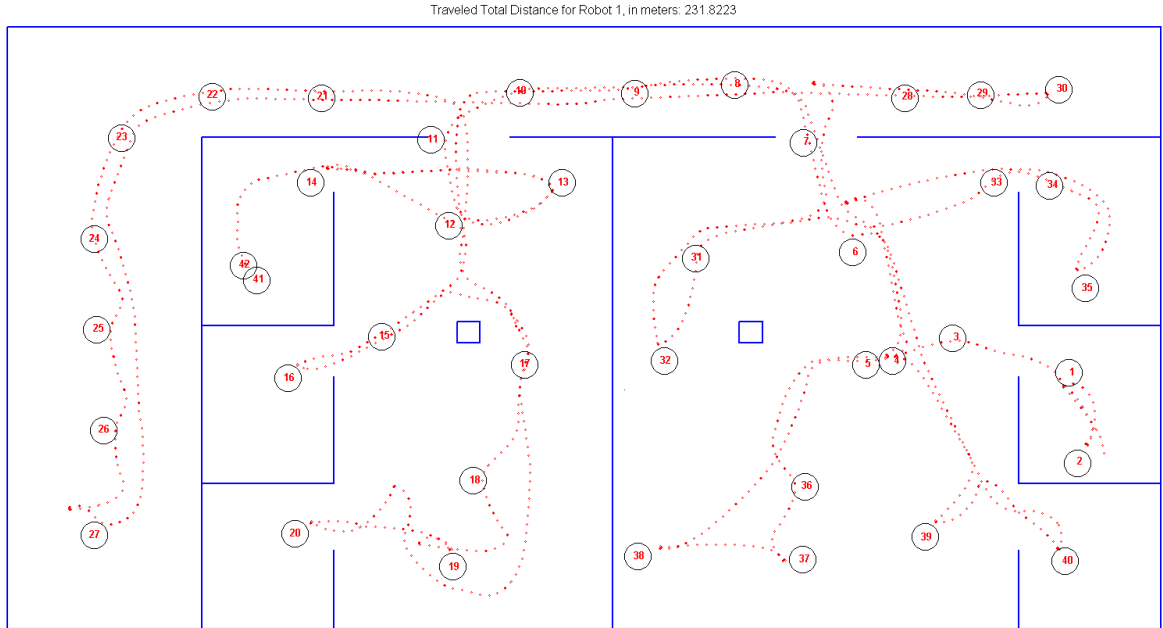


Şekil 4.15: A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (2)

En yakın sınır hücre kriteri kullanılarak fiyatlandırma yapıldığında robot o sınır hücresine gittiğinde ne kadar yeni bilgi kazanacağını önemsemeyen en yakındaki sınır hücresine hareket etmektedir. A* ve Dijkstra yol planları birbirlerinden farklı yöntemler ile robot ve hedef arasındaki mesafeyi hesapladıkları için en yakın hedefi belirlerken bu iki yöntemin farklı hedefleri seçmesi son derece doğaldır. Şekil 4.16 ve 4.17'de sırasıyla A* ve Dijkstra yol planları için robotun 24, 4, 1.57 konumundan başladığı durumdaki en yakın sınır hücre kriterinin yaptığı hedef seçimleri görünmektedir.



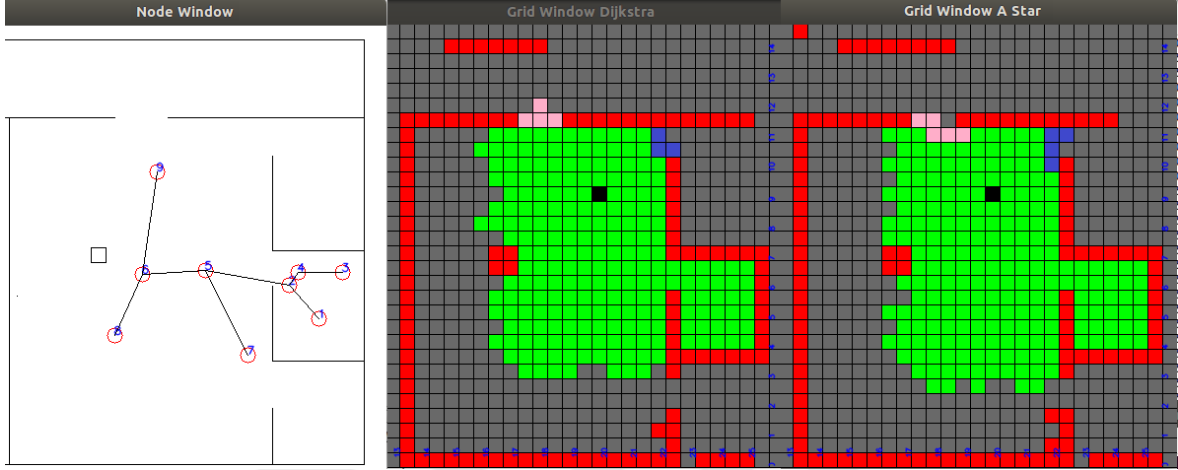
Şekil 4.16: A* en yakın sınır hücre kriteri



Şekil 4.17: Dijkstra en yakın sınır hücre kriteri

Robot A* yol planı ile 14. hedefine (Şekil 4.16) ve Dijkstra yol planı kullanarak 6. hedefine (Şekil 4.17) ulaştığında iki yol planı yöntemi için seçilen hedefler farklılaşmaya başlamıştır. Bu andaki doluluk ızgaraları ve topolojik harita Şekil 4.18'de verilmiştir. Bu anlarda, robot hemen hemen aynı konumdadır ve doluluk ızgaraları birbirine çok

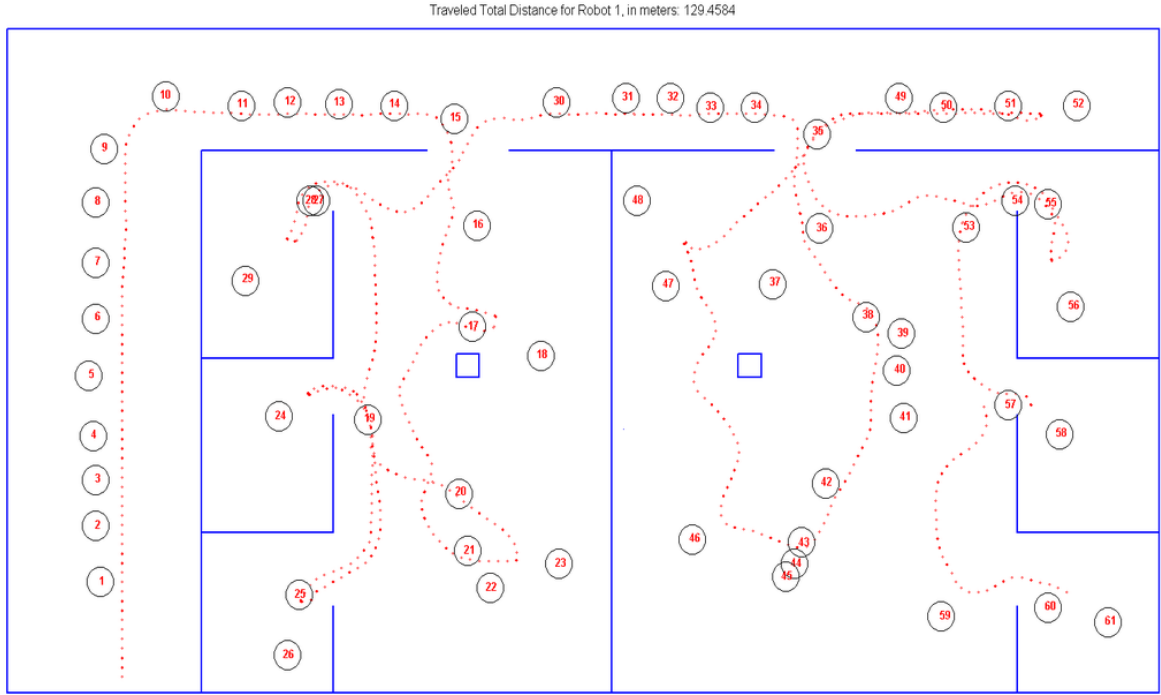
benzemektedir. Robot siyah, robota yakın olan hedefler ise pembe ve lacivert renklerle gösterilmiştir. A* yol planı kullanıldığında robot kendisine en yakın hedef olarak lacivert hedefi seçmektedir. Bununla birlikte, Dijkstra yol planı kullanıldığında robot önce kendisine en yakın düğüm olan 9. düğüme gidecektir. Bu noktada, lacivert hedef robota pembe hedeften daha uzak olduğu için robot pembe hedefi seçecektir.



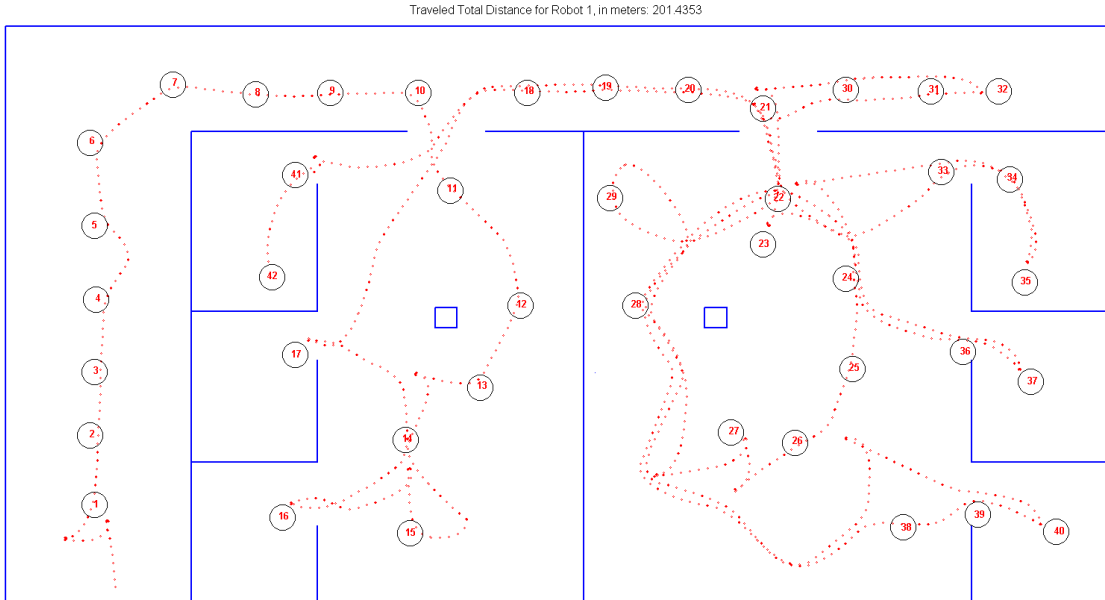
Şekil 4.18: A* ve Dijkstra yol planlarının hedef seçimi üzerindeki etkisi (3)

Market sınır hücre kriteri ise bir sonraki hedefi seçerken hem mesafeyi hem de o hedefe gidildiğinde kazanılacak olan yeni bilgileri hesaba katarak kararını vermektedir. Şekil 4.19 ve 4.20'de sırasıyla A* ve Dijkstra yol planları için robotun 2.5, 0.3, 1.57 konumundan başladığı durumdaki market sınır hücre kriterinin yaptığı hedef seçimleri görünmektedir. Şekil incelendiğinde robotun yukarıda bahsedilen iki kriteri birleştirerek çalıştığı gözlemlenmektedir.

Dijkstra yol planı kullanıldığında robotun A* yol planı kullanıldığı duruma göre daha az sayıda hedef belirlediği bütün farklı başlangıç noktaları ve fiyatlandırma kriterleri için gözlemlenmektedir. Bu sonucun iki sebebi bulunmaktadır. Birincisi, robot keşfe yeni başladığında ya da daha önce keşfedilmemiş bir alana girdiğinde eğer Dijkstra yol planı kullanılırsa ve hedef robot ile hedefe en yakın düğüm arasındaysa, robot asıl hedefini geçmekte ve daha sonra geri dönmektedir. Bu durum robotun asıl hedefinden ileri gitmesine ve ortam hakkında daha fazla bilgi kazanmasına neden olmaktadır. İkinci olarak, verilen görsel sonuçlar dikkatli incelendiğinde A* yol planı mesafeyi kısaltmak için genelde duvara yakın hücreleri tercih etmektedir. Dolayısıyla, robot yol planını izlerken daha az bilgi elde etmekte ve kısa aralıklarla tekrar tekrar hedef belirlemek zorunda kalmaktadır. Her hedef belirleme aşamasında robotun erkinleri arasında bilgi alışverişi olmakta ve hedefin belirlenmesi uzun bir süre almaktadır. Dolayısıyla, ortamın keşfini tamamlamak için daha



Şekil 4.19: A* market sınır hücre kriteri

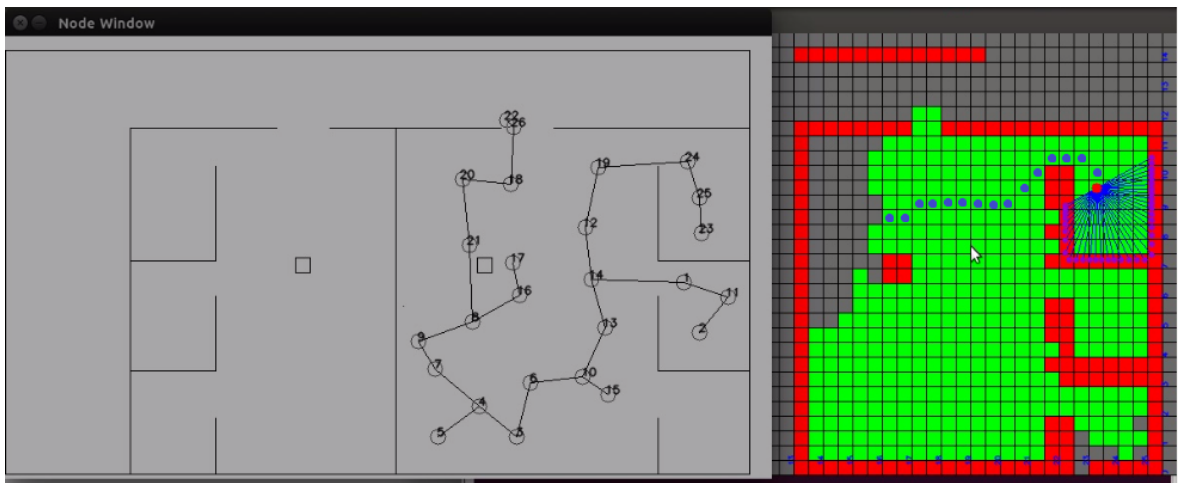


Şekil 4.20: Dijkstra market sınır hücre kriteri

çok hedefe ihtiyaç duyan A* yol planlama yöntemi ile keşfin Dijkstra yol planlama yöntemine göre daha uzun süre alması beklenmektedir. Bununla birlikte, sayısal sonuçlar incelendiğinde A* ve Dijkstra yol planları kullanıldığında robotun keşfi tamamlama sürelerinin hemen hemen aynı olduğu görülmektedir. Dolayısıyla, robot, Dijkstra yol

planını kullandığında zaman kaybetmektedir. Ayrıca Dijkstra yol planı ile robotun A* yol planına göre daha fazla mesafe katettiği görülmüştür. Bunun sebebi, topolojik harita oluşturulurken düğümlere karar verildikten sonra bu düğümlerin en küçük kapsayan ağaç yöntemi ile birbirlerine bağlanmasıdır. Örnek bir durum Şekil 4.21'de verilmektedir. Şekilde robot sağ üst küçük odada bulunmakta ve kırmızı bir daire ile gösterilmektedir. Bu noktada, robotun büyük odanın solunda bulunan bilinmeyen alana gitmesi beklenmektedir. Eğer A* yol planı kullanılırsa robot lacivert renkle gösterilen yol planını izleyecektir. Bununla birlikte, topolojik harita ve Dijkstra yol planı ile sırasıyla 24, 19, 12, 14, 13, 10, 6, 3, 4, 7, 9, 8, 21 numaralı düğümleri takip edecektir. Dolayısıyla, A* yol planına göre çok daha fazla mesafe katedecek ve zaman kaybedecektir. Öte yandan, ortamın belirtilen düğümleri kapsayan kısmı keşfedildiği için bu bölgeden yeni bilgi de elde edilemeyecektir. Bunlara ek olarak, robotun tekrar sağ alt küçük odaya gelmesi gerektiğinde benzer düğümleri kullanması gerekecektir. Bu durum Dijkstra yol planı ile bir noktadan birden fazla kere geçmesini açıklamaktadır. Sonuç olarak, A* yönteminde görüşmeler sırasında yaşanan kayıp Dijkstra yönteminde daha fazla yol katedilerek yaşanmakta ve her iki yol planı için de hemen hemen aynı keşif süreleri elde edilmektedir.

Gelecek çalışmalarda düğümler elde edildikten sonra topolojik haritanın oluşturulmasında en küçük kapsayan ağaç yöntemi yerine döngüler içeren ve ortamı daha iyi ifade eden yöntemler kullanılması planlanmaktadır. Böylece robotun sürekli aynı düğümleri kullanarak hedeflere gitmesinin önlenmesi ve katedeceği mesafe ve sürenin de azalması beklenmektedir. Bununla birlikte, yol planı yapılırken hem metrik harita hem de topolojik harita kullanılabilir. Bir düğümden daha önce geçilip geçilmediği belirlenerek ortamın o parçasının daha önceden keşfedilip keşfedilmediği anlaşılabilir ve bu anlarda yol planı A* ile yapılabilir.



Şekil 4.21: Dijkstra yol planlarının hedef seçimi üzerindeki etkisi

5. ÇOK ROBOTLU KEŞİF YÖNTEMİ

Kaleci (Kaleci, 2010), yaptığı çalışmada çok robotlu sistemlerin gerekliliği ve avantajları ile dezavantajlarını şu şekilde özetlemektedir: "Robotik alanındaki talepleri karşılamak için, 1990'lı yılların başlarına kadar genellikle tek robotlu sistemler kullanılmıştır. Bununla beraber, araştırmacılar robotların birçok zorlu alanda tek başlarına çalışmak yerine takım halinde çalışmalarının daha verimli olabileceğini düşünmüşlerdir. Bu amaçla, günümüzde birçok karmaşık problemde, görevlerin hızlı tamamlanması, daha gülbüz sistemlerin elde edilmesi, çözüm kalitesinin artırılması ve tek robotlu sistemler tarafından yapılamayan görevlerin yerine getirilmesi amacıyla çok robotlu sistemler, tek robotlu sistemlere tercih edilmektedir. Çok robotlu sistemlerin avantajlarından yararlanmak için, görevlere atanacak robot sayısının artırılması yeterli bir çözüm yolu değildir. Bununla birlikte, robotlar dinamik ortamlarda bulunmakta ve değişen görev taleplerine, sahip oldukları kısıtlı kaynak (zaman ve enerji) ile cevap vermeye çalışmaktadırlar. Bu noktada, robot takımlarının kendilerine verilen görevleri güvenilir ve verimli bir şekilde yerine getirebilmeleri için robotlar arası koordinasyonun çok iyi yapılması gerekmektedir. Çok robotlu sistemlerin koordinasyonu bugüne kadar yüzlerce çalışmanın yapıldığı, görev paylaşırma, haberleşme, robot takımının yapısı gibi konuları içinde barındıran önemli ve büyük bir problemdir. Tez çalışmasının kapsamı göz önüne alındığında haberleşmenin kısıtlı olduğu ortamlar çok robotlu keşfin verimliliğini artırmak amacıyla robotların görev paylaşımını nasıl yaptıkları, haberleşme menziline olduklarında sahip oldukları bilgileri nasıl birbirleri ile paylaştıkları ve birbirleri ile karşılaştıklarında birbirlerine çarpmadan yerel olarak nasıl koordine oldukları bu bölümde ele alınacaktır."

Bu bölümde, ilk olarak geçmiş çalışmalarda yer alan belli başlı koordinasyon ve keşif yaklaşımları, çok robotlu sınır hücre keşif yaklaşımları ve haberleşmenin kısıtlı olduğu durumlardaki keşif yaklaşımlarından öne çıkanlar incelenecektir. Daha sonra gerçekleşen yaklaşım alt bölümünde market tabanlı görev paylaşırma yaklaşımı kısaca anlatılacak; robotların haritaları nasıl birleştirdikleri ve çarpışma önleme protokolü detaylı olarak örneklerle verilecektir. Son olarak, deneysel kurulum açıklanacak; üç fiyatlandırma kriteri ile topolojik ve metrik harita kullanımının çok robotlu keşif üzerindeki olumlu ve olumsuz yönleri tartışılacaktır.

5.1 Geçmiş Çalışmalar

5.1.1 Çok robotlu sistemlerde koordinasyon ve keşif yaklaşımları

Singh ve Fujimura yaptıkları çalışmada içinde sabit engeller bulunan iki boyutlu bilinmeyen bir alanı doluluk kafesi ile ifade etmekte ve bu alanın haritasını çoktörel bir robot takımı yardımıyla doluluk kafesi formatında oluşturmayı amaçlamaktadırlar. Robot takımındaki her üye diğerlerinden büyüklüğü, hızı ve algılayıcı menzili gibi özellikleri ile ayrılmaktadır. Bilinmeyen bir ortamın haritasını çıkarmak için önerilen dağıtık yaklaşımda robotların konumlarını bildikleri ve birbirleri ile mükemmel bir şekilde haberleşebildikleri varsayımları yapılmaktadır. Başlangıçta robotlar ilk konumlarına en yakın hedefe doğru hareket planı hazırlamaktadırlar. Bu plan, diğer robotların mevcut konumlarını korudukları varsayımla çarpışmasız bir güzergahdan oluşmaktadır. Bu noktada, robot diğer robotlardan bağımsız olarak kendi hedefini keşfetmekte ve ortama ilgili yeni bilgileri kendi yerel haritasında güncellemektedir. Bu keşif sırasında robot bir hedef farkedip, büyüklüğünün o hedefe ulaşmasına engel olması gibi durumlar yüzünden hedefi keşfedemiyorsa bir tünel bulduğunu söylemekte ve diğer aday robotların, (tünel girebilecek büyüklükteki robotlar) robot çağrı kuyruğuna bu tüneli eklemektedir. Bir hedefe varıldığında robot yerel haritasını diğer robotlar ile paylaşmakta ve kendi çağrı kuyruğunu kontrol etmektedir. Eğer çağrı kuyruğu boş ise komşu hedefe geçerek keşfe devam etmektedir. Aksi takdirde, robot çağrı kuyruğundaki en yakın tüneli keşfetmek için hareket etmektedir. Bu çalışmada, robotlar harita oluşturma sürecinin büyük bir kısmında birbirlerinden bağımsız çalışmaktadırlar. Ancak bir robotun tek başına yapamayacağı bir görev olduğunda diğerlerinden yardım alınmaktadır (Singh ve Fujimura, 1993).

Rekleitis ve diğerleri önerdikleri yaklaşımla robotların ikili takımlar halinde geniş ve boş ortamları keşfetmelerini sağlamaktadırlar. Bu yaklaşımda, her robot, kameraya ve aynı zamanda özel bir desene sahip olmaktadır. Bir robot sabit kalırken diğer robot bu robotun görüş alanı içerisinde kalan bir başlangıç noktasından bir bitiş noktasına doğru ilerlemekte ve bu üç nokta arasında kalan boş alan keşfedilmektedir. Keşif stratejisi ortamın büyüklüğüne bağlı olarak değişmektedir. Eğer ortam küçükse az önce bahsedilen ortam üçgenlenmesi yöntemi kullanılır. Aksi takdirde, ortam yamuksal olarak parçalanmakta ve her parça için bir düğüm atanmaktadır. Robotlar düğümlerden oluşan bu çizge üzerinde tüm yamuksal alanları keşfederek ilerlemektedirler. Yamukların içindeki tüm alanı keşfetmek için robotlar rolleri değişerek hareket etmektedirler. Bu yaklaşımın ikiden fazla robot için genişletilmiş versiyonu da yayında yer almaktadır (Rekleitis vd. 2000).

Zlot ve diğerleri robotlar arasındaki koordinasyonu market tabanlı bir yaklaşımla sağlamaktadırlar. Bu yaklaşımda, robotlar sürekli olarak diğer robotlar ile müzakere ederek

mevcut planlarını iyileştirmekte ve hangi alanların keşfedilip edilmediği bilgilerini birbirleri ile paylaşmaktadırlar. Market yapısında, müzakerelerde kullanılacak olan teklif, görevin yapılması ile elde edilecek faydadan görevin maliyetinin çıkarılması ile bulunmaktadır. Maliyet hesabı için mesafe tabanlı bir ölçüt kullanılmıştır. Ortam doluluk kafesi ile gösterilmekte ve fayda bir hedef noktasının çevresindeki bilinmeyen hücre sayısı olarak hesaplanmaktadır. Hedef nokta seçiminde üç farklı strateji kullanılmıştır: 1) Hedef noktalar rassal olarak seçilmektedir. 2) Fırsatçı bir yaklaşımla robota en yakın bilinmeyen bölgenin merkezi hedef nokta olarak kabul edilmektedir. 3) Bilinmeyen hücreler dörtlü ağaç yapısında gösterilmekte ve bu ağacın yaprakları hedef nokta olmaktadır. Robotlar hedefleri en az maliyetle gezebilecek bir tur oluşturmaktadır. Bu noktada, robotlar turlarındaki tüm görevleri haberleşebildikleri tüm robotlara duyurmakta ve market tabanlı yaklaşımlarda kullanılan süreç işletilmektedir. Bir robot bütün müzakerelerini kapattıktan sonra ilk hedefine yönelmektedir. Robot bir hedefe ulaştığında, bu süreç tekrarlanmaktadır. Eğer bir robot diğer robotun alanına yakın bir yerde hedefe sahipse bu hedef diğer robota müzakere ile aktarılmaktadır. Ayrıca, belirli zaman aralıklarında robotlar kendi haritalarının küçük parçalarını diğer robotlar ile paylaşmaktadır. Önerilen yaklaşım, farklı sayıda gerçek robotla ve üç farklı ortamda uygulanmıştır. Yapılan uygulamalarda farklı hedef seçme stratejilerinin etkisi de görülmektedir (Zlot vd. 2002).

Makarenko ve diğerleri bir hedefe gidildiğinde kazanılacak faydayı hesaplarken o hedefin algılama sınırları içindeki tüm hücrelerin ortalama bilgi yitimini (entropy) kullanmaktadırlar. Eğer bölgedeki bilgi yitimi yüksek ise o bölgeden daha çok bilgi sağlanacağı için o hedef daha ilgi çekici olmaktadır. Bir hedefe gitmenin maliyeti ise mesafe olarak hesaplanmaktadır. Önerilen yaklaşımda literatürdeki diğer çalışmalardan farklı olarak hedefin seçilmesinde konumlandırma kalitesi olarak adlandırılan bir ölçüt daha kullanılmaktadır. Yazarlar bunun sebebinin robotun konumlandırma hatası yüzünden yeni bilginin eldeki bilgiler ile doğru olarak birleştirilemeyeceği ve kullanışsız olacağı olarak açıklamaktadırlar. Konumlandırma kalitesi ölçütü bir robotun mevcut haritadaki belli bir konumda kendini tekrar konumlandırması üzerine kurulmaktadır. Sonuç olarak, robot bilgi edinmek için bilinmeyen ortamlara hareket etmek istemekte ve maliyet kısıtı yüzünden en yakın bilinmeyen ortam seçilmektedir. Buna ek olarak konumlandırma kalitesi için robot bilinen harita içinde tutulmaya çalışılmaktadır. Bu üç kriterin ağırlıklı toplamı hedeflerin seçilmesinde kullanılmaktadır (Makarenko vd. 2002).

Berhault ve diğerleri yaptıkları çalışmada robotun birden fazla hedefe gideceği durumlarda tek-öge müzayede yöntemlerinin iki hedef arasındaki ilişkiyi ortaya koymakta yeterli olmayacağını ve kombinasyonel müzayedenin bu problemde daha iyi sonuçlar üreteceğini savunmuşlardır. Dolayısıyla, kombinasyonel müzayede yöntemini farklı teklif verme stratejileri ile birlikte keşif görevi için geliştirmişlerdir. Önerdikleri yöntemi kısmen

bilinen ve tamamen bilinmeyen ortamlarda üçlü kombinasyon, akıllı kombinasyon, en yakın komşu kombinasyonu ve çizge kesim kombinasyonu teklif verme stratejileri için katettiği mesafe, harcanan zaman ve robotun faydası kriterleriyle test etmişlerdir. Bununla birlikte tek-öge müzayede yöntemi ve eniyi çözümü de merkezi olarak hesaplayarak sonuçları karşılaştırmışlardır. Elde edilen sonuçlarda, kombinasyonel müzayede yönteminin tek-öge müzayede yöntemine göre çok daha iyi sonuç vermesinin yanında en iyi sonuca da oldukça yakın olduğu gösterilmiştir (Berhault vd. 2003).

Grabowski ve diğerleri (Grabowski vd. 2003) yaptıkları çalışmada keşif görevinde hedef seçiminde literatürde sıklıkla kullanılan komşu hücre tabanlı yaklaşımların aksine engel tabanlı bir yaklaşım önermişlerdir. Yazarlar, komşu hücre tabanlı yaklaşımlarda engellerin yeterince iyi tanımlanamadığını söylemektedirler. Bunun sebebi olarak da, kullanılan ultrasonik algılayıcıların yansıma problemleri ve algılama menzili gösterilmektedir. Bu durumda, yakın olan bir engel yansımadan dolayı daha uzakta algılanmakta ve komşu hücre esasında engelin arkasındaki hayali bir yer olarak seçilmektedir. Dolayısıyla robot asla ulaşamayacağı bir hedef hücresine yönelmekte ve verimlilik düşmektedir. Bu durumların üstesinden gelmek için bir sonraki en iyi görüş (next best view) yaklaşımı kullanılmaktadır. Robot bir sonraki hedefini seçerken mevcut engellerin durumunu en iyi şekilde algılayacağı yeri seçmektedir. Bu seçim yapılırken ters algılayıcı modeli adı verilen bir yöntem kullanılmaktadır. Bu modelde robotların algılayıcılarının kesiştikleri alanlar göz önüne alınmakta ve bu alan ne kadar az olursa engeller o kadar iyi belirlenmektedir. Bu işlem, bütün algılayıcı ölçümleri ve engel noktaları için yapılmakta, birleştirilmekte ve doluluk kafesi gibi bir harita üzerinde gösterilmektedir. Bu birleştirme sonunda robotların bir sonraki harekette gitmek istedikleri yerler ortaya çıkmakta ve robotlar bu hedeflerden en yakın olanı seçerek keşif sürecine devam etmektedirler.

Stroupe ve diğerleri yaptıkları çalışmada haritalama, haberleşme, seyrüsefer gibi birden çok görevin aynı anda yapıldığı keşif ve dinamik hedef izleme görevlerinde robotların tüm robot takımının verimliliğini artıracak şekilde bir sonraki hareketi seçmelerini sağlamaktadırlar. Bu yaklaşımda, robot önce ortamı algılamakta, takım arkadaşlarına son gözlemlerini ve tamamlanmış görevi duyurmakta ve takım arkadaşlarından gelen benzer duyuruları kullanarak ortam bilgisini ve robotların konumlarını güncellemektedir. Daha sonra, her adımda, her robot, bütün görevler için takım arkadaşlarının bir sonraki adımdaki katkılarını tahmin etmektedir. Bu çalışmada, keşif için kullanılacak hedefler daha önceden belirlenmiştir. Hedef belirlendikten sonra yapılacak en iyi hareket seçilmektedir. En iyi hareketin seçilmesinde haritalama, keşif ve haberleşme görevleri için değer fonksiyonları kullanılmakta ve mümkün bütün hareketler için bir değer döndürülmektedir. Haritalamada kullanılan değer fonksiyonu haritadaki belirsizlikleri en

aza indirmeyi, keşif için kullanılan değer fonksiyonu hedeflere en hızlı şekilde ulaşmayı ve haberleşmede kullanılan değer fonksiyonu ise robotların birbirlerinin görüş alanı içinde kalmalarını sağlamaktadır. Bu fonksiyonlardan gelen değerler görevin ağırlığını gösteren sabitler ile çarpılarak ağırlıklı toplam hesaplanmakta ve en yüksek değere sahip hareket bir sonraki adımda kullanılmaktadır. Farklı ağırlık değerleri ve farklı büyüklükte ortam ve robot takımları ile önerilen yaklaşım test edilmiştir (Stroupe vd. 2004).

Amigoni ve diğerleri yaptıkları çalışmada ortamı iki liste halinde ifade etmişlerdir. Bu listelerden biri engellerin kapladığı konumları diğeri ise bilinmeyen alanların sınırlarını gösteren ayrıtları saklamaktadır. Bu çalışmada, aday hedefler bir sonraki en iyi gözleme noktası yöntemine göre seçilmektedir. Bu adaylar ikinci listedeki ayrıtların üzerinden rastsal olarak belirlenen noktalar. Bu noktadan sonra bu adaylar Pareto-eniyi kriterine göre sıralanmakta ve çoğu durumda geriye birden fazla aday kalmaktadır. Bu durumda, robotun katettiği mesafeyi enküçükleyecek, kazanılan bilgiyi enbüyükleyecek ve haritaların en iyi üst üste oturacağı nokta ideal çözüm olarak adlandırılmakta ve seçilmektedir. Yazarlar, daha önceki çalışmalardan farklı olarak bir fayda fonksiyonu ile bu üç kriteri farklı ağırlıklar ile çarpmak yerine bu kriterleri ayrı ayrı değerlendirmişlerdir. Önerilen yöntem, üç farklı ortamda test edilmiş ve geçmiş çalışmalarda yer alan fayda fonksiyonları ile karşılaştırılmıştır. Bütün testlerde önerdikleri yöntem daha iyi sonuç vermiştir (Amigoni ve Gallo, 2005).

Sariel ve Balch yaptıkları çalışmada çok robotlu keşif problemi için tek-öge müzayedeye tabanlı bir görev paylaşırma yöntemi önermişlerdir. Esasında bu problem çoklu gezgin satıcı probleminin farklı bir versiyonudur. Ancak bu problem NP-Zor bir problemdir. Bu problemin en iyi çözümünü bulmak için yönelem araştırmasında yer alan tamsayı programlama yöntemleri kullanılabilir. Bununla birlikte, paylaşılacak olan görevlerin sayısı büyüdükçe çözüm üretme süresi çok uzamakta; bilinmeyen ve dinamik ortamlarda çözümün tekrar oluşturulması gibi zorluklar meydana gelmektedir. Buna karşın, yazarlar polinom zamanda eniyeye yakın çözümler ile görevleri paylaşırın sezgiseller önermişlerdir. Robotlar bu sezgiselleri kullanarak kendilerine uygun hedefleri belirlemekte ve bu hedefleri en kısa yoldan dolaşacak şekilde yol planını hazırlamaktadır. Bu noktada, robotlar müzayedeye sunacakları en uygun görevi belirlemekte ve diğer robotlar da bu görev için maliyet değerlerini göndermektedirler. Yapılan değerlendirmelerden sonra, yol planları eğer varsa değişen hedefler için tekrar yapılmakta ve bu süreç robotlar hedeflerine vardıkça tekrarlanmaktadır. Çeşitli sezgiseller ile yapılan testlerde sonuçların eniyeye ne kadar yakın olduğu gözlemlenmiştir (Sariel ve Balch, 2006).

Wu ve diğerlerinin temel düşüncesi haberleşmenin mükemmel olduğu varsayımını yaparak robotları bilinmeyen bölgelere dağıtmaktır. Bunu yapabilmek için bilinmeyen

bölgeleri k-ortalama kümeleme yöntemi ile robot sayısı kadar kümeye ayırmayı ve her bir robotu bir kümenin merkezine göndermeyi planlamışlardır. İlk olarak, doluluk ızgarası kullanarak bu prosedürü işlettiklerinde bilinmeyen bölgelerin kümelere ayrılması çok uzun süre almıştır. Bu durumda, ana fikirlerini gerçekleştirmek için Voronoi Diyagramını kullanarak ortamı topolojik olarak ifade etmişlerdir. Voronoi Diyagramı ile ortam poligonlara bölünürken poligonun dolu ya da boş olduğuna karar verilmekte ve boş poligonların merkezleri kümelemede kullanılmaktadır. Önerilen yöntem farklı büyüklükteki ortamlarda hem doluluk ızgarası hem de topolojik harita kullanılarak test edilmiştir. Sonuç olarak, topolojik harita ile küme yapıldığında keşif süresi metrik harita ile keşif yapıldığı duruma göre yarı yarıya azalmıştır (Wu vd. 2007).

Vincent ve diğerleri yaptıkları çalışmada yaklaşık 100 adet gezgin robotu kullanarak keşif ve haritalama yapmışlardır. Keşif ve haritalama esnasında koordinasyon, harita birleştirme, kısıtlı haberleşme, görev tespit etme ve paylaşırma gibi zor durumların hemen hemen hepsini hesaba katmışlardır. Robotlar küçük takımlar (kümeler) halinde farklı noktalardan keşif ve haritalama görevine başlamaktadırlar. Her kümede bir takım lideri bulunmaktadır. Küme içindeki robotlar birbirleri ile haberleşebilmekte ve kümedeki diğer robotların göreceli konumlarını tespit edebilmekte ve yerel haritalarını paylaşabilmektedirler. Bununla birlikte, sadece takım liderleri yerel haritaları birleştirebilmektedir. Oluşan küme haritası Voronoi Diyagramı ile topolojik olarak ifade edilmekte ve robotlar yeni görevlerini bu haritanın iskeletini kullanarak seçmektedirler. İki küme karşılaştıklarında iki takım liderinden ID numarası küçük olan yeni kümenin lideri olmaktadır. Haritalar birleştirilmekte ve süreç tüm ortam keşfedilene kadar devam etmektedir (Vincent vd. 2008).

Wu ve diğerleri yaptıkları çalışmada önceki çalışmalardan farklı olarak yeni bir eniyileme kriteri ortaya atmaktadırlar. Bu çalışmada, bilinmeyen bölgeler kümelere bölünmekte ve bütün kümelerin keşif sürelerinin aynı olması istenmektedir. Dolayısıyla, bölgelerin keşif süreleri arasındaki farkı enküçükleme çalışmaktadırlar. Bununla birlikte, robotların iş yüklerinin dengelenmiş olması gerekmektedir. Bunu yapabilmek için ilk olarak bilinmeyen bölge robotların konumları ve oluşacak kümelere robotların uzaklıkları gibi kriterler göz önüne alarak kümelenebilir. Daha sonra, robotlar iş yükleri dengeli olacak şekilde kümelere dağıtılmalıdır. Bu noktada, robotun atandığı kümede birden fazla hedef olabilir. Robotlar, diğer robotlar ile haberleşebildikleri için onların konumlarını, atandıkları kümeleri bilmektedir ve buna göre hedef seçimlerini yapmaktadırlar. Önerilen yöntem literatürde öne çıkan yaklaşımlar ile birlikte gerçekleştirilerek benzetim ortamında test edilmiştir. Test sonuçları, robot sayısı arttıkça önerilen yöntemin istenen kriteri en başarılı şekilde karşıladığını göstermiştir (Wu vd. 2010) (Puig vd. 2011).

Robotlara birden fazla hedef atanması gerektiğinde bu problem kombinasyonel ve NP-Zor bir problem haline gelmektedir. Kombinasyonel müzayede kullanarak bu problemi çözmeye çalışan yaklaşımlar olsa bile elde ettikleri sonuçlar eniyi altı olmaktadır. Carvalho ve diğerlerinin önerdikleri yaklaşımının motivasyon noktası, robotlara birden fazla işi kombinasyonel yöntemlere gerek olmadan atamaktır. Bunun için, çok robotlu keşif problemini dağıttık çizge renklendirme problemi ile modellemişlerdir. Her robot farklı bir rengi temsil etmektedir. Her hedef çizge de bir düğüme karşılık gelmektedir. Eğer, iki düğüm arasındaki mesafe belirli bir eşik değerinden büyükse bu düğümler arasında bir ayrıt yeralabilmektedir. Ayrıca, eğer düğümler arasında bir ayrıt yoksa bu düğümler aynı renge sahip olabilmektedir. Sonuç olarak, birbirine yakın olan düğümler aynı robot tarafından keşfedilmekte ve robotların katettikleri mesafe azalmaktadır. Keşif algoritmasında robotlar 4 durumda olabilmektedirler. İlk olarak, robotlar yeni hedefler bulmak amacıyla rastsal olarak gezinmektedir. Daha sonra buldukları yeni hedef ile ilgili bilgileri diğerleri ile paylaşmaktadırlar. Bir sonraki aşamada hedefler renklendirilmektedir. Son olarak, robot hedeflerini gezmeye başlamaktadır. Önerilen yöntem kombinasyonel müzayede yöntemi ile karşılaştırılmış ve zaman, mesafe ve haberleşme kriterlerinde daha iyi sonuç üretmiştir (Carvalho vd. 2013).

Hawley ve Butler yaptıkları çalışmada market tabanlı bir yaklaşım kullanarak keşif problemini çözmeye çalışmaktadırlar. Geleneksel market tabanlı yaklaşımlarda olduğu gibi görevler için müzayede açıldığı gibi bir de robotlar için müzayede açılmaktadır. Bu noktada yazarlar aynı hedefi keşfetmek amacıyla kendi istekleriyle ve aynı anda hareket eden robotlar kümesini koalisyon olarak tanımlamaktadırlar. Her koalisyon bir yönetici ve sıfır ya da daha fazla sayıda işçiden oluşmaktadır. Robotlar sadece bir rolde bulunabilmektedirler ancak bu rollerin değişmesi mümkündür. Bir robot yönetici iken bir görevi tamamladığında başka bir görev için işçi olabilmekte ya da bir işçi bir görev bulduğunda yönetici olabilmektedir. Robotlar daha kârlı bir görev bulmaları ya da koalisyonun görevinden uzaklaşmaları gibi nedenlerden dolayı koalisyonu bozabilmektedirler. Önerilen yöntem farklı sayıda robot içeren robot takımları, sadece görevler için müzayede yapan yöntem ve fırsatçı yöntem için 2 farklı ortamda test edilmiştir. Test sonuçları önerilen yöntemle özellikle büyük ve yapısal ortamlarda diğer iki yöntemle göre çok kısa sürede keşfin tamamlandığını göstermektedir (Hawley ve Butler, 2013).

Chand ve Carnegie yaptıkları çalışmada hiyerarşik bir yapı kullanarak keşif yapmayı amaçlamışlardır. Robot takımının üyeleri farklı algılama, hesaplama, hareket etme ve haberleşme yeteneklerine sahiptir. Hesaplama gücü ve haberleşme yeteneği yüksek olan robotlar yönetici olarak görev yapmaktadır. Bununla birlikte hesaplama gücü ve haberleşme yeteneği orta seviyede olanlar planlayıcı, hareket etme ve algılama gücü yüksek olanlar ise

kaşif olarak çalışmaktadırlar. Kaşif, kendine planlayıcı tarafından gönderilen yol planını uygulayarak hedefine gitmekle ve geçtiği ortamı algılayarak elde ettiği bilgileri yöneticiye göndermekle yükümlüdür. Planlayıcının iki temel görevi vardır. Bunlardan biri kaşiflere yol planlarını göndermek, diğeri ise küresel atamaları yerel atamalara çevirmektedir. Yöneticinin görevleri ise iş kuyruğu yardımıyla sınırlı kaynaklara sahip robotları koordine etmek ve küresel harita ve yerel haritaların durumlarını güncellemektir. Önerilen yaklaşım, geçmiş çalışmalarda bulunan yaklaşımlar ile karşılaştırılmış ve verimliliği gösterilmiştir (Chand ve Carnegie, 2013).

5.1.2 Çok robotlu sınır hücre keşif yaklaşımları

Yamauchi'nin 1998 yılındaki (Yamauchi, 1998) yayınında komşu tabanlı keşif yaklaşımı çok robotlu sistemler için genişletilmektedir. Bu yaklaşımda, her robot ortam hakkındaki bilgisini içeren kendi küresel kanıt ızgarasına sahiptir. Bir robot yeni bir komşuya ulaştığında, algılayıcıları ile edindiği bilgiler ışığında yerel bir kanıt ızgarası oluşturmakta ve kendi yerel kanıt ızgarasını küresel ızgara ile birleştirirken takım arkadaşlarından gelen yerel ızgaraları da aynı zamanda birleştirmektedir. Daha sonra bu yerel kanıt ızgarasını diğer robotlara duyurmaktadır. İki kanıt ızgarası basit bir yol ile birleştirilmektedir. İki ızgarada bir birine karşılık gelen hücrelerin olasılıklarının ortalaması yeni hücrenin olasılığı olarak saklanmaktadır. Bu yaklaşımın avantajı hem dağıtık hem de işbirlikçi olmasıdır. Herhangi bir robot tarafından bulunmuş olan bilgi diğer robotlar tarafından da kullanılabilir. Bu şekilde robotlar gidecekleri yeni komşuları belirlerken diğer robotlardan gelen bilgileri kullanabilmektedirler. Bu bilgiye dayanarak robot diğer robotlar tarafından keşfedilmemiş alanlara yönelebilecektir. Aynı zamanda dağıtık yapıda olan bu yaklaşımda bir robotun çalışmaz hale geldiği durumda diğer robotlar keşfe devam etmektedir. Bu yaklaşımın dezavantajlarından biri optimal çözümü garanti edememesidir. Robotlar aynı komşuya ulaşmak için zaman kaybedebilmektedirler.

Burgard ve diğerleri (Burgard vd. 2000) yaptıkları çalışmada teklif tabanlı bir yöntem ile keşif sürecinde robotları koordine etmektedirler. Önerilen yaklaşımda maliyetlerin hesaplanmasında robotun mevcut konumundan hedefe olan en iyi yola dinamik programlama yöntemi ile karar verilmektedir. Bu yaklaşımda, bir hücrenin maliyeti o hücrenin doluluk olasılığı ile doğru orantılıdır. Dolayısıyla maliyet hücrenin doluluk olasılığı ile o hücreye olan uzaklığın çarpılması ile hesaplanmaktadır. Komşu hücrelerin ziyaret edilmesi ile kazanılacak fayda hesaplanırken ilk başta her komşu hücrenin faydasına bir değeri atanmaktadır. Bir hedef hücre t' bir robot tarafından seçildiğinde, bu t' hedefinin d komşuluğundaki diğer hedef hücrelerin faydaları $P(d)$ olasılığı kadar azaltılmaktadır. Hedef hücrenin seçiminde bahsedilen fayda ve maliyet göz önüne alınarak tekrarlamalı bir yaklaşım kullanılmaktadır. Her turda en yüksek teklif verilen hedef hücreye bu teklifi

veren robot atanmakta ve bu atamaya bağılı olarak hedef hücrelerin faydaları tekrar hesaplanmaktadır. Bu süreç tüm robotlara hedef hücre atanana kadar devam etmektedir. Yapılan deneylerde koordine edilmiş sistemin koordine edilmemiş sisteme göre daha hızlı keşif görevini tamamladığı hem gerçek robotlarda hem de benzetim ortamında gösterilmektedir.

Simmons ve diğerleri (Simmons vd. 2000) robotların diğer robotların konumlarını bildikleri ve mükemmel bir haberleşmeye sahip oldukları varsayımlarını yapmaktadırlar. Önerilen yaklaşımda dağıtık hesaplama ve küresel karar verme mekanizmaları keşif görevini koordine etmek için birleştirilmektedir. Hedefe gitme maliyetinin hesaplanmasında komşu hücre ve robotun pozisyonu arasındaki en kısa yol bulunmaktadır. Hedefe gittiğinde elde edeceği bilgi komşu hücrenin belli bir komşuluğundaki bilinmeyen hücre sayısı olmaktadır. Robotlar merkezi haritacıdan her yeni bilgi aldıklarında bahsedilen maliyet ve bilgi kazancını hesaba katarak teklif hazırlamaktadırlar. Merkezi bir yönetici bu teklifleri değerlendirerek keşif görevi süresince robotların bir yeri birden fazla kez dolaşmalarını en aza indirecek şekilde görevleri atamaktadır. Görevlerin atanmasında, merkezi yönetici ilk olarak, en yüksek teklife sahip komşu hücreyi bulmakta ve teklifi veren robota bu hedefi atamaktadır. Bu noktada, tekliflerde bir indirim yapılmaktadır. Bu indirim miktarı atanan hedef ile atanacak hedefin paylaştığı bilinmeyen hücre sayısı ile doğru orantılı olmaktadır. Böylece daha önce atanmış hedeflere yakın olan hedefler için verilecek teklifler azalacak ve robotlar başka bölgelere atanacaktır. Son olarak, bir robotun mevcut atanmış hedefinin bilgi alanının içinde bir komşu hücre mevcutsa bu hedefin bilgi kazancı 0.85 gibi bir sabitle çarpılmaktadır. Böylece robotun başka bir alana gitmeden önce bulunduğu alandaki komşu hücreleri ziyaret etmesi sağlanmaktadır.

Diaz ve diğerleri (Diaz vd. 2001) yaptıkları çalışmada ortamın yapısında bulunan kapı, koridor gibi yapılar sayesinde robotların koridorlarda gezinebilmesini, kapıları keşfedebilmesini ve odalara girebilmesini sağlamaktadır. Önerilen yaklaşımda iki temel davranış, "koridor boyunca ilerle" ve "kapıya doğru git", tanımlanmaktadır. Bu davranışların birleştirilmesi ile robotlar bilinmeyen bir ortamın keşfini yapabilmektedirler. Koridorların tanınmasında sadece koridorun genişliği ve açısına ihtiyaç duyulmakta ve bu bilgiler lazer ışınları kullanılarak elde edilmektedir. Kapıların tanınmasında kapının genişliği kullanılan robota göre seçilmektedir. Robotun elde ettiği lazer ışınları koridorun merkez çizgisi ve x eksenini çakışacak şekilde dönüştürülmektedir. Daha sonra kutu filtresi ile kapılar tespit edilmektedir. Ancak yanlış tespitler olabilmekte ve bunu önlemek için kapı ile aynı yöndeki ölçümlere bakılmaktadır. Önerilen yaklaşım, biyolojik tehlike arz eden bir maddeyi bulmak için kullanılmaktadır. Robot binanın bir ucundan başlayarak bahsedilen davranışları ard arda yaparak maddeyi bulmakta ve resmini göndermektedir.

Stachniss ve Burgard (Stachniss ve Burgard, 2003) robotların ortamın durumunu ifade etmek için kullanılan yeni bir olasılıksal gösterim şekli önermişlerdir. Doluluk kafesi yönteminde, bir hücrenin çok az bir kısmı dahi bir engel içerse bu hücre dolu hücre olarak sınıflandırılmaktadır. Önerilen yaklaşımda, hücrelerin doluluk oranları olasılıksal olarak tutularak ortamın gerçek durumunun ifade edilmesi sağlanmıştır. Ortam bu yöntem ile ifade edildikten sonra eniyi görüş noktası tabanlı bir keşif algoritması kullanılmıştır. Bu noktada, eniyi görüş noktası seçilmesinde iki farklı yaklaşım kullanılmaktadır. Bir yaklaşımda, robotun haritasındaki belirsizlikler en aza indirilmeye çalışılırken diğer yaklaşımda robotun gideceği mesafe azaltılmaya çalışılmaktadır. Bir hücrenin belirsizliğinin belirlenmesinde bilgi yitimi (entropy) yaklaşımı kullanılmaktadır. Eniyi görüş noktası seçiminde üç farklı strateji kullanılmıştır. İlkinde, robot bilgi kazanımını göz ardı ederek sadece robotun gideceği yolu en aza indirmek amacıyla seçimler yapmaktadır. İkinci strateji, robotun en fazla bilgiyi alabileceği noktalara gitmeyi amaçlamaktadır. Son strateji ise bu iki yöntemi belli ağırlıklarda birleştirmektedir. Önerilen yaklaşımı test etmek amacıyla yapılan deneylerde, yolu en aza indirmeye çalışan stratejinin hem yapılan ortalama ölçüm sayısında hem de robotun katettiği mesafede diğer yöntemlerden daha iyi sonuç verdiği görülmüştür.

Solanas ve Garcia yaptıkları çalışmada çok robotlu sistemin avantajını ortaya çıkarmak, robotların daha az mesafe katetmesini ve keşfin hızlı olmasını sağlamak amacıyla robotları ortamın farklı bilinmeyen noktalarına göndermeyi amaçlamışlardır. Bunu yapabilmek için, tamamen sınırlı bir ortamda çalıştıkları varsayımını yaparak ortamı doluluk ızgarasıyla ifade etmişlerdir. Bu noktada, bilinmeyen hücreler k-ortalama kümeleme yöntemi ile uygun robot sayısına göre kümeler ayrılmakta ve robotlar kendilerine en yakın kümeyi keşfetmekle yükümlü kılınmaktadır. Bu sayede geçmiş çalışmalarda yer alan maliyet ve fiyat gibi unsurların dengelenmesi için gerekli olan deneysel ayarlamalara da gereksinim azalmakta ve robotlar sadece en yakın kriterini kullanmaktadırlar. Robotların kendi bölgelerinde bilinmeyen kümeler olması durumunda başka bir bölgedeki bilinmeyen küme için önemli bir ceza yüklenmeleri gerekmektedir. Bu durumda, robotlar buldukları bölgeleri keşfetmektedirler. Robotlar keşiflerini sürdürerek yeni alanlar buldukça bilinmeyen bölgeler sürekli olarak yeniden kümelenebilir ve bu süreç ortam tamamen keşfedilene kadar devam etmektedir. Yapılan deneyler bu yöntemin geçmiş çalışmalarda yer alan yaklaşımlara göre daha hızlı keşif yaptığını göstermiştir (Solanas ve Garcia, 2004).

Burgard ve diğerleri teklif tabanlı karar teorik (decision- theoretic) bir yöntem ile keşif sürecinde robotları koordine etmektedirler. Yazarlar, başlangıçta haberleşmenin mükemmel olduğunu varsayarak önerdikleri yaklaşımı haberleşmenin kısıtlı olduğu durumlara uyarlayarak sunmuşlardır. Hedef hücrenin seçiminde fayda ve maliyetin göz önüne alındığı tekrarlamalı bir yaklaşım kullanılmaktadır. Her turda en yüksek teklif verilen

hedef hücrelerine bu teklifi veren robot atanmakta ve bu atamaya bağlı olarak hedef hücrelerin faydaları tekrar hesaplanmaktadır. Bu süreç tüm robotlara hedef hücre atanana kadar devam etmektedir. Haberleşmenin sınırsız olduğu durumlarda, robot atandığı hedefe vardığında, en son atamadan sonra belirli bir mesafe katettiğinde ya da belirli bir zaman geçtikten sonra robota yeni atamalar yapılmaktadır. Haberleşmenin sınırlı olduğu durumda robotlar diğer robotların son hedef noktalarını tutarak bu noktalara gitmeyi önleyecek bir yapı kullanılmaktadır. Robot takımının bütün üyelerinin haberleşemediği uç durumda dahi tüm alanın keşfedileceği ancak bu durumda yaklaşımın koordine edilmemiş yaklaşımla aynı sonuçları ürettiği görülmüştür (Burgard vd. 2005).

Fox ve diğerleri (Fox vd. 2006) yaptıkları çalışmada robotların birbirlerinin ilk konumlarını bilmediği durumlarda bilinmeyen bir ortamın haritasını yüksek doğrulukta üretmesini sağlamaktadırlar. Robotlar edindikleri bilgileri paylaşılan bir haritada birleştirmekte ve bu haritayı da keşif görevini koordine etmekte kullanılmaktadır. Bu haritanın doğruluğunu artırmak amacıyla robotlar aktif olarak, yani diğer robotların yanına giderek, onların konumlarını doğrulamaktadır. Bu yöntemde, her robot keşif ve haritalama yaparak işe başlamaktadır. İki robot haberleşebildiğinde algılayıcı bilgileri ve göreceli konum tahminlerini birbirlerine göndermektedirler. Eğer iyi bir tahmin gerçekleştiyse robotlar belirli bir noktada buluşarak aktif olarak doğrulama yapmaktadırlar. Bu noktada bu iki robot bir keşif kümesi oluşturarak haritalarını birleştirmekte ve keşfi koordine etmeye başlamaktadırlar. Eğer doğrulama başarısız olursa robotlar sadece algılayıcı bilgilerini birbirlerine göndererek keşif yapmaya devam etmektedirler. Bir keşif kümesi içinde bulunan robotlar iki görevden birini yerine getirmektedirler. Bu görevler keşfe devam ederek bir hedef hücreye gitmek ya da başka bir robotu keşif kümesine katmak amacıyla aktif doğrulama yapmaya gitmektir. Robotlar bu iki görevden hangisini yapacaklarına fayda-maliyeti ençoklamaya çalışarak karar vermektedirler. Maliyet hesabında, komşu hücre için, robot ve komşu hücre arasındaki en kısa mesafe, doğrulama için ise robotların buluşabilecekleri en kısa mesafe ve buluşup buluşamamanın maliyeti kullanılmaktadır. Fayda, komşu hücre için, bilinmeyen hücre sayısı ile, doğrulama için ise buluşulacak robotun harita boyutu ve robotların buluşma olasılıklarının çarpımı ile bulunmaktadır. Bu noktada bütün kombinasyonlar için fayda ve maliyet kullanılarak doğrusal programlama yardımıyla en iyi atamalar bulunmaktadır. Haberleşmenin kısıtlı olduğu durumlarda, haberleşme alanından çıkan robot keşfe devam etmektedir. Haberleşme alanına girdiği zaman, topladığı bilgileri paylaşmaktadır. Robot takımının bütün üyelerinin haberleşemediği uç durumda dahi tüm alan keşfedilmekte ancak verimlilik azalmaktadır.

Howard ve diğerleri yaptıkları çalışmada keşfedilmemiş bir binanın içine robotları yerleştirmeyi, bu robotların binanın haritasını çıkarmasını ve binadaki istenmeyen öğelerin (insan, hayvan, başka bir robot gibi) belirlenmesini ve takip edilmesini amaçlamaktadırlar.

Bunu yapabilmek için yaklaşık olarak 80 robotun bulunduğu heterojen bir robot takımı kullanmışlardır. Bu kadar büyük bir robot takımını oluştururken maliyeti göz önünde bulundurmamak gerekmektedir. Dolayısıyla oluşturulan robot takımı az sayıda, lazer mesafe algılayıcı, kamera ve güçlü bilgisayara sahip, yüksek yetenekli ve pahalı robota sahipken; çok sayıda robot ise, sadece mikrofon, basit bir kamera ve zayıf bilgisayara sahip, az yetenekli ve ucuz olacak şekilde seçilmiştir. Önerilen yöntemde, keşif sırasında sadece yerel doluluk ızgarası kullanılmıştır. Dolayısıyla, eski bilgiler haritadan silinirken yeni bilgiler eklenmektedir. Bu durumda, robotun seçtiği hedef nokta yeni gözlemlerde farklı bir durumda bulunabilir. Örneğin başka bir robot tarafından keşfedilmiş ya da engel içeren bir bölgede olabilir. Bu noktada robot, rastsal olarak yeni bir hedef belirlemektedir. Bu yaklaşımın önemli avantajı robotlar arasındaki iletişimin kapalı bir biçimde, yani doğrudan değil ortamdaki değişiklikler üzerinden yapılmasıdır. Bununla birlikte, robotun kısa süreli yerel haritaya sahip olması keşfettiği bölgeleri unutmasına ve tekrar tekrar keşfetmesine neden olacaktır. Benzer şekilde, kapalı iletişim yüzünden birden fazla robot aynı bölgeyi keşfedebilirler (Howard vd. 2006).

Stachniss ve diğerleri (Stachniss vd. 2006) binaların genelde koridorlar, yanyana dizilmiş odalar ve ofislere açılan yapılar olduğunu söyleyerek, keşif görevinde bu bilgiden yararlanılabileceğini savunmaktadır. Lazer uzaklık ölçerden alınan ölçümler yardımıyla ortamdaki koridorları bulmak için sistem eğitilmektedir. Daha sonra eğitilen bu sistem yardımıyla belli bir noktanın koridor mu yoksa başka bir yer mi olduğuna saklı Markov modeli kullanılarak karar verilmektedir. Bu bilgi, hedef noktalarının faydalarını hesaplamakta kullanılmaktadır. Eğer bir hedef koridorda ise onun faydası odalardaki hedeflere göre daha yüksek olmaktadır. Böylece robotlar ortama daha iyi dağılmakta ve hızlıca birçok potansiyel hedef noktası buldukları için keşif süresi önemli ölçüde azalmaktadır. Önerilen yaklaşımda maliyetler mesafe tabanlı hesaplanmaktadır. Fayda ise o noktaya gittiğinde kazanacağı bilgi olarak tanımlanmıştır. Robotların aynı hedefe gitmelerini önlemek için doğrusal bir indirim fonksiyonu uygulanmaktadır. Faydadan maliyet çıkarılarak her hedef için değerler belirlenmekte ve en yüksek değerden başlayarak her robota hedefler atanmaktadır. Yapılan deneylerde bu yöntem kullanıldığında keşif süresinin önemli ölçüde azaldığı görülmüştür.

Visser ve Slamet (Visser ve Slamet, 2008b) yaptıkları çalışmada komşu hücrelerin belirlenmesinde güvenli bölgeler olarak tanımlanan yöntemden yararlanmaktadırlar. Bu güvenli bölgelerin sınırları komşu hücreler olarak seçilmektedir. Bir komşu hücre için fayda fonksiyonu komşu hücrenin bulunduğu keşfedilmemiş alanın robotun komşu hücreye olan uzaklığı oranı ile hesaplanmaktadır. Komşu hücreler ve faydaları bahsedildiği gibi hesaplandıktan sonra bütün robot-hedef hücre atamaları için bir fayda matrisi U oluşturulmaktadır. Bu matristeki faydalar hesaplanırken robot-hücre arasındaki uzaklık

kartezyen uzaklık olarak kullanılmaktadır. Tüm matris oluşturulduktan sonra en yüksek faydalı atama için bir yol planı oluşturulmaktadır. Bu yol planı sonucunda çıkan fayda ile mevcut fayda aynı ise bu atama yapılmakta ve ilgili satır ve sütunlar matristen çıkarılmaktadır. Bir sonraki en yüksek fayda için aynı süreç işletilmektedir. Bu süreç tüm robotlara komşu hücre atanana kadar devam etmektedir. Robotun konumundan komşu hücreye gidecek yol planı yapılırken engeller göz önüne alınmakta ve yana öncelikli arama (breath-first search) algoritması kullanılmaktadır.

Wurm ve diğerleri (Wurm vd. 2008) önerdikleri yaklaşımda komşu hücre gibi tek bir hedef yerine ortamın belli bir bölümünü (segment) koordinasyon amacıyla kullanılmaktadır. Robotlara atama yapmak amacıyla o ana kadar keşfedilmiş alanlar bölümlere ayrılmaktadır. Bu ayrışma sayesinde robotlar ortama verimli bir şekilde dağılmakta, tekrar gezilen alanlar azalmakta ve keşif süresi önemli ölçüde kısalmaktadır. Yaklaşımda hedeflerin robotlara atanması için Macar algoritması kullanılmaktadır. Bununla beraber haritanın bölümlere ayrılması için bir çizgenin parçalanması yaklaşımı öne çıkmaktadır. Bu amaçla, çok popüler çizge tabanlı bir gösterim olan Voronoi çizgeleri kullanılmaktadır. Ortamın Voronoi çizgesi çıkarıldıktan sonra bu çizge kritik noktalar adı verilen ve kapı, dar koridor girişleri olarak tanımlanan yerlerden k ayrık kümeye parçalanmaktadır. Bu kümelerin her biri robotlara atanabilecek bölümlerdeki düğümlerden oluşmaktadır. Ortam bahsedilen şekilde parçalandıktan sonra bölümlerin içindeki komşu hücrelere karar verilmektedir. Robotların bölümler içindeki kendilerine en yakın olan komşu hücrelere gitme maliyetleri hesaplanmaktadır. Eğer bir robot bir bölümün içinde ise o bölümle ilişkili maliyet bir katsayı ile çarpılarak azaltılmaktadır. Böylece robotun bulunduğu bölüm içindeki bütün komşu hücreleri gezdikten sonra bölümden ayrılması amaçlanmaktadır. Maliyetler hesaplandıktan sonra bölümler robotlara Macar algoritması kullanılarak atanmaktadır. Macar algoritması, robot sayısı bölüm sayısından fazla olmadıkça her bölüme bir robot atamaktadır. Bununla birlikte, böyle bir durum ile karşılaşıldığında bölüm içindeki yerel atamalarda da Macar algoritması kullanılmaktadır. Yapılan deneylerde robotların birbirlerinin konumlarını bildiği varsayımı yapılmakta ve robotlar ortak bir haritayı kullanmaktadırlar. Ayrıca, merkezi bir planlama bileşeni tüm robotlar ile haberleşebilmekte ve yeni görevlerini robotlara atamaktadır. Haberleşmenin kısıtlı olduğu durumlarda, haberleşebilen robotlar içlerinden birini merkezi planlayıcı olarak seçerek göreve devam edebilmektedirler.

Marjovi ve diğerleri labirent benzeri ortamlarda yangın tespiti yapmayı amaçlamışlardır. Bu çalışmada robotların farklı bölgelere dağılımları ve yangın bölgelerini en kısa sürede tespit etmeleri istenmektedir. Bunu yapabilmek için robotlar A* yol planı ile birlikte engellerden sakınmak için potansiyel alan yaklaşımını kullanmışlardır. Robotlar elde ettikleri bilgileri merkezi bir sunucuya göndermektedirler. Bu merkezi sunucu küresel

bir harita tutmakla birlikte, mevcut sınır hücrelerini, eğer varsa bu sınır hücrelerine atanan robotlar gibi bilgileri topolojik bir haritada tutmaktadır. Robotlar görevleri daha iyi paylaşabilmek ve daha iyi koordine olabilmek için sunucudan küresel haritayı isteyebilmektedirler. Yazarlar son olarak robotlar arasında meydana gelebilecek çarpışmaları önlemek için bir kurallar kümesi önermişlerdir. Örneğin dar bir koridorda robotların birbirlerine zıt yönelimlerde buldukları bir durumda, bu durumu çözmek için yeniden görev dağıtımı ya da bir robotun hedefine gitmesi gibi kurallar belirlemişlerdir. Robotlar iki farklı renkte kimliğe sahiptirler ve görüntü işleme yolu ile birbirlerini tanımakta ve davranışlarını koordine etmektedirler. Son olarak, önerilen yöntemi bu kısıtlanmış kurallar ile birlikte labirent benzeri ortamlarda test etmişlerdir (Marjovi vd. 2009).

Franchi ve diğerleri algılayıcı tabanlı rassal çizge adını verdikleri gösterim ile ortamın haritasını ifade etmeye çalışmaktadırlar. Robotun heryöne algılayıcısı ile robotun bulunduğu noktanın belli bir komşuluğu içindeki boş bölgeler yerel güvenli bölge olarak tanımlanmaktadır. En az bir robot ile ziyaret edilmiş bu yerel güvenli bölgeler algılayıcı tabanlı rassal çizgenin düğümlerini oluşturmaktadır. Bu çizgenin ayrıtları ise bu güvenli bölgeler arasındaki güvenli yolları ifade etmektedir. Bu ayrıtlar en az bir robotun bu yol üzerinden geçmesiyle ya da birleştirilebilecek düğümlerin belirlenmesi ile oluşturulmaktadır. Robot üzerinde dört erkin bulunmaktadır. Bunlardan ilki hareket planlayıcısı olarak adlandırılmakta ve diğer robotlarla koordineli ve işbirlikçi olarak gidilecek bir sonraki hedefi seçmektedir. Algılayıcı tabanlı rassal çizge yöneticisi yerel haritayı güncellemek ile sorumludur. Bu güncelleme robotun kendi hareket planlayıcısından aldığı ya da takımdaki diğer robotların gönderdikleri bilgiler ile yapılmaktadır. Yayınlayıcı erkin haberleşme menzilineki diğer robotlara güncel haritayı duyurma görevini yerine getirirken dinleyici erkin ise diğer robotlardan harita bilgilerini dinlemektedir. İki algılayıcı menzili kadar mesafede olan robotlar birbirleri ile haberleşerek haritalarını eşlemektedirler. Bu işlem bir zincir halinde uzanıp gitmekte ve bu zincire dahil bütün robotların haritaları eşleşene kadar devam etmektedir. Böylece, robotlar koordineli ve işbirlikçi olarak çalışabilmektedirler. Önerilen yöntem, iki farklı ortamda test edilmiş ve robotların yükleri paylaştıkları ve işbirlikçi ve koordineli bir şekilde çalıştıkları gösterilmiştir (Franchi vd. 2009).

Haumann ve diğerleri keşif problemine kapsama probleminden esinlenerek yeni bir çözüm yolu bulmuşlardır. Önerdikleri çalışmada keşif problemindeki hedef seçme ve yol planı hazırlama adımlarını birleştirmişlerdir. Ortam Voronoi diyagramı ile ifade edilmiş ve her robota bulunduğu Voronoi hücresinin tamamını kapsaması görevi verilmiştir. Robotların bunu yapabilmesi için sadece tek bir Voronoi hücresinden elde ettikleri bilgiyi kullanarak belirlenen amaç fonksiyonunu eniyilemeleri yeterli olacaktır. Bu yaklaşımın bir

diğer avantajı ise şudur: Robotlar eğer atandıkları Voronoi hücrelerinde kalırlarsa doğal olarak robotlar arasında oluşabilecek çarpışmalar da önlenecektir. Bunun sebebi, birbirine komşu olan Voronoi hücrelerindeki robotların haberleşebildikleri varsayımının yapılmasıdır. Böylece her robot kendi hücresi için belirlenen amaç fonksiyonunu eniyileyerek paralel olarak ortamın keşfini yapmakta ve komşu hücrelerdeki robotlarla haritalarını birleştirmektedir. Bu noktada, amaç fonksiyonuna her robot için robotun önünde bulunan sınır hücreler ile yaptığı açıları en küçüklemek olarak karar verilmiştir. Bulunan açı değeri doğrudan robotun birinci derece dinamik denklemlerine uygulanmaktadır. Önerilen yöntem hem tek robotlu hem de çok robotlu sistemler için test edilmiştir. Önerilen yöntem ile her robot farklı bir açı değeri üretmiş ve robotlar ortama hızlıca yayılmışlardır. Ortamın büyük kısmı hızlıca keşfedildikten sonra geri kalan küçük alanlara keşfin sonuna doğru hareket edilmiştir (Haumann vd. 2010).

Julia ve diğerleri yaptıkları çalışmada davranış tabanlı bir yaklaşım önermişlerdir. Bu yaklaşım ile birleştirilmiş keşif problemini ele almışlardır. Bu problemi, üretilen haritadaki ve haritadan oluşturulacak olan yollardaki hatayı en küçükleyecek şekilde yapılacak olan hedef seçimleri olarak tanımlamaktadırlar. Dolayısıyla, sadece keşif süresini azaltmaya değil aynı zamanda robotun her an konumlandırma hatasını da en küçüklemeye çalışmaktadırlar. Yazarlar tepkisel ve bilinçli olarak iki katmandan oluşan ancak her iki katmanın da aynı önem seviyesine sahip olduğu melez bir yaklaşım önermişlerdir. Bilindiği üzere, tepkisel yaklaşımların en önemli dezavantajı yerel enküçük noktalarında takılmalarıdır. Bu durumda, robot bu durumu saptamak ve çözmek için yeniden plan yapmak zorundadır. Dolayısıyla bu süreçte zaman kaybedecektir. Ayrıca, bu duruma çözüm bulmak için genellikle rassal bir yön seçilerek bu yerel enküçük noktadan uzaklaşmak amaçlanmaktadır. Seçilen rassal yön robotu hedefinden uzaklaştırabilmektedir. Yazarlar, bu gibi durumların üstesinden gelebilmek için beklenen güvenli bölge ve geçiş hücresi tanımlarını yapmaktadırlar. Beklenen güvenli bölge robotun merkezinde bulunduğu belirli bir yarıçap içerisindeki bilinmeyen ve boş hücrelerden oluşmaktadır. Geçiş hücresi ise hali hazırda robotun beklenen güvenli bölgesinde olan ama bir sonraki boş hücre ile bu alandan çıkılan boş hücreyi tanımlamaktadır. Bilinçli davranışsal katman ile robot mevcut beklenen güvenli bölgeyi keşfetmek, bölge değiştirmek ya da konumlandırmayı tekrar yapmak üzere eski pozisyonlardan birine dönmek gibi uzun vadeli yapacaklarını planlamaktadır. Bununla birlikte, tepkisel davranışsal katman ise sınır hücreye gitmek, engellerden sakınmak ya da geçiş hücresine gitmek gibi kısa vadeli hareketleri planlamaktadır. Yazarlar, önerdikleri yaklaşımı geçmiş çalışmalarda yer alan yaklaşımlar ile karşılaştırmış ve önerdikleri yöntemin verimliliğini göstermişlerdir (Juliá vd. 2010).

Wang ve diğerleri yaptıkları çalışmada çok robotlu keşifi koordine etmek için parçacık sürüsü eniyilemesi kavramını kullanmışlardır. Bu yaklaşımda, ortam iki boyutlu

bir ızgara ile ifade edilmektedir. Robot bu ızgara üzerinde sadece sağ, sol, ön ve arka olmak üzere dört yöne ve tek hücre hareket edebilmektedir. Algoritma ızgarayı alt ızgaralara bölerek ve robotları ise rassal olarak bir alt ızgaraya yerleştirerek keşfe başlamaktadır. Robotlar keşif ve yürüme olarak adlandırılan iki durumdan birinde bulunmaktadır. Keşif durumunda, robot bulunduğu alt ızgarayı keşfetmek ile görevlidir ve bu alt ızgaranın keşfi tamamlanmadan başka bir alt ızgaraya geçemez. Alt ızgaraların keşfi sırasında robot bulunduğu hücreyi boş hücre olarak işaretlemekte ve dört komşuluğundaki hücrelere bakmaktadır. Eğer bu komşular alt ızgaranın sınırında ise bunları duvar olarak işaretlemektedir. Bu şekilde sadece boş hücreleri kullanarak alt ızgarasının keşfini tamamlamaktadır. Bu noktada ikinci durumu olan yürüme durumuna geçmektedir. Yürüme durumunda robot başka bir alt ızgaraya geçiş yapacaktır. Bunu yapmak için seçilecek hareketler robot takımının pozisyonu ve küresel ızgaranın durumuna göre belirlenmektedir. Robot, diğer robotların alt ızgaralarının sınırlarını duvar olarak işaretlediği için duvarlarla karşılaşmayacağı sınır hücrelerini seçmektedir. Dolayısıyla robotlar sürekli olarak ortama dağılacakları için birbirlerinin yolunu tıkama gibi bir durumla karşılaşmamakla birlikte ortamın keşif süresi azalacaktır. Önerilen yaklaşım hem benzetim ortamında hem de gerçek robotlarla test edilmiştir. Elde edilen sonuçlar, yöntemin tekrar keşfedilen alanları azalttığını ve robot sayısının artması durumunda ölçeklenebilir olduğunu göstermiştir (Wang vd. 2011).

Bautin ve diğerleri robot takımının ortama dengeli bir şekilde yayılmasını sağlayacak dağıtık ve hesaplama maliyeti açısından verimli bir sınır hücre paylaşırma yaklaşımı önermişlerdir. Bu yaklaşımda, geçmiş çalışmalardan farklı olarak her robot her sınır hücreye ulaşmak için gerekli olan mesafe açısından diğer robotlar arasındaki sırasını belirlemektedir. Bu sıralamada en düşük sıraya sahip olan robotlar sınır hücrelere atanmaktadır. Başka bir deyişle, sınır hücre ile bu sınır hücreye atanacak robot arasında en az robot olması istenmektedir. Bu çalışmada, haberleşmenin mükemmel olduğu varsayımı yapılmakta ve robotlar takım arkadaşlarının pozisyonlarını bilmektedirler. Ayrıca, ortam doluluk ızgarasıyla ifade edilmiştir. Doluluk ızgarasındaki her hücreden sınır hücrelerine boş ya da bilinmeyen hücreler kullanılarak mesafe hesabı yapılmaktadır. Dolayısıyla, diğer robotların bu sınır hücreye olan uzaklıkları kolaylıkla bulunabilmektedir. Sonuç olarak, robotlar sınır hücreler için mesafe açısından kaçınıcı sırada olduklarına dağıtık ve hesaplama açısından verimli bir şekilde karar verebilmektedirler. Bu sıralamaları kullanarak her robot bir sonraki sınır hücrelerini kendi belirlemektedir. Yazarlar bu yöntemi en küçük pozisyon olarak adlandırmışlardır. Önerilen yaklaşım test edildiğinde, bu yöntem ile robotların ortama hızlıca yayıldıkları ve keşif süresinin azaldığı gösterilmiştir (Bautin vd. 2012).

Faigl ve Kulich yaptıkları çalışmada çok robotlu keşfin hedef seçme stratejisinin görev paylaşırma problemi olarak formüle edildiğinde aday sınır hücre sayısının keşif

süresi üzerindeki etkisini tartışmışlardır. Robotların bir sonraki hedeflerini belirlerken ortaya çıkan sınır hücre sayısı yüzler ölçeğinde olduğunda Macar algoritması ya da gezgin satıcı problemini çözen sezgiseller için bu büyük bir hesaplama maliyetine neden olmaktadır. Bu durumu önlemek ve ayrıca mesafe maliyeti ve robotun pozisyonundan bağımsız olarak yapılacak en iyi robot-hedef atamaları keşfin verimliliğini önemli ölçüde artıracaktır. Bu noktada, doluluk ızgarasında bulunan boş hücrelerin oluşturduğu tek ve birbirine bağlı sınır kenar hücreleri tek bir bileşen olarak düşünülmekte ve aday sınır hücrelerinin üretilmesi problemi bütün bu boş bileşenleri kapsayacak noktaların bulunması olarak tekrar tanımlanmaktadır. Bu problem esasında hesaplama geometrisindeki sanat galerisi probleminin kısıtlı görülebilir bir versiyonudur ve bütün bileşenleri görece en az sayıda nokta belirlemek olarak özetlenebilir. Bahsedilen bu boş bileşenleri gören nokta her tekrarlama kalan bileşenler için yapılarak en az sayıda nokta belirlenmektedir. Yazarlar önerdikleri yaklaşımı hem Macar algoritması hem de çoklu gezgin satıcı problemini çözen sezgisellerde kullanarak en iyi atamaları yapmışlardır. Bu sayede robot takımının sık sık tekrar planlama yapmasına olanak sağlamışlardır (Faigl ve Kulich, 2013).

Nieto ve diğerleri yaptıkları çalışmada çok robotlu keşif ve haritalama için koordinasyon stratejileri üzerine odaklanmışlardır. Yazarlar, üç farklı koordinasyon stratejisi önermişlerdir. Önerdikleri stratejileri robotların yeni belirlenen hedeflere yakınlığı ve robotların yollarının kesişmesi kriterlerine göre değerlendirmişlerdir. Robotların hepsi daha önceden belirlenen bir noktadan görevlerine başlamaktadır. İlk koordinasyon stratejileri yedekli (reserve) olarak türkçeleştirilebilir. Bu stratejide herhangi bir göreve atanmayan robotlar başlangıç noktasında beklemektedirler. Diğer robotlar yeni bir ayrışma noktası, koridor gibi yapı bulduklarında bu göreve en yakın robot aktif duruma geçmekte ve bu göreve doğru yola çıkmaktadır. Bu strateji, yeni belirlenen hedeflere yakınlık kriteri açısından başarısız olmasına rağmen, robotlar birbirine uzak olduğu için robotların yollarının kesişmesi kriterine göre başarılıdır. Böl ve yönet olarak adlandırılan ikinci stratejide ise tüm robot takımı lider olarak belirledikleri bir robotun arkasından ilerlemektedirler. Bir ayrışma noktası ile karşılaşıldığında robotlar iki takıma ayrılmaktadırlar. Her ayrışma noktasında benzer durum yaşanmaktadır. Takımın birinde tek robot kaldığında, robota yeni görev atanabilmesi için robotun bir çıkmaz bölgeye girmesi ya da daha önceden keşfedilmiş bir bölgeye geri gelmesi gerekmektedir. Bir robot ya da bir takım bir çıkmaz bölgeye geldiklerinde keşif yapan en yakın takıma katılmaları gerekmektedir. Bu strateji yeni belirlenen hedeflere yakınlık kriteri açısından oldukça başarılı olmasına rağmen robotların keşfin büyük kısmında birbirlerine yakın olmasından dolayı robotların yollarının kesişmesi kriterine göre en başarısız stratejidir. Son olarak kanka (buddy) stratejisinde robotlar keşfe ikili takımlar halinde katılmaktadır. Bir ayrışma noktası ile karşılaşıldığında robotlar ikili takımlar halinde ayrılarak keşfe devam etmektedirler. Eğer bu sırada yeniden bir ayrışma noktası ile karşılaşırsa başlangıç

noktasından ikili bir robot takımı yeni hedefe gönderilmektedir. Dolayısıyla bu strateji ilk iki stratejinin birleşimi gibi çalışmaktadır ve her iki kritere göre de iyi sonuçlar üretmektedir. Önerilen stratejiler, hem benzetim ortamında hem de gerçek robotlar üzerinde test edilmiştir. Test sonuçları böl ve yönet stratejisinin en iyi çalışan strateji olduğunu göstermiştir. Kanka stratejisi böl ve yönet stratejisine çok yakın sonuçlar üretmiştir. Bununla birlikte, yazarlar robot sayısının önemli ölçüde arttığı durumlarda kanka stratejisinin böl ve yönet stratejisine üstün olacağını düşünmektedirler. Yedekli strateji ise beklendiği üzere robotların yollarının kesişmesi kriterlerini karşılayamamış ve aynı zamanda en yavaş keşif sonuçlarını üretmiştir (Nieto-Granda vd. 2014).

5.1.3 Haberleşmenin kısıtlı olduğu durumlardaki keşif yaklaşımları

Roy ve Dudek (Roy ve Dudek, 2001) yaptıkları çalışmada haberleşmenin görüş açısı ile sınırlı olduğu bilinmeyen bir ortamı keşfetmek için birbirlerinin ilk konumlarını bilmeyen robotlar kullanmışlardır. Yazarlar, robotların randevu noktaları olarak tanımlanan yerlerde buluşarak bilgi paylaşımı yapmalarını amaçlamaktadırlar. Bu aşamada, randevu noktalarını belirlemek için algılayıcıların ölçümlerini argüman olarak alan ve ayırt edicilik düzlemine haritalayan bir fonksiyon kullanılmaktadır. Bu düzlem üzerindeki uç noktalar ise randevu noktaları olarak belirlenmektedir. Ancak, her robot kendi yerel çevresini göz önüne alarak randevu noktalarını oluşturmaktadır. Robotların ortak randevu noktalarında buluşmalarını sağlamak için algılayıcı gürültüsü, harita ortaklığı ve senkronizasyon gibi faktörlerde randevu noktalarına karar vermekte kullanılmaktadır. Bu noktadan sonra, robotların belirlenen randevu noktalarını hangi stratejilerle gezeceklerine karar verilmektedir. Bu stratejilerden birinde bir robot kendi randevu noktalarından birinde beklerken diğeri kendi randevu noktalarının tamamını dolaşmaktadır. Eğer randevu gerçekleşmezse ilk robot diğeri bir randevu noktasına gitmekte ve yine diğeri robotu beklemektedir. İkinci bir stratejide robot yüksek ayırtediciliği olan randevu noktalarına ilk önce gitmektedir. Her iki robotta bu yöntemi kullandığında daha erken buluşabilmektedir. Bu şekilde üretilen stratejiler analitik ve pratik olarak incelenmektedir. Son olarak önerilen yaklaşım robot üzerinde test edilmekte ve tek robotlu sistemlere göre daha verimli çalışmaktadır. Bu çalışmada yazarların temel amacı mümkün olduğunca kısa aralıklarla robotların randevu noktalarına gelmelerini sağlayarak bilgi paylaşımını artırmaktır.

Arkin ve Diaz (Arkin ve Diaz, 2002) yaptıkları çalışmada keşif görevi boyunca robotların birbirlerinin görüş açısı (line-of-sight) içinde kalması ve haberleşmenin bozulmaması amaçlanmaktadır. Robotların görüş açılarını kontrol etmeleri için her robota ayırt edici bir renk verilmiş ve kameraları yardımıyla bu renkleri tanımaları sağlanmıştır. Bir robotun (x olsun) başka bir robotun (z olsun) görüş açısında olması için ya z direk olarak x 'i görmekte ya da z başka bir robotu (y olsun) direk olarak görüyor ve y 'de x 'i direk

olarak görmektedir. Önerilen yaklaşımda, üç farklı davranışsal keşif stratejisi sunulmaktadır. İlk stratejide robotların ortam hakkında herhangi bir bilgileri bulunmamaktadır. Robot takımının bir üyesi ilk konumunda kalarak haberleşme sunucusu olarak görev yapmaktadır. Daha sonra diğer robotlar bu sunucunun görüş açısında kalacak şekilde teker teker başıboş gezinme (wander) davranışı sergilemektedirler. Bir robot görüş açısından çıktığı zaman görüş açısına girmek için davranışını değiştirmekte ve görüş açısı iletişimi kurulduğunda yerinde kalmakta ve bir diğer robot başıboş gezinme davranışına başlamaktadır. İkinci keşif stratejisinde robotlara ortam ile ilgili düşük seviyede bilgi verilmektedir. Robotlar bu bilgi ışığında verilen hedef nesnenin konumunu belirlemeye çalışmaktadırlar. Keşif görevi başıboş gezinme davranışı ile başlamakta ancak robotların sahip olduğu bilgi sayesinde hedef nesnenin bulunabileceği konumlara doğru bir çekim oluşturarak devam etmektedir. Son keşif stratejisinde robotlara ortamla ilgili detaylı bilgi verilmektedir. Robotlar bu bilgiyi kullanarak görüş açısı kısıtını da göz önüne alarak hedef nesneye ulaşacak yol planlarını yapmaktadırlar. Ancak, verilen bu bilgilerin doğruluğu kesin olmadığı için robotların tekrar plan yapmaları gerekebilmektedir.

Howard ve diğerleri (Howard vd. 2002) yaptıkları çalışmada bilinmeyen bir ortama bir robot takımını yerleştirmek için bir yaklaşım önermişlerdir. Bu yaklaşımda robotların algılayıcılarının kapsadığı alan en çoklanmak istenirken aynı zamanda robotların en az bir robotun görüş açısı içinde kalması hedeflenmektedir. Robotlar ortama teker teker yerleştirilmekte ve her robot daha önce yerleşmiş robotlardan bilgi toplayarak yerleşeceği konumu belirlemektedir. Bu yaklaşımda robot takımının tektürel olduğu, her robotun bir mesafe algılayıcıya, haberleşme modülüne ve konumlandırma sistemine sahip olduğu varsayılmaktadır. Robotlar yerleşme süresince üç durumdan birinde bulunmaktadırlar. Bu durumlar bekleme, aktif ve yerleşmiş durumlarıdır. Başlangıçta bir robot dışında tüm robot takımı bekleme durumundadır. Bir robot ise yerleşmiş durumunda bulunmakta ve diğer robotlar için bir başlangıç noktası oluşturmaktadır. Hedeflerin seçiminde ise iki sezgisel kullanılmaktadır. Sınır sezgiseli robotların boş ve bilinmeyen alanlar (komşu hücre yaklaşımıyla aynı mantık ile) arasına yerleştirmeyi amaçlamaktadır. Kapsama sezgiseli ise robotun mümkün olan en geniş alanı kapsayacak yeri hedef olarak seçmektedir. Bu sezgisellerin belirlediği noktalar daha önceden karar verilmiş dört hedef seçme politikasına uygun olarak seçilmekte ve bu politikaların başarıları yapılan deneyler ile araştırılmaktadır. Hedeflerin atanmasında robotların düğümleri oluşturduğu bir çizge kullanılmaktadır. Hedef nokta bu çizgede boş bir düğüm olarak yaratılmakta ve sıradaki ilk robotun konumundan hedef noktasına en kısa yol bu çizge üzerinde bulunmaktadır. Ancak, bir robot başka bir robotun yolunu kapaması durumunda robot takımı tektürel olduğu için görevleri değiştirebilmektedirler. Böyle durumlarda hedefe ulaşmak amacıyla, hedef ile sıradaki ilk robot arasında kalan bütün robotların durumları aktif olarak değiştirilmekte ve görev bir su dalgası gibi yayılarak gerçekleştirilmektedir.

Vazquez ve Malcolm (Vazquez ve Malcolm, 2004) robotların bilinmeyen bir ortamı keşfederken aynı zamanda yerel bir haberleşme ağında kalmalarını sağlayan dağıtık ve davranış tabanlı bir yaklaşım önermişlerdir. Bu dağıtık yaklaşımda, robotlar her zaman en az bir robot ile haberleşme halinde kalmakta ve bu sayede hedeflerini paylaşarak koordinasyonu sağlamaktadırlar. Yaklaşımda kullanılan davranışlar: 1) Robotlar arası çarpışmayı önlemek, 2) bilinmeyen bir alanı keşfetmek ve 3) gezgin ağın içinde kalmak olarak tasarlanmıştır. Bu davranışlardan hangisinin yapılacağına mevcut ağın şartlarına göre karar verilmektedir. Bu noktada, 3 tane uzaklık $d_3 > d_2 > d_1$ tanımlanmaktadır. İki robot arasındaki mesafenin karesi d_3 uzaklığından büyük ise robotlar arası haberleşme kesilmektedir. d_2 uyarı kısıtı olarak tanımlanmakta ve genelde $d_2 = 0.9d_3$ şeklinde seçilmektedir. d_1 çarpışma kısıtı olarak belirlenmektedir. d_1 ve d_2 arasında kalan alana rahat alan adı verilmektedir. İki robot arasındaki mesafe d_2 'den büyük ise bağlantıyı sağla davranışına geçilmekte ve tekrar rahat alanı oluşturmak için hareket edilmektedir. Robot rahat alanda ya da diğer robotlardan d_1 uzaklığından daha büyük bir uzaklıkta ise bağlantı içinde kal davranışında kalınmaktadır. Eğer robotlar arasındaki mesafe d_1 uzaklığından küçükse çarpışmayı önle davranışına geçilmektedir. Yaklaşımda kullanılan keşif algoritması (Simmons vd. 2000) ile hemen hemen aynı olmakla birlikte bu yaklaşımda merkezi yerine dağıtık bir sistem kullanılmıştır. Önerilen yaklaşım çeşitli haberleşme alanları ve robot sayıları ile test edilmiş ve sonuçlar literatürdeki sonuçlar ile uyumlu çıkmıştır.

Rekleitis ve diğerleri (Rekleitis vd. 2004) yaptıkları çalışmada haberleşmenin kısıtlı olduğu ortamlarda kapsama problemini incelemektedirler. Yazarlar, aynı alanı birden fazla kere kapsanmasını önlemek amacıyla robotların birbirine yakın hareket etmeleri gerektiğini savunmaktadırlar. Önerilen çalışmada ortam hücrelere bölünmektedir. Hücrelerin üst ve alt sınırları aynı hıza sahip iki robot tarafından kapsanmaktadır. Bu iki robot aynı zamanda birbirlerinin görüş açısı içinde kalarak hareket etmektedirler. Bu iki robotun görüş açısı arasına bir engel girdiğinde bir kritik noktaya ulaşıldığı ve hücrenin bittiğine karar verilmektedir. Bu noktada, engelin konumu ve geometrisine göre robot takımı ya iki alt takıma ayrılmakta ya da tüm takım olarak yeni hücreyi keşfetmektedir.

Powers ve Balch (Powers, Balch, vd. 2004) yaptıkları çalışmada robot takımındaki üyeler arasındaki görüş açısı haberleşmesini koruyacak davranış tabanlı bir yaklaşım önermişlerdir. Bu davranış tabanlı yaklaşım, takım üyelerinin paylaştıkları konum bilgilerini, ölçülen haberleşme sinyal kalitesi ve harita tabanlı haberleşme sinyal kalitesi tahminlerini kullanarak hareket vektörleri hesaplamaktadır. Hesaplanan bu hareket vektörleri robotun seyrüsefer yaparken takım arkadaşları ile arasındaki haberleşmenin yok olmasını engellemektedir. Bu yaklaşımda, ilk olarak her robotun mevcut sinyal kalitesi ölçülmektedir. Daha sonra, robotun yapabileceği her hareket için tahmini bir sinyal kalitesi

hesaplanmaktadır. Robotun bir sonraki adımda yapabileceği her hareket az önce hesaplanan tahmine göre ölçeklendirilmiş bir vektör ile gösteriliyor. Son olarak, bu vektörlerin toplamı yapılacak hareketi belirlemektedir.

Sheng ve diğerleri (Sheng vd. 2006) tamamen dağıtık ve teklif tabanlı bir yaklaşım önermişlerdir. Ortamın gösteriminde ızgara tabanlı bir yaklaşım kullanılmış ve keşifte kullanılacak hedefler için komşu hücre yöntemi seçilmiştir. Önerilen yaklaşımda robotlar birbirlerine yakın konumlardan göreve başlamakta ve birbirlerinin konumlarını bilmektedirler. Herhangi bir anda bir robot: 1) algılama ve haritalama, 2) teklif ve 3) gezinme durumlarından birinde bulunmaktadır. Robot yeni bir bilgi edindiğinde ya da alt ağdaki robotlardan biri yeni bir bilgi duyurduğunda yerel haritasını güncellemektedir. Aynı zamanda bu yeni haritasını alt ağdaki robotlara duyurmaktadır. Bu noktada, hedef hücreler belirlenerek algılama ve haritalama durumu tamamlanmaktadır. Robot daha sonra alt ağında bir teklif süreci olup olmadığını kontrol etmektedir. Eğer bir teklif süreci varsa robot yerel haritasını göndererek bu sürece dahil olmaktadır. Tekliflerin hesaplanmasında üç temel unsur kullanılmaktadır. Bilgi kazancı hesaplanırken, algılama ve haritalama yapan robotların algılama alanı ya da gezinme durumundaki robotların hedef hücreleri içinde olmayan bilinmeyen hücre sayısı kullanılmaktadır. Maliyet hesabında robot ve hedef hücre arasındaki en kısa mesafe kullanılmaktadır. Bunlara ek olarak, haberleşmenin kısıtlı olduğu durumları göz önüne almak için yakınlık ölçüsü olarak tanımlanan bir kriter kullanılmaktadır. Bu kriter sayesinde robotlar birbirlerinden ayrılmamakta ve sürekli haberleşme durumunda oldukları için aynı yerin birden fazla kere keşfedilmesi önlenmektedir. Her robot hesapladığı teklifi diğer robotlara duyurmakta ve belirli bir süre beklemektedirler. Eğer bekleme süresince başka robotlar haritalarını yollamış iseler teklifler güncel haritaya göre tekrar hesaplanmaktadır. Süreç boyunca başka bir katılımcı ya da daha iyi bir teklif yoksa robot görevi kazanmakta ve hedefe doğru gezinme durumuna geçmektedir. Diğer robotlar ise kazanan açıklandıktan sonra bu yeni duruma göre tekliflerini güncelleyerek yeni teklif süreçlerine katılmaktadır.

Rooker ve Birk (Rooker ve Birk, 2007) yaptıkları çalışmada bilinmeyen bir alanı bir robot takımı ile keşfederken bir grup robotun tüm işlem boyunca birbirleri ile haberleşebilir olmasını amaçlamaktadırlar. Bu noktada, robotlar gruplar halinde keşif yaparken haberleşmenin kopmaması birbirlerinden için ayrılamamaktadırlar. Bu durum tüm robotların haritalarını güncellemediklerinde ya da robot gruplarındaki hareket sınırlı olduğunda ortaya çıkmakta ve kilit durumu olarak adlandırılmaktadır. Bir kilit durumu ile karşılaşıldığında robot gruplarından bir robot buluşma noktası olarak seçilmekte ve olduğu yerde beklemekte diğer robotlar ise keşif görevlerini bırakarak bu robotun haberleşme alanına girene kadar ona doğru hareket etmektedirler. Robot grubunun tamamı tekrar haberleşme alanına girdiklerinde tekrar keşif görevlerine dönmektedirler. Bu noktada, bir

robotun seçilmesi yerine robotların bir komşu hücrede buluşması fikri öne sürülmektedir. Ancak, bu fikir beklenen ilerlemeyi sağlamamakta çünkü bu durumlar oluşmaya başladığında alanın büyük bir kısmı keşfedilmiştir.

Visser ve Slamet (Visser ve Slamet, 2008a) önerdikleri yaklaşımda bilgi kazancındaki haberleşme olasılığını tahmin etmek için sisteme bir ölçüt eklenmektedir. Tek robot için bu olasılık komuta merkezi ile robot arasındaki mesafe kullanılarak hesaplanabilirken çok robotlu sistemlerde diğer robotların konumları da bu ölçütün içine katılabilmektedir. Ortam ile ilgili bilgiyi en çok artıracak olan bir sonraki keşif hareketi planlanırken aynı zamanda bu bilginin komuta merkezine ulaşması beklenmektedir. Ortamın gösteriminde doluluk ızgarası kullanılmaktadır. Komşu hücrelerin belirlenmesinde ise güvenli bölgeler olarak tanımlanan yöntemden yararlanılmaktadır. Bu güvenli bölgelerin sınırları komşu hücreler olarak seçilmektedir. Bir komşu hücre için fayda fonksiyonu komşu hücrenin bulunduğu keşfedilmemiş alan ile komşu hücrede haberleşme olasılığı çarpımının robotun komşu hücreye olan uzaklığına oranı olarak tanımlanmaktadır. Komşu hücreler ve faydaları bahsedildiği gibi hesaplandıktan sonra bütün robot-hedef hücre atamaları için bir fayda matrisi oluşturulmaktadır. Bu matristeki faydalar hesaplanırken kartezyen uzaklık kullanılmaktadır. Tüm matris oluşturulduktan sonra en yüksek faydalı atama için bir yol planı oluşturulmaktadır. Bu yol planı sonucunda çıkan fayda ile mevcut fayda aynı ise bu atama yapılmakta ve ilgili satır ve sütunlar matristen çıkarılmaktadır. Aksi takdirde bir sonraki en yüksek fayda için aynı süreç işletilmektedir. Bu süreç tüm robotlara komşu hücre atanana kadar devam etmektedir. Robotun konumundan komşu hücreye gidecek yol planı yapılırken engeller göz önüne alınmakta ve seviye öncelikli arama (breadth-first search) algoritması kullanılmaktadır. Önerilen yaklaşım P3-AT robotları ile USARSIM ortamında gerçekleştirilmektedir.

Mosteo ve diğerleri (Mosteo vd. 2008) yaptıkları çalışmada rotalama problemini haberleşmenin kısıtlı olduğu ortamlarda incelemektedirler. Önerilen çalışmada, yazarlar, belirlenmiş bir eniyileme kriteri altında tüm hedeflerin ziyaret edileceği ve her hedefin sadece bir robot tarafından ziyaret edildiği durumlarda robotların izleyecekleri rotaları belirlemeyi amaçlamaktadırlar. Bununla beraber, robot takımındaki herhangi bir üye herhangi başka bir üyeye doğrudan ya da başka robotlar üzerinden haberleşebilir olması gerekmektedir. Bunu sağlamak amacıyla, robotlar arası sinyal kalitesi sanal bir yayın fiziksel kuvvet modeli olarak modellenmektedir. Ayrıca robotlar ise bir en küçük kapsayan ağacın düğümleri olarak temsil edilmektedir. Bu ağaçtaki herhangi bir ayırıt üzerindeki sinyal kalitesi belirli bir seviyenin altına düştüğünde robotların tekrar güvenilir bir sinyal seviyesine kavuşmalarını sağlayan güçler üretilmekte ve bunlar hıza dönüştürülerek robotlara uygulanmaktadır. Görev paylaşırma stratejisinin temelini görevleri bir robot kümesine ve her robot kümesine tek görev atanmaktadır. Görevin tamamlanması sırasında

robotlar arası haberleşme ağının kopmaması için robot kümesine yeni robotlar katılabilmektedir. Bu temel strateji altında görevlerin robot kümelerine atanması seviyesinde fırsatçı, TSP tabanlı, saat ve müzayede tabanlı yaklaşımlar kullanılmıştır. Fırsatçı yaklaşımda herhangi bir robota en yakın olan görev bu robot kümesine atanmaktadır. TSP tabanlı yaklaşımda tüm görevler tek robotlu küresel bir çözüm ile hesaplanmakta ve daha sonra bu çözümdeki görevler o andaki mevcut robot kümelerine atanmaktadır. Saat yaklaşımında görevler polar koordinat sistemine göre sıralanmakta ve en yakın robotlara atama yapılmaktadır. Dolayısıyla robotlar bir saat gibi hedefleri süpürerek ilerlemektedir. Son olarak, müzayede tabanlı yaklaşımda her görevi yapmak için gerekli olan robot sayısı başlangıçta hesaplanmakta ve tek robotla yapılacak görevlerden başlanarak robot takımı boyutundaki görevlere kadar müzayedeler yapılarak görevler dağıtılmaktadır. Önerilen yaklaşım, tektürel robot takımı ile farklı haberleşme menzillerinde test edilmiştir. Elde edilen sonuçlarda, TSP tabanlı ve saat yaklaşımlarının haberleşme menzillerinden diğer iki yaklaşım kadar kötü etkilenmedikleri gözlemlenmiştir.

De Hoog ve diğerleri (De Hoog vd. 2009) yaptıkları çalışmada komşu hücre tabanlı keşif algoritmasını haberleşmenin kısıtlı olduğu afet bölgelerinde kullanmak için genişletmişlerdir. Bununla birlikte, robotların elde ettikleri bilgiler mümkün olduğunca kısa sürede kontrol merkezine ulaştırılmalıdır. Bu yaklaşımda robot takımının her üyesi aşağıda detaylı olarak anlatılan rollerden birini üstlenmektedir:

- **Kaşif (*Explorer*):** Kaşifler ortamı en uç noktalarına kadar keşfetmekle görevlidirler. Elde ettikleri bilgileri periyodik olarak daha önceden belirlenmiş randevu noktalarına gelerek ulaklara aktarırlar.
- **Ulak (*Relay*):** Ulaklar kaşiflerden aldıkları bilgileri komuta merkezine ulaştırmak ile görevlidirler. Bunu yapmak için daha önceden belirlenmiş randevu noktalarında kaşifler ile buluşmakta ve kaşif ile kendi elindeki tüm bilgileri değiştirerek komuta merkezine dönmektedirler.

Bu çalışmada, robotlara rolleri önceden atanmıştır. Bir ulak takip ettiği kaşifin haberleşme alanına girdiğinde, kaşif mevcut konumunu bir sonraki buluşma yeri olarak belirlemekte ve keşfe son vererek ulak ile buluşmak üzere daha önceden kararlaştırılmış olan randevu yerine gitmektedir. Kaşif, bir sonraki randevu yerini ve ulaşın yapacağı hareketi bildiği için bir sonraki randevuya gitmek için ne kadar süresi olduğunu hesaplayabilmektedir. Sonuç olarak, randevu noktaları robotun mevcut konumu olarak seçildiğinden her buluşma ortamının daha derin kısımlarında gerçekleşmekte ve keşfedilen alanların yüzdesi artmaktadır. Önerilen yaklaşım, 4 farklı komşu hücre yöntemi ve

haberleşme için 3 farklı ortamda test edilmiştir. Test sonuçlarında rol tabanlı yaklaşımın tüm alanı keşfettiği ama haberleşme kısıtı olmayan komşu hücre yaklaşımına göre daha yavaş bir yaklaşım olduğu gözlemlenmiştir. Bununla birlikte, kontrol merkezindeki bilgi rol tabanlı yaklaşımda daha yüksek olmaktadır.

De Hoog'un 2010 yılındaki (Hoog vd. 2010b) yayınında takım düzeni ağaç yapısı ile gösterilmektedir. Ağacın kök düğümünde komuta merkezi, ağacın yapraklarında ise kaşifler yer almaktadır. Komuta merkezi ile kaşifler arasında birden fazla ulak olabilmektedir. Bir ulak ve bir kaşif buluştuklarında ortamla ilgili tüm bilgilerini birbirlerine aktarabilmektedir. Dolayısıyla, o anda her ikisi de aynı harita ve bilgiye sahip olmaktadır. Önerilen yaklaşımın başarılı olabilmesi için seçilecek olan randevu noktaları çok önemli bir konuma gelmektedir. Bu noktada yazarlar ilk önce ortamın haritasını inceltme (thinning) gibi yöntemler ile iskelet biçimine getirmektedirler. Bu iskeletin çıkarılmasında Voronoi gibi yöntemlerde kullanılabilir. Daha sonra birleşme noktaları (3 ya da daha fazla kenarın birleştiği düğümler) bulunmaktadır. Bu birleşme noktalarından uygun olanları randevu noktaları olarak belirlenmektedir. Bu noktada, elde bulunan olası randevu noktalarından hangisinin en iyi olduğuna karar vermek gerekmektedir. Bir çok faktörün göz önüne alındığı değerlendirmeler sonucunda kaşifin bir sonraki sınır seçiminin hemen yanında ve en önemli (en çok kenarın birleştiği) birleşme noktası randevu noktası olarak seçilmektedir. Yazarlar son olarak ulak veya kaşifin randevu noktasına gelemediği (duvarın yıkılması, moloz vs) durumlarda oluşabilecek senaryoları incelemekte ve sonunda bir noktada ulak ve kaşifin bir araya gelerek bilgileri değiştireceklerini söylemektedirler.

De Hoog'un 2010 yılındaki (Hoog vd. 2010a) yayınında rol tabanlı keşif algoritması dinamik takım yapısı için genişletilmiştir. Bu yaklaşımda, herhangi bir anda ulak gidilecek olan sınıra kaşifden daha yakınsa roller değişmektedir. Burada rol değişimini tetikleyen unsur daha kısa yol katedilmesidir. Dolayısıyla, robot takımının tektürel olduğunda (tüm robotların hızları aynı) bu durum daha hızlı keşif olarak yorumlanabilmektedir. Uygulamada roller değiştirilirken bir mesaj kullanılmaktadır ve mesajın içerisinde ID, rol, durum, sınır listesi, çocukların ID'leri, ebeveyn ID'leri, çocuk randevuları, ebeveyn randevuları ve mevcut hedef yer almaktadır. Böylece ağaç yapısının herhangi bir yerinde olan iki robot yapıyı bozmadan rolleri değiştirebilmektedir. Deney sonuçları incelendiğinde dinamik rol tabanlı keşif algoritmasının diğer yaklaşımlardan daha hızlı keşif yaptığı (robot takımının sadece yarısı keşif ile ilgilenmesine rağmen) ortaya çıkmıştır. Ayrıca komuta merkezindeki ve robotlar arasındaki farkındalığın arttığı gözlemlenmektedir. Yazarlar bu durumun afet bölgelerinde çok önemli olduğuna vurgu yapmaktadırlar.

Doniec ve diğerleri çok robotlu keşif probleminde haberleşme kısıtlarını ortadan kaldırmak için dağıtık kısıt sağlama problemini ele almışlardır. Bu problem sonlu sayıdaki

değişkenlerin belirlenen kısıtları sağlayarak alabilecekleri değerlerin hangi aralıkta olduğu ile ilgilenmektedir. Ayrıca, robotun sadece sekiz yönde hareket edebileceği varsayımı yapılmaktadır. Dolayısıyla, bir sonraki hareket üç kısıt tamamen sağlamalıdır: 1) Bir sonraki nokta robotun ağ ile bağlantısını koparmamalıdır. 2) Bir sonraki nokta başka robotların algılayıcı menzilleri ile çakışmamalıdır. 3) Bir sonraki nokta robotun yeni keşfedilmemiş alanlar bulmasını sağlamalıdır. Her robot bir sonraki hedef noktasını belirlerken dağıtık kısıt sağlama problemini verilen üç kısıt ile çözmektedir. Sonuç olarak robotların haberleşme menzili içinde kalarak mümkün olduğunca dağılımları amaçlanmıştır. Önerilen yaklaşım, benzetim ortamında test edilmiştir. Elde edilen sonuçlar, robot takımının boyutunun artması ile keşif süresinin önemli ölçüde azaldığını bununla beraber haberleşme yükünü artırmadan gerçekleştiğini göstermektedir (Doniec vd. 2009).

Pei ve diğerleri yaptıkları çalışmada gerçek zamanlı ve tekrarlamalı bir keşif yöntemi önermişlerdir. Bu yöntemde robotların elde ettikleri bilgileri aktardığı bir komuta merkezi bulunmaktadır. Robotlar birbirleri ve bu komuta merkezi ile her an bağlantı içinde olmalıdır. Bunu başarabilmek için robotlar keşif ve ulak robotu olarak görev yapmaktadırlar. Keşif robotları bilinmeyen ortamlara doğru ilerlerken ulak robotları ise bütün robotların ve komuta merkezinin haberleşebilmesi için gerekli olan pozisyonu almaktadır. Böylece keşif robotlarının elde ettiği bilgiler her an komuta merkezine aktarılabilen ve keşif koordinasyonu komuta merkezi tarafından yapılabilmektedir. Robot takımında hangi robotun keşif ve hangi robotun ulak robotu olacağına tekrarlamalı bir şekilde karar verilmektedir. Öncelikle, keşif robotlarına karar verilmektedir. Daha sonra kalan ulak robotları ile tüm robotların ve komuta merkezinin birbirleri ile haberleşebildikleri kontrol edilmektedir. Eğer haberleşme sağlanamazsa keşif robotlarından biri ulak robotu yapılmakta ve robot takımı tekrar alana yerleştirilmektedir. Bu süreç bütün robotların birbirleri ile ve komuta merkezi ile haberleşebildikleri düzeni elde edene kadar devam etmektedir. Önerilen yöntem, geçmiş çalışmalarda yer alan yöntemler ile karşılaştırılmıştır. Elde edilen sonuçlar, önerilen yöntem ile keşif süresinin önemli ölçüde kısaldığını bununla birlikte haberleşme kalitesinin arttığını göstermiştir (Pei vd. 2010).

Wellman ve diğerleri yaptıkları çalışmada robotları arama sektörleri adını verdikleri bölgelere dağıtmakta ve belirli zamanlarda robotları bir araya getirerek bilgilerini paylaşmalarını sağlamaktadır. Dolayısıyla, robotlar keşif süresinin tamamında haberleşmeye ihtiyaç duymamaktadır. Bu yaklaşımda, robotlar birbirlerine yakın konumlardan keşife başlamaktadırlar. Ayrıca, kullanılan robot takımının bütün üyeleri aynı yeteneklere sahiptir. Robotlar başlangıç noktasında arama sektörlerini oluşturmaktadır. Bunu yapabilmek için robotların önündeki yarı çemberler kullanılmaktadır. Örneğin, robot takımı üç robottan oluşuyorsa yarı çember üç eşit parçaya bölünecek ve robotlar

birbirlerinden altmış derece ile ayrılacaklardır. Bu çalışmada, robotların buluşup bilgilerini paylaşacakları randevu noktaları yerel haritalarının merkezleri olarak seçilmiştir. Robotlar randevu noktasına ulaştığında bilgilerini paylaşacaklar ve her ikisinde yerel haritası aynı olacak dolayısıyla bir sonraki randevu noktası olarak da aynı yeri kendiliğinden belirleyecekler. Son olarak, robotların ne kadar zamanda bir randevu noktasına gidecekleri ve bu noktada ne kadar süre bekleyecekleri konuları tartışılmıştır. Önerilen yöntem, iki parametre seti için haberleşmenin mükemmel olduğu, belli bir menzilde haberleşme olduğu ve haberleşmenin olmadığı durumlar için test edilmiştir. Elde edilen sonuçlar, belli bir menzilde haberleşme olduğu durumda robotların bütün haritalarını paylaştıkları parametre seti için en iyi performansa sahip olduğunu göstermektedir. Önerilen yaklaşım son olarak rol tabanlı keşif yöntemi ile karşılaştırılmış ve bütün parametre setleri için daha iyi sonuçlar ürettiği gözlemlenmiştir (Wellman vd. 2011).

Bir başka çalışmada da randevu tabanlı bir keşif yaklaşımı önerilmiştir (Meghjani ve Dudek, 2011). Bu çalışmada yazarlar randevu noktalarının seçimi üzerine yoğunlaşmışlardır. İlk olarak, kısmi olarak keşfedilmiş alanın inceltme yöntemi ile iskeletini çıkarmışlardır. Bu iskeleti yönsüz bir çizge ile ifade etmişler ve muhtemel randevu noktalarını bu çizge üzerinde belirlemişlerdir. Bu noktada randevu noktalarını sıralamak için üç kriter kullanmışlardır: alan, doğrusal uzaklık ve ağırlıklı uzaklık. Alan kriterinde robotun randevu noktasına gittiğinde algılayıcıları tarafından kapsanacak alan kullanılmaktadır. Doğrusal ve ağırlıklı uzaklık kriterlerinde alan bu uzaklıklara bölünmektedir. Yazarlar, robotların randevu noktalarında nasıl davranacakları konusunda üç farklı randevu stratejisi tanımlamışlardır. Bunlardan ilkinde asimetrik sıralı adımı vermişlerdir. Bu stratejide robotlardan biri dururken diğeri ise duran robotu aramaktadır. Simetrik sıralı stratejide ise iki robotta birbirlerini aramaktadırlar. Son olarak, üstel yaklaşımda ise üstel olasılık yoğunluk fonksiyonu kullanılmaktadır. Önerilen yaklaşım üç farklı ortamda test edilmiştir. Doğrusal uzaklık kriteri üç farklı strateji ve üç farklı ortam için ağırlıklı uzaklık ve alan kriterlerine göre daha kısa sürede ortamların keşfini tamamlamıştır. Bununla birlikte, randevu stratejilerinin performansı ortamın yapısına göre değişiklik göstermektedir.

Matignon ve diğerleri yaptıkları çalışmada dağıtık Markov karar sürecini kullanarak çok robotlu keşif problemine çözüm aramışlardır. Her robot dağıtık değer fonksiyonu adı verilen kriter ile bir sonraki hedefini seçmektedir. Bu noktada, robotlar kendilerinin ve takım arkadaşlarının konumlarını bilmekte ve yerel haritalarını birleştirebilmektedir. Bu sayede ortama dağılarak, en az sayıda iletişim ile ortamın keşfini tamamlamaktadırlar. Robotlar bir şekilde dar bir yerde karşılaşılırsa yerel koordinasyon devreye girmekte ve robotları farklı bölgelere yöneltmektedir. Önerilen yöntem, hem benzetim ortamında hem de gerçek robotlar ile test edilmiştir. Yazarlar, ortamın büyüklüğüne bağlı olarak uygun

sayıda robot kullanıldığında robotların birbirleri ile hiç karşılaşmadan keşif görevlerini tamamladığını göstermiştir. Bununla birlikte, haberleşme kısıtları yalnızca yöntemin verimliliğini etkilemektedir. Haberleşmenin olmadığı ya da kısıtlı olduğu durumlarda robot takımı görevlerini tamamlamaktadır (Matignon vd. 2012).

Visser ve diğerleri (Hoog vd. 2010a) yayınında önerdikleri rol tabanlı keşif yaklaşımını diğer keşif yaklaşımları ile karşılaştırmış; olumlu ve olumsuz yönlerini tartışmışlardır. İlk olarak, yazarların amacı haberleşmenin kısıtlı olduğu ortamları hızla keşfetmekle beraber komuta merkezi adını verdikleri konuma hızlı ve güvenilir biçimde bilgileri ulaştırmak ve bu noktadan verilen komutları da hızlı ve güvenilir biçimde robot takımına göndermektir. Rol tabanlı keşif ile, bir ulağın birden fazla keşif robotuna hizmet vermesi gerektiği durumlarda hızla organize olabilir. Aynı zamanda ulak ve keşif robotları dinamik olarak rol değiştirebildiği için dinamik durumlara hızlıca cevap verebilmekte ve çoktörel robot takımları için uyarlanması kolay olmaktadır. Sonuç olarak, robot takımının en yüksek önceliğinin komuta merkezine düzenli ve güvenilir bilgi göndermek olduğu; haberleşme menziline ortamın boyutuna göre küçük olduğu; robot takımının boyutunun büyük olduğu durumlarda rol tabanlı keşifi kullanmak uygun olacaktır. Bununla birlikte, hızlı keşif yapmanın öncelikli olduğu; robotların belirlenen bir noktaya dönmelerinin ya da ortamsal koşullardan dolayı tekrar eski izlerini kullanmamaları durumunda rol tabanlı keşif uygun olmamaktadır (Visser vd. 2013).

Tuna ve diğerleri yaptıkları çalışmada sınır hücre, rol ve market tabanlı keşif stratejilerini üç farklı haberleşme modeli ile birlikte gerçekleştirmektedir. Haberleşme modelleri statik çember, görüş alanı ve yayılma olarak belirlenmiştir. Statik çember haberleşme modelinde, bir robotun haberleşme menziline içinde olan diğer robot ile aralarında engel olup olmamasına bakılmaksızın haberleşilebilmektedir. Görüş alanı haberleşme modelinde robotların görüş alanlarının arasında herhangi bir engel olmaması gerekmektedir. Yayılma haberleşme modeli en gerçekçi modeldir. Bu modelde engellerin kalınlığı haberleşme menziline karar verilirken hesaba katılmaktadır. Bu üç haberleşme modeli ve üç keşif stratejisi üç farklı ortamda test edilmiştir. Elde edilen sonuçlar sınır hücre keşif stratejisinin en hızlı keşifi sağladığı rol tabanlı stratejinin ise en yavaş olduğunu göstermektedir. Bu testlerde en gerçekçi model olduğu için yayılma haberleşme modeli kullanılmış ve farklı sayılardaki robot takımları için testler tekrarlanmıştır. Robot sayısının artması ile beraber keşif süresi seçilen stratejiye göre yüzde 15 ile 25 arasında azalmıştır (Tuna vd. 2013).

Jensen ve diğerleri robot takımını haberleşmenin menziline kadar ortama dağıtmakta; daha sonra ise onları bu haberleşme menzili içinde tutarak keşif yapmaktadırlar. Önerdikleri yaklaşımda, robotların üzerlerinde ortama bırakabilecekleri

işaretçiler bulunmaktadır. Robotlar diğer yaklaşımlardan farklı olarak sadece durumlarındaki son güncellemeleri değil yaptıkları hareketleri de diğer robotlar ile paylaşmakta ve böylece birbirlerine yakın kalmaktadırlar. Robotlar ve işaretçilerin durumları yedi mesaj ile ifade edilebilmekte ve bu da üç bitlik bir haberleşme yükü anlamına gelmektedir. Robotlar aynı başlangıç noktasından keşfe başlamaktadırlar. Bir robot haberleşme menziline sınırına geldiğinde başka bir robotu yanına çağırılmaktadır. Bu süreç bütün robotların keşfe başlamasına kadar sürmekte ve robotlar ortama haberleşme menziline sınırları ile doğru orantılı olarak dağılmaktadır. Bu durumda, başlangıç noktasına en yakın robot bulunduğu noktaya bir işaret bırakarak en uca gelmektedir. Dolayısıyla, ortamın bir bölümü bu şekilde keşfedilip bittikten sonra robotlar bu bölümü keşfedilmiş olarak işaretliyerek ortamın diğer kısımlarına geçmektedirler. Keşif tamamlandığında, robotlar arkalarında bıraktıkları işaretçileri kullanarak başlangıç noktasına dönmektedirler. Önerilen yöntem, yazarların daha önceden önerdikleri başka bir yöntem ile karşılaştırılmış ve keşif süresinin 1.5 kat azaldığı gözlemlenmiştir (Jensen vd. 2014).

Bu alt bölümde verilen geçmiş çalışmalar göz önüne alındığında çok robotlu sistemlerde koordinasyon ve keşif yaklaşımları davranış tabanlı yaklaşımlardan, tek öğeli ve kombinasyonel müzayede yöntemlerine, çoklu gezgin satıcı problemine çözüm olan sezgiselleri kullanan yaklaşımlardan, k-ortalama ile robotları ortama dağıtmayı amaçlayan yaklaşımlara kadar oldukça geniş bir yelpazede yer almaktadırlar. Ayrıca, keşif sırasında en iyi görüş yönteminden, sanat galerisi probleminin değiştirilmesine, ortamın bölütlenmesinden geçmiş çalışmaların çok büyük bir kısmında bulunan sınır hücre yaklaşımına kadar bir çok farklı hedef belirleme stratejisi de geçmiş çalışmalarda yer almaktadır. Son olarak, rol tabanlı, görüş açısı tabanlı, randevu tabanlı, robotları bir arada tutmaya çalışan ve belirli bir menzilde haberleşebilen keşif yaklaşımları da geçmiş çalışmalarda yer almaktadır. Bu çok geniş geçmiş çalışmalar incelenip, tez çalışmasının kapsamı da gözden geçirildikten sonra robotların küresel koordinasyonu ve robotlar arası görev paylaşımı için market tabanlı bir yaklaşımın kullanılmasının uygun olacağı düşünülmüştür. Bunun sebebi, geçmiş çalışmalarda da belirtildiği üzere zaman karmaşıklığı açısından en verimli küresel koordinasyon yaklaşımlarından biri olmasıdır. Bununla birlikte, sağladığı esnek yapı ile görevlerde ya da robot takımında yapılacak olan değişiklikler hızlıca sisteme eklenebilir. Son olarak, sadece teklif, ödül ya da görev gibi az sayıda ve boyutta mesaj kullandığından hem bant genişliği olarak hem de haberleşme yükü olarak uygun çözümler sunmaktadır. Robotların hedeflerini belirlemede geçmiş çalışmaların büyük çoğunluğunda yer alan sınır hücre yöntemi kullanılmıştır. Bunun sebebi ise, tez kapsamında hedeflerin belirlenmesi değil hangi kriter ile robotların hedeflere atanacağı üzerinde durulmak istenmesidir. Dolayısıyla, hedef belirleme yöntemlerinin verimliliğinin karşılaştırılması bu tez çalışmasının kapsamında yer almamaktadır.

Haberleşme göz önüne alındığında ise robotların sadece belirli bir menzilde olduklarında haberleşebilmelerinin, diğer robotların görevleri için teklif verebilmeleri ve haritalarını diğer robotlardan gelen haritalar ile paylaşmaları, uygun olacağı düşünülmüştür. Afet ortamları göz önüne alındığında en büyük öncelik keşfin tamamlanması olduğu ve rol tabanlı ya da randevu tabanlı yaklaşımlar kaynaklarının bir kısmını (zaman ya da enerji) diğer robotlar ile haberleşebilmek için harcadığından bu problem için uygun değildir. Robotları haberleşsinsinler diye birbirlerine yakın tutmakta benzer şekilde keşfi yavaşlatacaktır. Bu durumda, robotların belirli bir menzil içinde haberleşmeleri uygun olacaktır. Dolayısıyla, robotlar yaptıkları seçimler ile en başta ortama yayılıp daha sonra bir noktada buluşabilirlerse en verimli haberleşme yöntemi bu şekilde olacaktır. Son olarak, sınır hücreleri kümeleyerek kombinasyonel müzayedede yöntemi ile ya da gezgin satıcı problemini çözen sezgiseller kullanarak küresel bir koordinasyon sağlanabilir. Bu noktada, kullanılan sezgisellerin zaman karmaşıklığının analiz edilmesi market tabanlı yaklaşım ile karşılaştırılması yapılabilir. Diğer yandan yayılma haberleşme modeli olarak adlandırılan ve engellerin kalınlıklarını da haberleşme menziline hesaplarken hesaba katan daha gerçekçi bir haberleşme modelinin kullanılması da uygun olabilir. Bu alt bölümden sonra gerçekleştirilen yaklaşım ve deneysel sonuçlar detaylı olarak verilecektir.

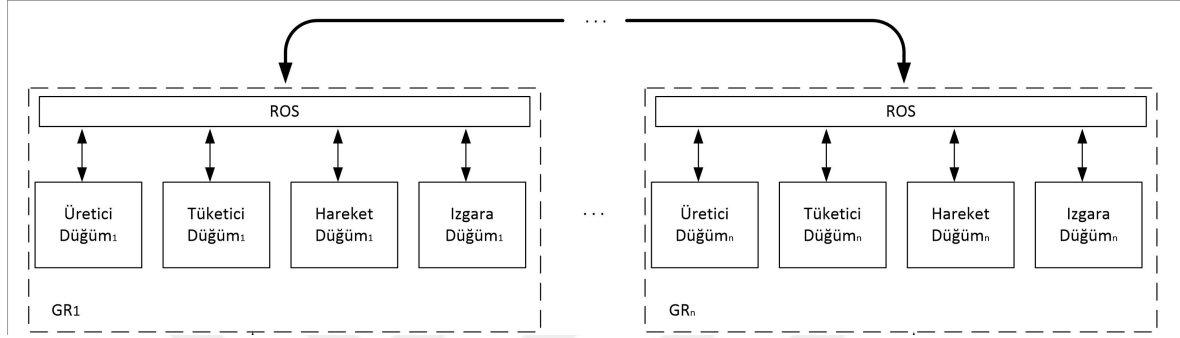
5.2 Gerçeklenen Yaklaşım

5.2.1 Market tabanlı koordinasyon ve görev paylaşırma

Stentz ve Dias market tabanlı yaklaşımlarla ilgili ilk tanımlamaları (Stentz ve Dias, 1999) yayınında yapmışlardır. Bu çalışmada, robotların markette nasıl konumlanacağı detaylı olarak anlatılmıştır. Diğer robotlardan kendisine servis vermesini isteyen bir robot (Üretici) bu görev için bir fiyat belirlemektedir. Görevler için fiyatların nasıl belirleneceği ve bu fiyatlar sayesinde marketin yöneliminin nasıl olacağı gibi konular örneklerle yayında verilmektedir. Bununla birlikte, servis verecek olan robot (Tüketici) bu görev için ne kadar masraf yapacağını belirlemektedir. Bu masrafın diğer bir deyişle maliyetin nasıl belirleneceği ve marketin bu maliyet karşısında nasıl tepki vereceği gibi konular detaylı olarak anlatılmaktadır. Dolayısıyla, hem görevleri sunanlar açısından hem de görevi yapacaklar açısından sırasıyla fiyat ve maliyetlerin belirlenmesi robotlar arası koordinasyon ve görev paylaşırma probleminin çözümünü önemli ölçüde etkilemektedir. Ayrıca, üretici ve tüketici arasındaki alış veriş sürecinin nasıl gerçekleştiği ve sonlandırıldığı da önem taşımaktadır. Bu süreçte çok çeşitli müzayedeler ya da oylama gibi yöntemler kullanılabilir. Benzer şekilde bu sürecin sonlandırılmasında da farklı farklı yöntemler kullanılmaktadır.

Bu tez çalışması kapsamında market tabanlı yaklaşımın alış veriş sürecinde çok-öğeli ve tek turlu bir müzayedede yöntemi kullanılmaktadır. Diğer bir deyişle, üretici robot birden

fazla görevi bir anda markete sunabilmekte ve bu görevler için teklifleri toplayabilmektedir. Her tüketici bir anda bir görev alabilmekte ve mevcut görevi sonlanana kadar başka bir görev için teklif verememektedir. Dolayısıyla, çok turlu müzayede kullanmanın bir anlamı yoktur. Son olarak, teklifler değerlendirilerek sonuçlar açıklanmakta ve süreç tamamlanmaktadır. Önerilen yaklaşımda, temel olarak tek robot sınır hücre keşif yaklaşımı kullanılmıştır. Bu yaklaşım, çok robotlu sistemler için genişletilmiştir. Sistem mimarisi Şekil 5.1'de verilmiştir.



Şekil 5.1: Çok Robotlu Sistem Mimarisi

Keşifte çok robotlu yöntemin kullanılabilmesi için sistem mimarisinde bulunan erkinlerde bazı düzenlemeler yapılmıştır. İlk olarak, ızgara erkinin metrik ve topolojik haritayı oluşturmak, güncellemek, arayüzde göstermek ve istenildiğinde diğer erkinlerle paylaşmak ile sorumludur. Bunlara ek olarak, eğer robotlar haberleşme menziline içinde ise, diğer robotlardan gelen topolojik ve metrik haritaları kendi haritaları ile birleştirmekte bu erkinin sorumluluğundadır.

Üretici erkin belirli aralıklarla yapılması gereken görevleri belirlemekte, bu görevleri fiyatlandırmakta ve robot için en faydalı görevi seçmektedir. Çok robotlu yöntemde eğer robotlar haberleşme menzili içindelerse üretici erkin kendi tüketici erkininden teklif alabildiği gibi diğer robotların tüketici erkinlerinden gelen teklifleri de toplamaktadır. Daha sonra, hangi robotun hangi görevi hangi teklif ile kazandığını haberleşme menziline bulunan bütün tüketici erkinlere bildirmektedir.

Tüketici erkin, belirlenen görevler için maliyet hesaplamakta ve üretici erkine teklif göndermektedir. Çok robotlu yöntemde eğer robotlar haberleşme menzili içindelerse tüketici erkin kendi üretici erkinin ürettiği görevlere teklif verebildiği gibi diğer robotların üretici erkinlerinden gelen görevlere de teklif gönderebilir. Eğer birden fazla görevi yapmaya hak kazandıysa teklifi büyük olan görevi yapmak için hareket erkinine haber vermektedir.

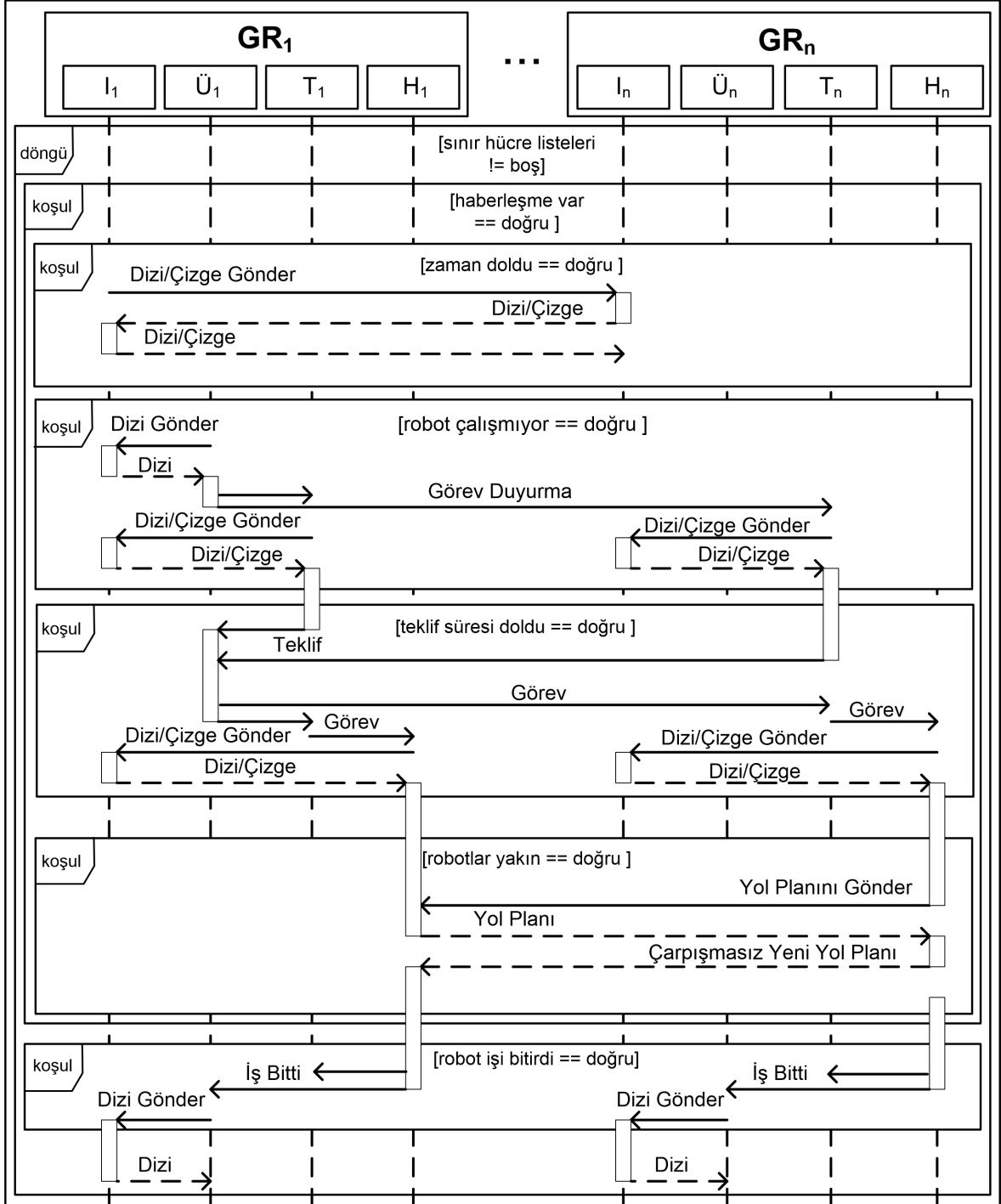
Hareket erkini iki temel görevden sorumludur. Bunların ilki, robotun bulunduğu noktadan belirlenen hedefine gidebilmesi için yol planının hazırlanmasıdır. Yol planı oluştuktan sonra robotun engellerden sakınarak güvenli bir şekilde hedefine varabilmesi için VAH kullanılmıştır. Bunlara ek olarak, robotlar arasındaki mesafe belirli bir eşik değerinin altına indiğinde robotlar arasındaki çarpışma önleme protokolünün işletilmesinden de hareket erkini sorumludur.

Şekil 5.2'de erkinler arası iletişimi gösteren sıralama diyagramı çok robotlu keşif için verilmiştir. Şekilde görüldüğü gibi, robotlar haberleşme menzili içinde olduklarında ızgara erkinleri belirli zaman aralıklarında iletişime geçmekte, metrik ve topolojik haritalarını birleştirmektedirler. Tüketici ve hareket erkinleri kullanacakları yol planına bağlı olarak çizge ya da diziyi istemektedirler. Robotlar arasındaki mesafe belirli bir eşik değerinin altına indiğinde robotların hareket erkinleri haberleşerek çarpışmasız yol planlarını üretmektedirler. Bu sıralama diyagramı sadece robotların haberleşme menzili içinde oldukları durum için verilmiştir. Haberleşmenin mümkün olmadığı durumlarda robotlar önceki iki bölümde anlatılan tek robotlu sistemler ile aynı şekilde çalışmaktadırlar. Bu noktadan sonra ilk olarak, robotların metrik ve topolojik haritalarını nasıl birleştirdikleri detaylı olarak anlatılacaktır. Daha sonra ise robotların birbirleriyle çarpışmalarını önlemek amacıyla geliştirilen protokol sunulacaktır.

5.2.2 Metrik ve topolojik haritaların birleştirilmesi

Çok robotlu sistemlerde, robotların yerel metrik ya da topolojik haritalarının birleştirilerek küresel bir haritanın elde edilmesi amacıyla çeşitli çalışmalar yapılmıştır. Ko ve diğerleri yaptıkları çalışmada robotların birbirlerinin ilk konumlarını bilmediği durumlarda harita birleştirme problemini ele almaktadırlar. Bir robot bir diğer robotun haberleşme menziline girdiğinde robotlardan biri diğer robotun konumuna kendi yerel haritasını ve parçacık süzgecini kullanarak karar vermektedir. Ancak, bu durumda dahi robotlar birbirlerinin konumlarını yanlış tespit edebilmektedir. Bu riski ortadan kaldırmak için robotlar uygun bir yerde buluşup aktif olarak konum doğrulaması yapmaktadırlar. Haberleşme menziline çıkan robot görevine devam etmektedir. Haberleşme menziline girdiği zaman, topladığı bilgileri paylaşmaktadır (Ko vd. 2003).

Bir diğer çalışmada Huang ve Beevers yerel topolojik haritaları birleştirmek için bir yöntem önermişlerdir. Bu yöntemde, ilk olarak iki yerel haritada bulunan düğüm ve ayrıtların birbirleri ile ilişkilerini açıklayan hipotezler oluşturmaktadırlar. Bu aşamada, düğüm ve ayrıtların özellikleri kullanılarak iki haritadaki düğüm ve ayrıtların uyumları test edilmektedir. İkinci olarak, her hipotez için düğüm yönelimlerine bağlı olarak bir dönüşüm



Şekil 5.2: Çok Robotlu Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)

tahmini yapılmaktadır. Daha sonra birbirleri ile tutarlı olan hipotezler kümelenmekte ve en iyi küme seçilerek birleştirilmektedir (Huang ve Beavers, 2005).

Carpin metrik haritaları birleştirmek için yeni bir yaklaşım önermiştir. Bu yaklaşımda, haritaları birleştirmek için gerekli olan öteleme, döndürme gibi dönüşüm kümeleri oluşturulmaktadır. Dönüşüm kümeleri oluşturulurken haritadan elde edilen spektral bilgilerde kullanılmaktadır. Hough dönüşümü ve hızlı yerdeğiştirme yöntemi sırasıyla yönelim ve öteleme tahmini yapılabilmesi için kullanılmışlardır. Bu noktadan sonra ise anlaşma (her iki haritada da boş veya dolu olan hücreler) ve uyuşmazlık olarak tanımladıkları ölçütleri kullanarak iki harita birleştirilerek tek bir küresel harita elde edilmektedir. Önerilen yöntem, tekrarsız, kararlı ve hızlıdır (Carpin, 2008).

Ozkucur ve Akın yaptıkları çalışmada robotların bir noktada karşılaşacakları varsayımını yapmaktadırlar. Robotlar görsel algılayıcıları ile birbirlerini ve ortamdaki eşsiz olmayan referans noktalarını (non-unique landmarks) farketmekte ve harita birleşiminde kullanılmaktadır. Robotlar karşılaştığında, robotun haritası, pozisyonu ve bu pozisyona göre referans noktalarının pozisyonları mesaj olarak karşılıklı olarak paylaşılmaktadır. Bu bilgileri kullanarak her robot ilk önce robotların aynı referans noktalarına sahip olacağı dönüşüme karar vermektedir. Dönüşüm yapıldıktan sonra haritaların birleştirilmesi her hücrenin tek tek işlenmesi ile gerçekleştirilmektedir (Özkucur ve Akın, 2009).

Topal ve diğerleri önerdikleri yöntemde metrik haritaları birleştirebilmek için her robotun yerel haritasından referans noktaları saptamaktadırlar. Bu noktaların belirlenmesi için SIFT algoritması kullanılmaktadır. Bu noktalardan birbirine karşılık gelenler kullanılarak dönüşüm matrisi elde edilmekte ve yerel haritalardan birine uygulanmaktadır. Bu aşamadan sonra da yerel haritalar anlaşma/uyuşmazlık fonksiyonlarına bağlı olarak olasılıksal bir yöntem kullanılarak birleştirilerek küresel bir harita oluşturulmaktadır (Topal vd. 2010).

Bu çalışmalar göz önüne alındığında, robotların yerel haritalarını birleştirmek için ilk önce robotların yerel koordinatlarının ya küresel koordinatlara ya da robotlardan birinin yerel koordinat sistemine dönüştürülmesi gerekmektedir. Bunun sebebi, robotların yerel haritalarının yerel koordinat sistemlerine göre oluşturulmasıdır. Dolayısıyla, yerel haritaların birleştirilmesi için robotların konum ve yönelimlerinin yüksek kesinlik ile saptanabilmesi gerekmektedir. Geçmiş çalışmalarda da genellikle odaklanılan konu robotların birbirlerine göre göreceli konumlarının ya da küresel konumlarının belirlenmesidir. Bu konumlardan biri yüksek doğrulukta elde edildikten sonra haritaları birleştirmek için yerel haritaların öteleme ve/veya döndürme dönüşümlerinin yapılması gerekmektedir. Tez çalışması kapsamında, robotların konumlarının kesin ve küresel olarak

bilindiği varsayılmaktadır. Bu durumda, geçmiş çalışmalarda ağırlıklı olarak araştırılan, bütün robotların aynı referans eksenine oturtulmasını amaçlayan işlemlere gerek duyulmamaktadır.

Çok robotlu keşif yaklaşımında robotların ızgara erkinleri yerel haritaların birleştirilmesi görevini üstlenmişlerdir. Bu amaç doğrultusunda, diğer robotun konum bilgisi belirli zaman aralıklarında (*tHarita*) dinlenmekte ve iki robot arasındaki mesafe önceden belirlenmiş bir eşik değerin (*hMenzil*) altına indiğinde diğer robotun yerel haritaları ile kendi yerel haritaları birleştirilmiştir.

5.2.2.1 Metrik haritaların birleştirilmesi

Tez kapsamında önerilen yöntemde, ızgarada bulunan her hücre için geçmiş çalışmalarda yer alan anlaşma/uyuşmazlık fonksiyonlarına benzer bir yaklaşım ile durum güncellemesi yapılmaktadır. Böylece robotların yerel metrik haritaları birleştirilerek küresel bir metrik harita elde edilmektedir. Bu noktada süreç, bir robotun ızgara erkininin, diğer robotun ızgara erkininden güncel metrik haritasını istemesi ile başlamaktadır. Bu isteği algılayan diğer robotlara ait ızgara erkinleri, kendi güncel metrik haritalarını istekte bulunan ızgara erkinine göndermektedirler. Daha sonra doluluk ızgarasındaki her hücre için, diğer robotun ızgarasındaki aynı indekslere denk düşen hücre karşılaştırılmaktadır. Eğer iki hücre de aynı durumdaysa (boş, dolu ya da bilinmeyen), birleştirilen doluluk ızgarasında da bu hücre aynı durum ile temsil edilmekte ve anlaşma durumu olarak tanımlanmaktadır. Eğer anlaşma durumu yoksa, ilk amacımız dolu hücrelerin saptanması olduğu için, her iki robotun doluluk ızgarasından birinde dolu hücrenin bulunması birleşmiş haritada o hücrenin dolu olmasına yeterli olmaktadır. Bunlara ek olarak, robotun kendi haritasında boş durumda bulunan hücreye karşılık gelen diğer robottaki hücre bilinmeyen ise birleşmiş haritada bu hücre boş olarak ifade edilmektedir. Ayrıca, diğer robotlardan gelen ve boş durumunda olan hücreleri, robotun kendi ızgarasındaki hücreler ile birleştirirken dikkatli olmak gerekmektedir. Eğer hem robotun kendi hücresi hem de diğer robotun hücresi boş ise herhangi bir sıkıntı olmamaktadır. Ancak, robotun kendi hücresi bilinmeyen durumunda iken diğer robotun hücresi boş durumunda ise birleştirilen haritada bu hücreye boş olarak güncellenmemelidir. Bu noktada daha önce üç durum ile ifade edilen ızgara hücrelerine *DIGER_BOS* adını verdiğimiz yeni bir durum eklenmiştir. Bu özel durum, robotun kendi haritasında bilinmeyen durumda bulunan hücreler için, diğer robottan gelen hücrenin durumunun boş olması olayını tanımlamak için kullanılmaktadır. Son olarak, ızgara erkini güncel metrik haritayı, çalışmakta olan diğer erkinlere gönderirken *DIGER_BOS* durumunda olan bütün hücreleri *BOS* olarak değiştirmektedir. Böylelikle diğer erkinlerdeki işleyişin bozulması önlenmektedir. Metrik haritaların birleştirilmesinde kullanılan işlemler Algoritma 10'da verilmiştir.

Algoritma 10: Metrik Haritaların Birleştirilmesi

Girdi: Kendi metrik haritası KMH , Diğer robotun metrik haritası $DRMH$

Çıktı: Birleşmiş metrik harita BMH

```

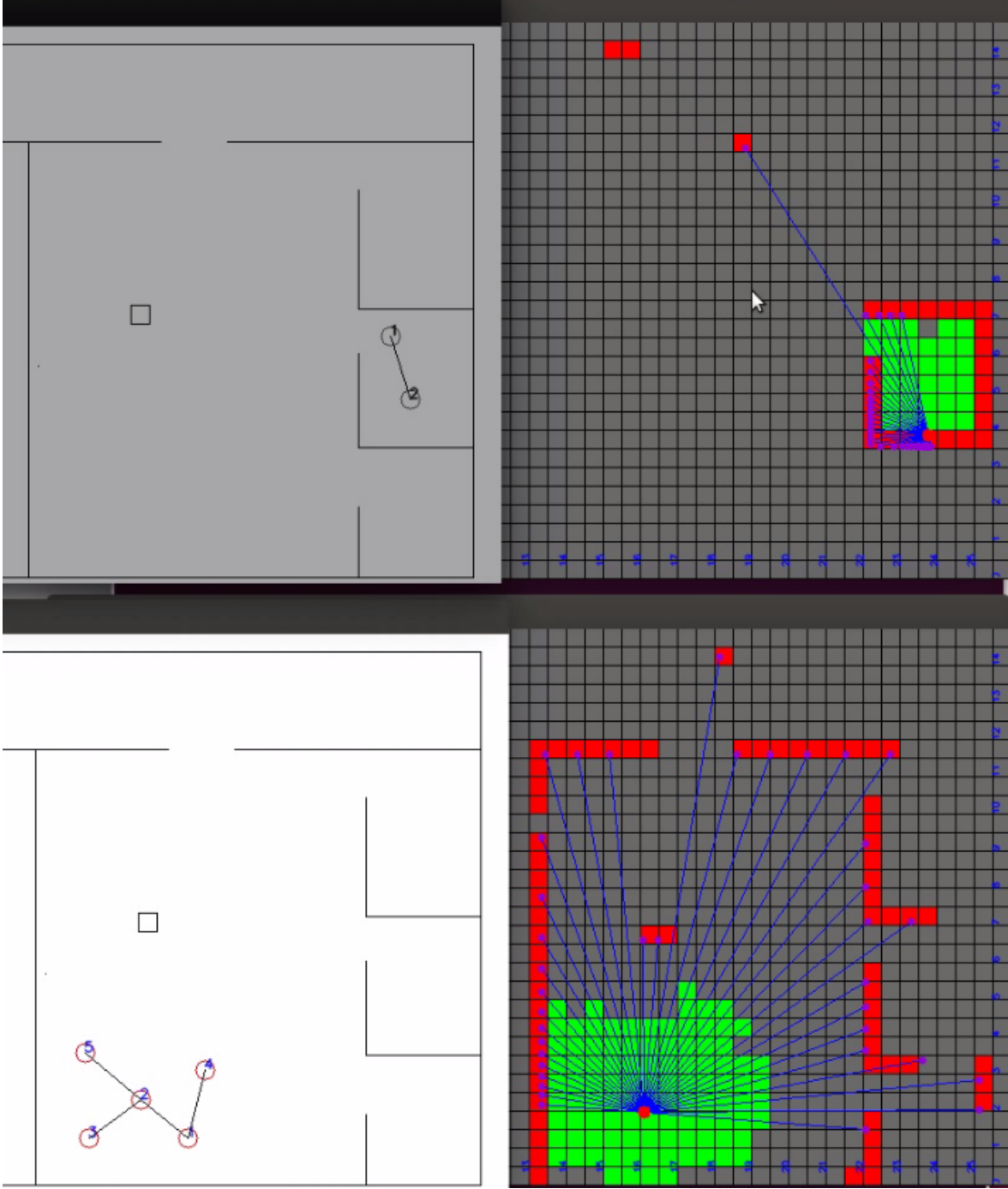
for  $i \leftarrow 0$  to  $x\_boyut$  do
  for  $j \leftarrow 0$  to  $y\_boyut$  do
    if  $KMH(i,j) == DOLU$  ya da  $DRMH(i,j) == DOLU$  then
      |  $BMH(i,j) = DOLU$ ;
    else if  $KMH(i,j) == BILINMEYEN$  ve  $DRMH(i,j) == BILINMEYEN$  then
      |  $BMH(i,j) = BILINMEYEN$ ;
    else if  $KMH(i,j) == BOS$  ve  $DRMH(i,j) \neq DOLU$  then
      |  $BMH(i,j) = BOS$ ;
    else if  $KMH(i,j) == BILINMEYEN$  ve  $DRMH(i,j) == BOS$  then
      |  $BMH(i,j) = DIGER\_BOS$ ;
    else if  $KMH(i,j) == DIGER\_BOS$  then
      |  $BMH(i,j) = DIGER\_BOS$ ;
    end
  end
end

```

5.2.2.2 Topolojik haritaların birleştirilmesi

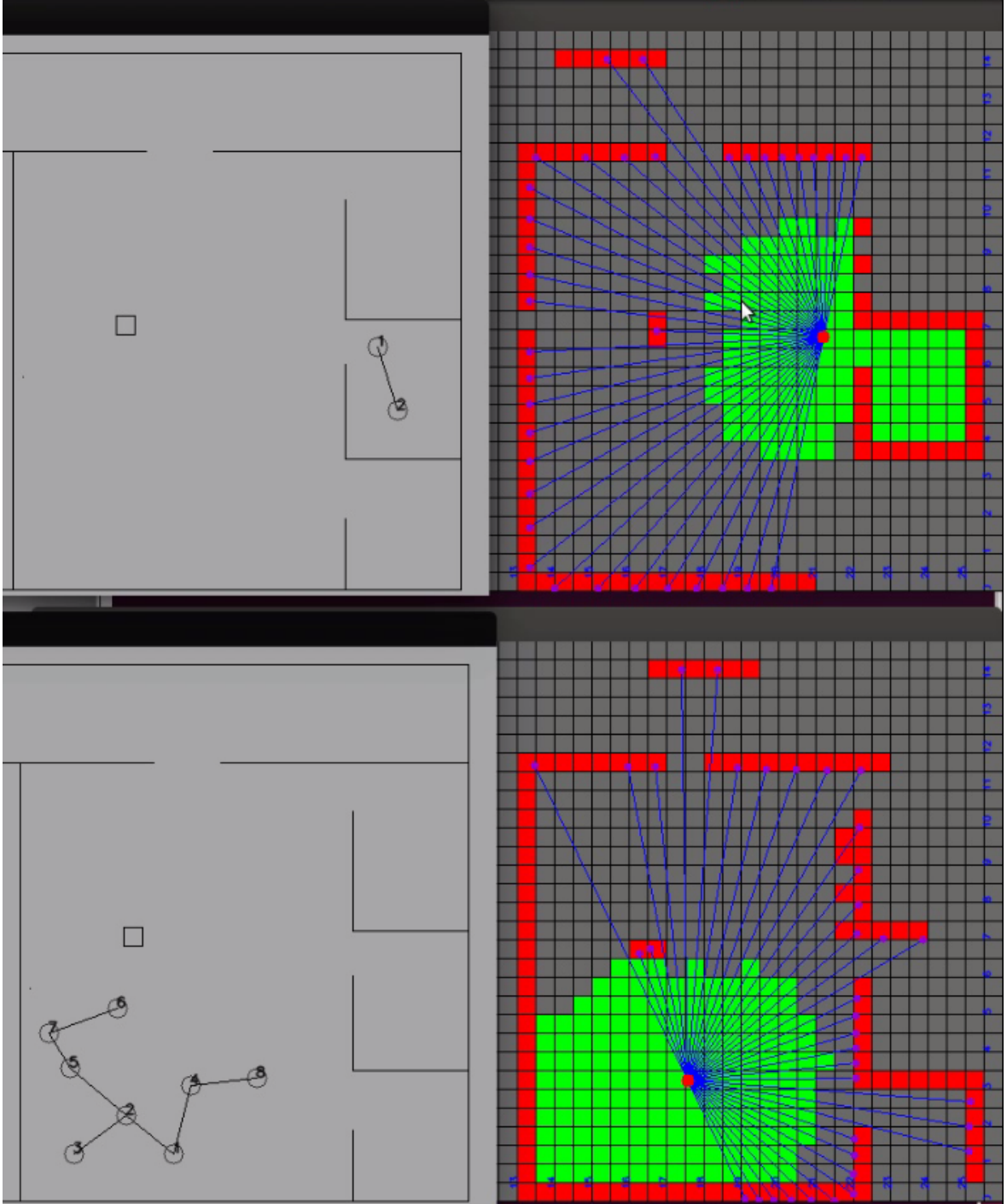
Tez çalışması kapsamında, her robotun kendi boş hücrelerini kullanarak topolojik haritasını oluşturması amaçlanmıştır. Bunun sebebi spektral kümelemenin dezavantajlarından biri olan ilginlik matrisinin boyutunun çok fazla büyümesini önlemektir. Ayrıca, aynı boş hücrelerin tekrar tekrar her robotun topolojik haritasının üretiminde kullanılması birleştirilecek topolojik haritada üst üste veya birbirine çok yakın fazla sayıda düğüm oluşturacaktır. Bu sıkıntıların üstesinden gelmek amacıyla $DIGER_BOS$ durumu kullanılmaktadır. Bu yeni durumla birlikte boş hücreler robotun kendi boş hücreleri ve diğer robotlardan gelen boş hücreler olarak ayrılabilen ve diğer robotların boş hücreleri topolojik harita üretiminde kullanılmamaktadır.

Robotlar ilk defa haberleşme menziline girdiklerinde her robot kendi topolojik haritasında bulunan bütün düğümleri diğer robota göndermektedir. Bununla birlikte, her robot son gönderdiği düğümün indeksini de tutmaktadır. Diğer robottan gelen düğümleri kabul eden robot kendi düğümlerinden sonraki ilk indeksten başlamak üzere yeni gelen düğümleri indekslemekte ve topolojik haritasına katmaktadır. Bu süreç tamamlandığında her robot en küçük kapsayan ağaç algoritması ile düğümleri birbirine bağlamaktadır. Her iki robotta da farklı indekslere sahip olmalarına rağmen birbirinin tamamen aynısı olan düğümler bulunmaktadır. Dolayısıyla birleştirilen topolojik harita her iki robot için de bire bir aynı olmaktadır. Bu noktadan sonra ($tHarita$) süresi dolduğunda robotlar hala ($hMenzil$) içinde bulunuyorlarsa sadece yeni ürettikleri düğümler için az önce anlatılan süreci işletmektedirler.



Şekil 5.3: Harita birleştirme aşamaları (1)

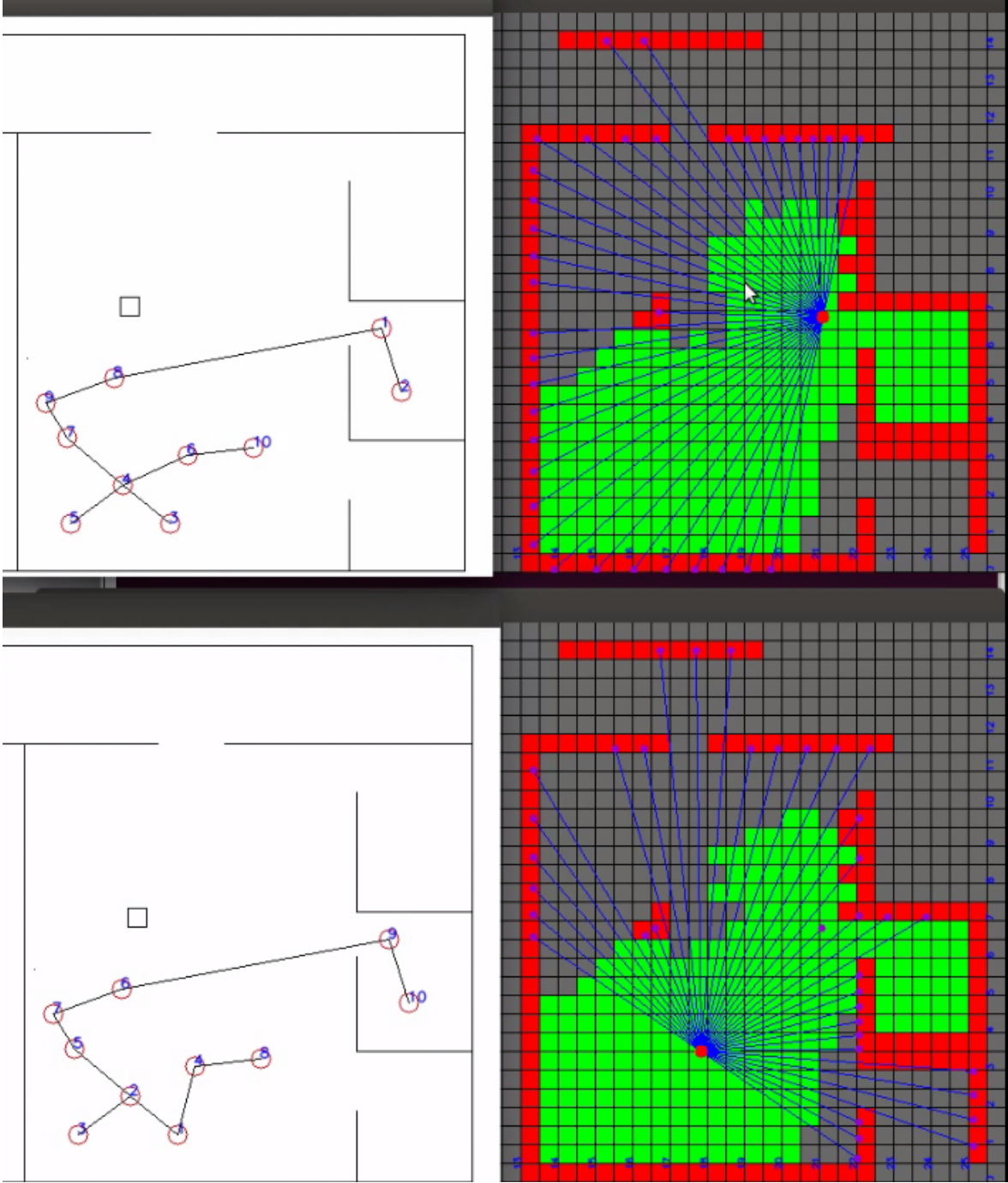
Metrik ve topolojik haritaların birleştirme aşamaları adım adım verilmiştir. Şekil 5.3'te alta bulunan robot 5 düğüm, sağda bulunan robot ise 2 düğüm üretmişlerdir. Robotlar haberleşme menzili dışında oldukları için harita birleştirme yapılmamaktadır.



Şekil 5.4: Harita birleştirme aşamaları (2)

Şekil 5.4'te robotlar bilinmeyen alanlara doğru ilerlemektedirler. Bu süreçte altta bulunan robot topolojik haritasına 3 düğüm daha eklemiştir. Robotlar hala haberleşme menziline giremedikleri için haritalar birleştirilememiştir.

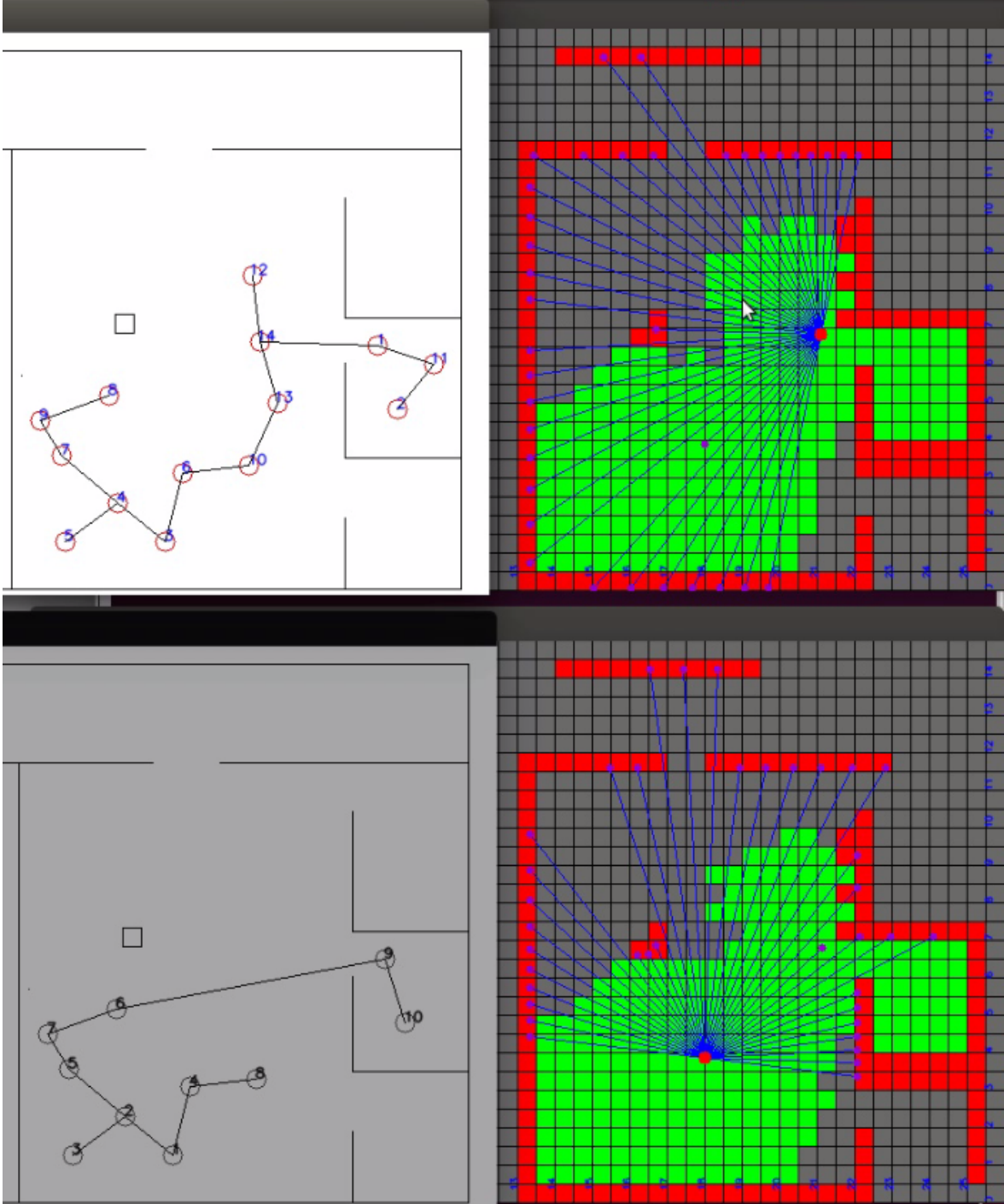
Şekil 5.5'te robotlar birbirlerinin haberleşme menzillerinin içinde bulunmaktadırlar. Bu noktada metrik haritalar birleştirilmiş ve her iki robotta aynı metrik haritaya sahip



Şekil 5.5: Harita birleştirme aşamaları (3)

olmuşlardır. *DIGER_BOS* durumunda olan hücreler için ayrıca bir renk kullanılmamıştır. Bununla birlikte, robotların birbirlerini engel olarak görmesini engellemek amacıyla diğer robotun 8 komşuluğundaki hücreler güncellenmemektedir. Şekilde görüldüğü üzere sol taraftaki robot sağ taraftaki robotun bulunduğu hücreyi lazer ışını ile göstermektedir. Ancak, bu hücreyi dolu olarak güncellememektedir. Topolojik haritada beklendiği üzere her iki robot için birebir aynıdır. Sadece düğümlerin indeksleri farklı gösterilmiştir. Sol tarafta bulunan robotun haritaların birleşmesinden önce 8 düğümü bulunmaktaydı. Sağ taraftaki

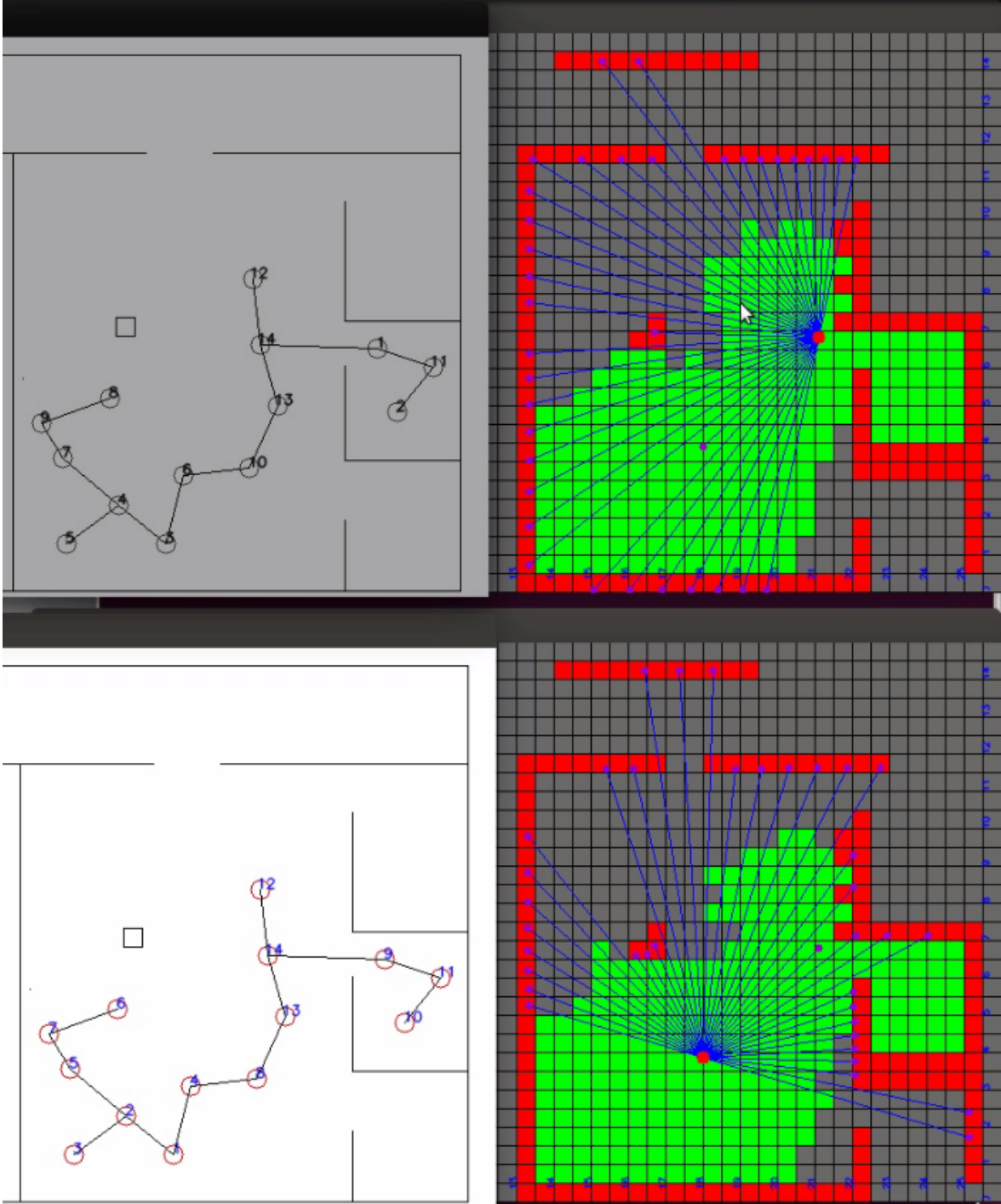
robottan gelen 2 düğüm 9 ve 10 numaralı düğümler olarak haritaya eklendiler. Benzer durum diğer robot için de görülmektedir.



Şekil 5.6: Harita birleştirme aşamaları (4)

Şekil 5.6'da sağ taraftaki robot 4 yeni düğümü topolojik haritasına eklemiştir. Bununla birlikte, sol taraftaki robotta bu yeni güncelleme yapılmamıştır. Ayrıca, metrik

haritalarda birbirlerine çok benzemelerine rağmen birebir aynı değildir. Bu durumun sebebi güncelleme için gerekli olan sürenin (*tHarita*) dolmamış olmasıdır. Afet ortamları göz önüne alınarak haberleşmenin çok verimli olmayacağı düşüncesi ile robotlar harita birleştirme işlemlerini belirli aralıklarla yapmaktadırlar.



Şekil 5.7: Harita birleştirme aşamaları (5)

Şekil 5.7'de ise güncelleme için gerekli olan süre tamamlanmış ve robotlar haritalarını birleştirmişlerdir. Sol taraftaki robot ilk birleştirmeden sonra topolojik haritasına yeni düğüm ekmediği için sağ taraftaki robotun bulunduğu 4 düğüm her iki topolojik haritada da aynı indekse sahiptir.

5.2.3 Çarpışma önleme protokolü

Birden fazla robotun aynı ortamda çalıştığı durumlarda robotlar birbirleri ile sıklıkla karşılaşmaktadırlar. Bu anlarda, robotların birbirlerine çarpmadan görevlerini yerine getirebilmeleri için geçmiş çalışmalarda çeşitli yaklaşımlar önerilmiştir. Birleştirilmiş (coupled) planlama yöntemlerinde bütün robotların planları aynı anda yapılmaktadır. Bu yöntemler, en iyi sonuçları üretmekte ve robotlar birbirlerine ya da ortamdaki herhangi bir nesneye çarpmadan görevlerini tamamlamaktadırlar. Bununla birlikte, bu yöntemler planlama yapmak için uzun zaman harcamakta, ölçeklenebilir olmamakta ve dinamik planlamaya ihtiyaç duyulan ortamlarda verimsiz çalışmaktadır. Ayrışmış (decoupled) planlama yöntemlerinde ise başta robotların yol planları yapılmakta daha sonra çarpışmaları önlemek amaçlanmaktadır. Ayrışmış planlama yöntemleri içinde çarpışmaları önlemek için çeşitli yaklaşımlar kullanılmıştır. Bunlardan bazıları, robotların hızlarını ayarlayarak, bazıları ise planları birleştirerek ya da çizelgeleyerek merkezi (centralized) bir yapıda robotlar arası çarpışmayı önlemektedir. Dağıtık (decentralized) olarak çarpışmayı önleyen yöntemler genellikle potansiyel alanlar ya da robotlara öncelik vererek ve trafik kurallarına benzer kurallar kullanmışlardır.

Bu tez kapsamında, robotların çarpışmasını önlemek için ayrışmış ve dağıtık bir yaklaşım kullanılmıştır. Robotların arasındaki mesafe belirli bir eşik değerin altına düştüğünde (*çMenzil*) robotlardan biri diğerinden yol planını istemektedir. Robot çarpışma protokolü ilk olarak robotların hedeflerini kontrol etmektedir. Bunu yapabilmek için, hedeflerin doluluk ızgarasında hangi hücrede olduklarına karar verilmektedir. Daha sonra, hedeflerin ilk koordinatları arasındaki farkın mutlak değeri daha önceden belirlenen ve *kEşik* olarak adlandırılan eşik değerinden küçük ise hedeflerin ikinci koordinatları benzer şekilde kontrol edilmektedir. Hedeflerin ilk ve ikinci koordinatları için bu koşullar sağlanıyorsa robotların aynı hedefe gitmek istedikleri anlaşılmaktadır. Robotların birbirlerine çarpmalarını önlemek için robotlardan birisinin durmasının uygun olacağı düşünülmüştür. Sonuç olarak, yakın robotun ortak hedefe gitmesi ve diğer robotun beklemesi ile robotların birbirlerine çarpmaları önlenmektedir.

Robotlar farklı hedeflere sahiplerse bu durumda robotların yol planlarının incelenmesi gerekmektedir. Robotlar çarpışma protokolünü aralarındaki mesafe (*çMenzil*) az ise çalıştırmaktadır. Dolayısıyla, robotların yol planlarının ilk bir kaç elemanına (*pEşik*)

bakılması yeterli olacaktır. Bu noktada, robot planlarının birinin ilk elemanı ile başlanmakta ve diğer robotun planının ($pEşik$) kadar elemanına bakılmaktadır. Yol planlarının ilk koordinatları arasındaki farkın mutlak değeri $kEşik$ küçük ise ikinci koordinatları benzer şekilde kontrol edilmektedir. İlk ve ikinci koordinatları için bu koşullar sağlanıyorsa robotların yol planlarının çakıştığı ve eğer bu planları uygulamaya devam ederlerse robotların çarpışacakları anlaşılmaktadır. Bu koşullar sağlanmıyorsa bu işlem ya çakışma bulana kadar ya da ilk elemanına bakılan yol planının $pEşik$. elemanına ulaşana kadar devam etmektedir.

Robotların yol planlarında çakışma tespit edildiğinde ilk olarak robotların hedeflere olan uzaklıkları hesaplanmaktadır. Daha sonra da robotların çarpışmasını önlemek için birbirlerinden uzaklaştırmak veya birini bekletmek üzere yol planları hazırlanmaktadır. Eğer her iki robotta birinci robotun hedefine ikinci robotun hedefinden daha yakınsa robotlardan herhangi birinin birinci robotun hedefine gitmesi çarpışma tehlikesini ortadan kaldırmayabilir. Bu durumda ikinci robotun hedefine daha yakın olan robotun bu hedefe gitmesi ve diğer robotun da beklemesi gerekmektedir. Birinci robotun hedefine hiçbir robot gitmeyecektir. Eğer her iki robotta ikinci robotun hedefine birinci robotun hedefinden daha yakınsa robotlardan herhangi birinin ikinci robotun hedefine gitmesi çarpışma tehlikesini ortadan kaldırmayabilir. Bu durumda birinci robotun hedefine daha yakın olan robotun bu hedefe gitmesi ve diğer robotun da beklemesi gerekmektedir. İkinci robotun hedefine hiçbir robot gitmeyecektir. Son olarak, birinci robotun ikinci robotun hedefine ve ikinci robotun ise birinci robotun hedefine yakın olduğu anlarda robotların birbirleri ile hedeflerini değiştirmelerinin çarpışmayı önleyeceği düşünülmüştür.

Bölüm sonunda verilecek olan görsel sonuçların daha iyi analiz edilebilmesi için bazı ön tanımlamalar yapılmıştır. İlk olarak, birinci robot ile ilgili herşey kırmızı, ikinci robot için ise siyah ile gösterilecektir. Eğer robotlar aynı hedefe gitmek isterlerse o hedef yeşil çember ile gösterilmektedir. Çemberin içindeki rakamın rengi hangi robotun ortak hedefe geldiğini göstermektedir. Bununla birlikte, gidilmeyen düğümler için X harfi, bekleyen robot için kare şekli, birinci robotun ikinci robotun hedefine gittiğini göstermek için baklava şekli, ikinci robotun birinci robotun hedefine gittiğini göstermek için beşgen şekli, robotların hedef değişimleri için üçgen şekli kullanılmıştır.

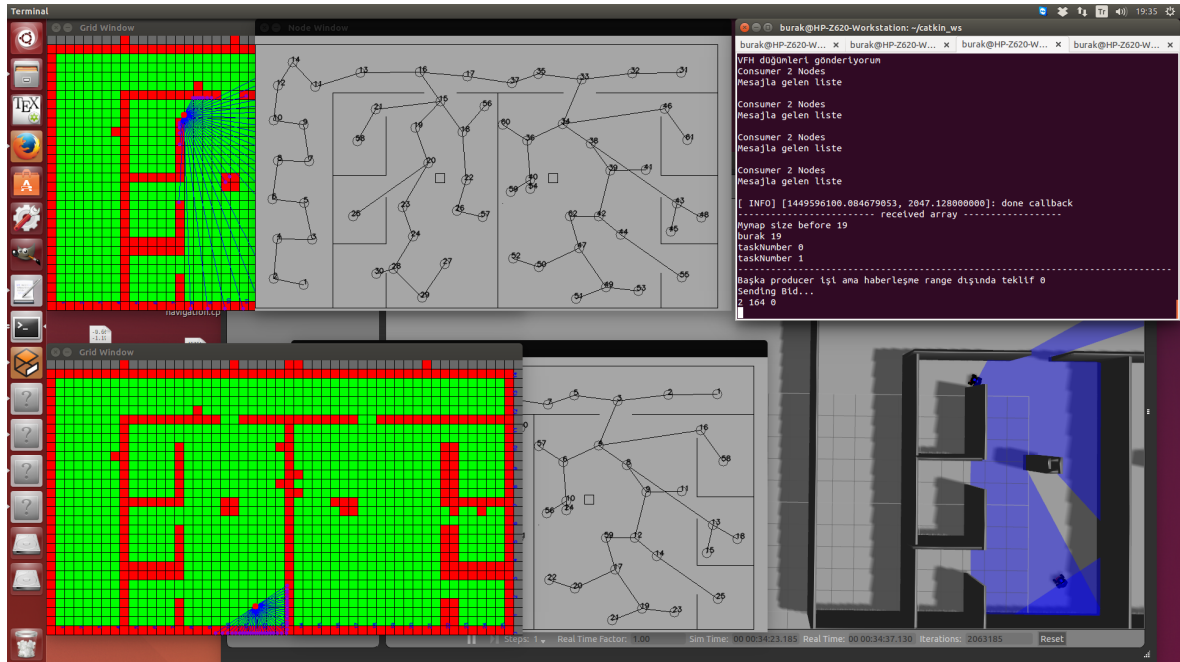
5.3 Sonuçlar

Bu alt bölümde ilk olarak deneysel kurulum yer alacaktır. Deney ortamı, erkinleri çalıştırırken kullanılan parametreler gibi detaylar bu kısımda yer alacaktır. Daha sonra A* ve Dijkstra yol planı ile birlikte farklı fiyatlandırma kriterleri için sayısal ve görsel sonuçlar

verilecek ve her bir fiyatlandırma kriteri için yol planlarının sonuçlar üzerindeki etkileri irdelenecektir. Son olarak çarpışma önleme protokolünün ve robotların başlangıç noktaları arasındaki farklılıkların robotlar arası koordinasyonu nasıl etkilediği analiz edilecektir.

5.3.1 Deneysel kurulum

Önerilen yöntem, Gazebo benzetim ortamında modellenen Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Laboratuvar binasında test edilmiştir. Ortamın boyutları 26×15 metredir ve 52×30 'luk metrik ızgara ile ifade edilmiştir. Şekil 5.8'de robotlar keşfi tamamladıktan sonra elde edilen metrik ve topolojik harita örnekleri verilmiştir.



Şekil 5.8: Çok Robot Metrik ve Topolojik Harita Örnekleri

Çok robotlu sistemde robotların haberleşme menzilleri ($hMenzil$) 5 metre olarak seçilmiştir. Robotlar haberleşme menzili içindeyse topolojik ve metrik haritaları güncelleme periyodu ($tHarita$) 1 dakika olarak belirlenmiştir. Robotların çarpışma önleme protokolünü işletmeleri için birbirlerine 2 metreden az bir mesafede olmaları gerekmektedir ($çMenzil$). Robotların aynı hedefe sahip olup olmadıklarını kontrol etmek için kullanılan $kEşik$ parametresi 2 hücre olarak belirlenmiştir. Son olarak robotların yol planlarında bir çakışma olup olmadığına bakmak için yol planlarındaki 3 hücre analiz edilmektedir ($pEşik$).

Uygulamalar gerçekleştirilirken (A* ve Dijkstra) yol planı ile birlikte üç farklı fiyatlandırma kriteri (en yakın sınır hücre, en çok kenar hücreye sahip sınır hücre ve market sınır hücre) kullanılmıştır. Robotlar üç farklı yerden görevlerine başlamakta ve belirlenen fiyatlandırma kriteri ve yol planına göre bütün haritanın keşfi tamamlanana kadar görevini sürdürmektedir.

5.3.2 Deneysel sonuçlar

A* ve Dijkstra yol planı ile birlikte üç farklı fiyatlandırma kriterinin (en yakın sınır hücre, en çok kenar hücreye sahip sınır hücre ve market sınır hücre) keşif süresine ve robotun katettiği mesafe olan etkileri 3. ve 4. bölümde detaylı olarak analiz edilmiştir. Bu alt bölümde, sadece iki robotlu sistemde bu etkilerdeki değişiklikler anlatılacaktır.

Çizelge 5.1'de hem A* hem de Dijkstra yol planları ile üç farklı fiyatlandırma kriteri için sayısal sonuçlar verilmiştir. Sonuçlar fiyatlandırma açısından incelendiğinde, tek robot kullanılarak yapılan keşifte olduğu gibi en çok kenar hücreye sahip sınır hücre kriteri kullanıldığında; robotların üç farklı başlangıç durumunda da, market ve en yakın sınır hücre kriterlerine göre daha uzun mesafeler katettiği ve keşfin daha uzun süre aldığı açıkça görülmektedir. Bununla birlikte, en yakın sınır hücre kriterleri kullanıldığında ise genellikle keşif diğer kriterlere göre daha kısa sürede tamamlanmakta ve robotlar daha az mesafe katetmektedirler.

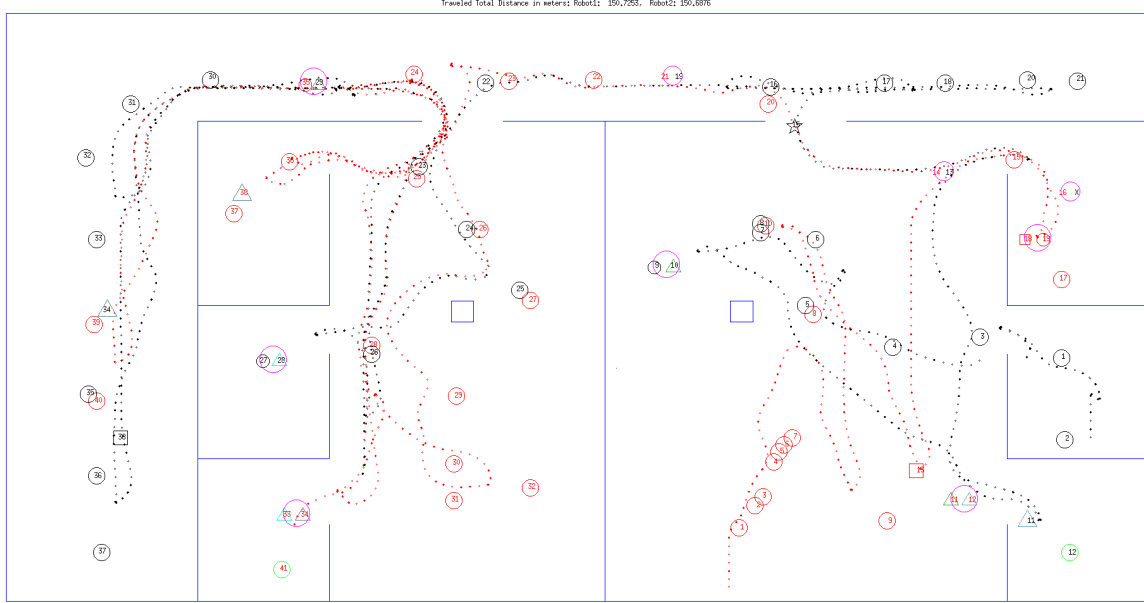
Tek robot kullanılarak yapılan keşifte olduğu gibi Dijkstra yol planı ile bütün fiyatlandırma kriterleri için robotların katettiği mesafe, A* yol planına göre daha fazladır. Bununla birlikte, Dijkstra yol planı ile robotların keşfi tamamlama süresi A* yol planına göre daha kısadır. Bu durumu açıklamak amacıyla üç fiyatlandırma kriteri için A* ve Dijkstra ile elde edilen görsel sonuçlar verilecektir. Her fiyatlandırma kriterinde robotlar her iki yol planı içinde aynı noktadan keşfe başlamaktadır. Görsel sonuçlarda kırmızı çemberlerin içinde kırmızı rakamlarla belirtilen konumlar sırasıyla Robot 1 için seçilen hedefleri göstermektedir. Bu çemberler arasında kalan kırmızı noktalar ise robotun izlediği yolları ifade etmektedir. Benzer şekilde Robot 2 için ise siyah renk kullanılmıştır.

Çizelge 5.1: İki Robot A* ve Dijkstra Sonuçları

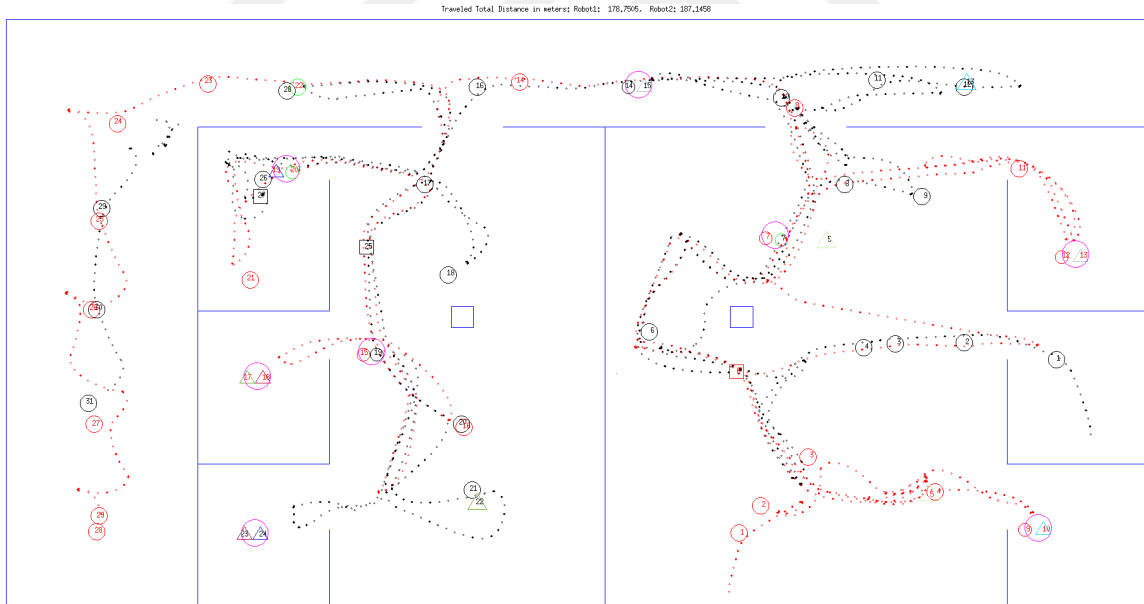
Fiyatlandırma Kriteri	Konum		Süre (Dk)		Mesafe (m)			
	R1	R2			Robot 1		Robot 2	
			A*	Dijk	A*	Dijk	A*	Dijk
En çok kenar hücresi	2.5, 0.3, 1.57	24.8, 13.2, 3.14	52	52	136.33	152.02	115.76	183.37
En yakın			44	34	92.98	101.37	88.77	119.07
Market			39	35	81.80	107.54	87.71	116.78
En çok kenar hücresi	16, 0.3, 1.57	24, 4, 1.57	67	57	150.72	178.75	150.68	187.14
En yakın			47	37	86.53	118.96	83.77	124.33
Market			55	44	131.80	166.85	102.53	131.69
En çok kenar hücresi	9, 7, 1.57	11, 7, 1.57	73	67	158.18	236.19	196.37	247.63
En yakın			45	38	80.16	121.46	86.78	114.69
Market			51	41	125.39	128.30	98.60	135.55

Şekil 5.9 ve 5.10'da sırasıyla A* ve Dijkstra yol planları için birinci robotun 16,0.3,1.57 konumundan ve ikinci robotun ise 24,4,1.57 konumundan başladıkları durumdaki en çok kenar hücresine sahip sınır hücre kriterinin yaptığı hedef seçimleri görünmektedir.

Şekiller incelendiğinde, tek robot kullanılarak yapılan keşifte olduğu gibi Dijkstra yol planı kullanıldığında robotun A* yol planı kullanıldığı duruma göre daha az sayıda hedef belirlediği gözlemlenebilir. Dolayısıyla, A* yol planı kullanıldığında robotlar yeni hedeflerin atanması için Dijkstra yol planına göre daha fazla zaman harcayacaklardır. Ayrıca, çok robotlu sistemde, robotların haberleşme menzili içinde olduğunda, görev paylaşımı esnasında yapılan görüşmeler uzamaktadır. Bunun sebebi, bir robot tam görüşmeleri kapatırken diğer robotun bu görüşmeye katılmasıyla robotlara atanacak görevlerin tekrar değerlendirilmesi gerekmektedir. Dolayısıyla robotlar kendilerine atanacak olan görevleri daha uzun süre beklemektedirler. Diğer yandan, Dijkstra yol planı kullanıldığında robotların A* yol planı kullanıldığı duruma göre ortamla ilgili daha fazla bilgi edineceği açıktır.



Şekil 5.9: A* en çok kenar hüresine sahip sınır hücre kriteri

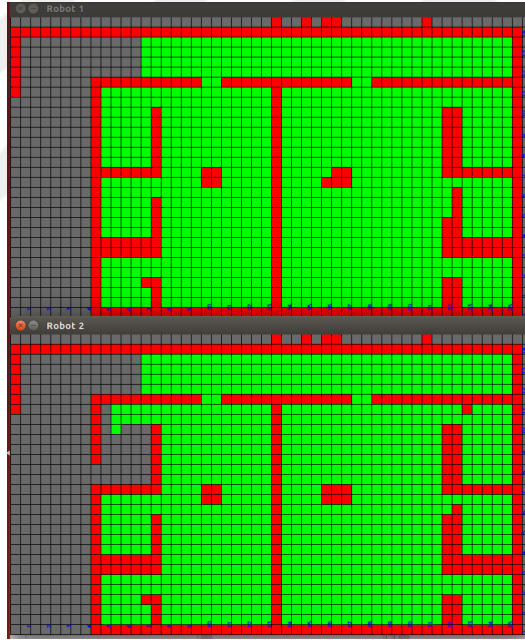


Şekil 5.10: Dijkstra en çok kenar hüresine sahip sınır hücre kriteri

Tek robot kullanılarak yapılan keşifte A* yol planı ile görüşmelerde kaybedilen zaman robotun daha az mesafe katetmesiyle; Dijkstra yol planı ile robotun daha fazla mesafe katederek kaybedilen zaman ise daha fazla bilgi edinmesiyle dengelenmiş ve hemen hemen aynı zamanlarda keşifler tamamlanmıştı. Çok robotlu sistemde ise görevler paylaşılıp, robotlar daha az mesafe kaydettiği için Dijkstra'nın dezavantajı azalmakta ve

robotların görüşme süreleri de arttığı için A* dezavantajı artmaktadır. Sonuç olarak, çok robot kullanarak keşif yapıldığında iki yol planı arasında keşif süresi farkı bu şekilde açıklanabilir.

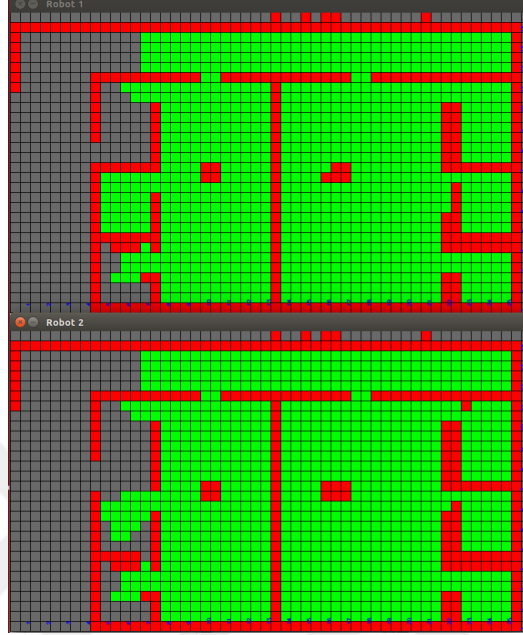
En çok kenar hücrelerine sahip sınır hücre kriteri koordinasyon açısından incelendiğinde ise robotların A* kullanıldığında 5 kere iş değiştirdikleri ve 3 kere bekledikleri görülmektedir. Benzer şekilde Dijkstra kullanıldığında ise 6 kere iş değiştirip, 3 kere beklemektedirler. Ayrıca, sol büyük odanın keşfi sırasında her iki yol planı için de robotlar benzer hedefler belirlemektedirler. Bunun sebebi, haberleşme kısıtları yüzünden robotlar arasındaki koordinasyonun sağlanamamasıdır. Robotlar haberleşme menzili içinde olduklarında haritalarını güncelleyebilmekte ve görevler için teklif gönderip robotlar arası koordinasyonun iyileşmesini sağlayabilmektedir. Bu durum, robotun yeni bilgiler kazanmasını sağlarken, haberleşme kesildiğinde eski bilgiler yüzünden yanlış hedef seçimleri yapmasına da sebep olmaktadır. Şekil 5.11'de bu duruma bir örnek verilmektedir.



Şekil 5.11: Robotların $hMenzil$ dışında olduklarında hedef seçimi

Bu örnekte, birinci robotun 21. hedefine ve ikinci robotun ise 24. hedefine (Şekil 5.10) ulaştığı andaki doluluk ızgaraları verilmiştir. Birinci robot 24. hedefine ikinci robot ise 19. hedefine giderken yolda karşılaşmış ve hedeflerini değiştirmişlerdir. Dolayısıyla robotlar bu anlarda $hMenzil$ içinde bulunmakta ve haritalarını güncellemektedirler. Daha sonra birbirlerinden uzaklaşıp, $hMenzil$ dışına çıkmışlardır. İkinci robot sahip olduğu eski bilgiler yüzünden birinci robotun hali hazırda keşfettiği üst küçük odaya hareket etmekte ve

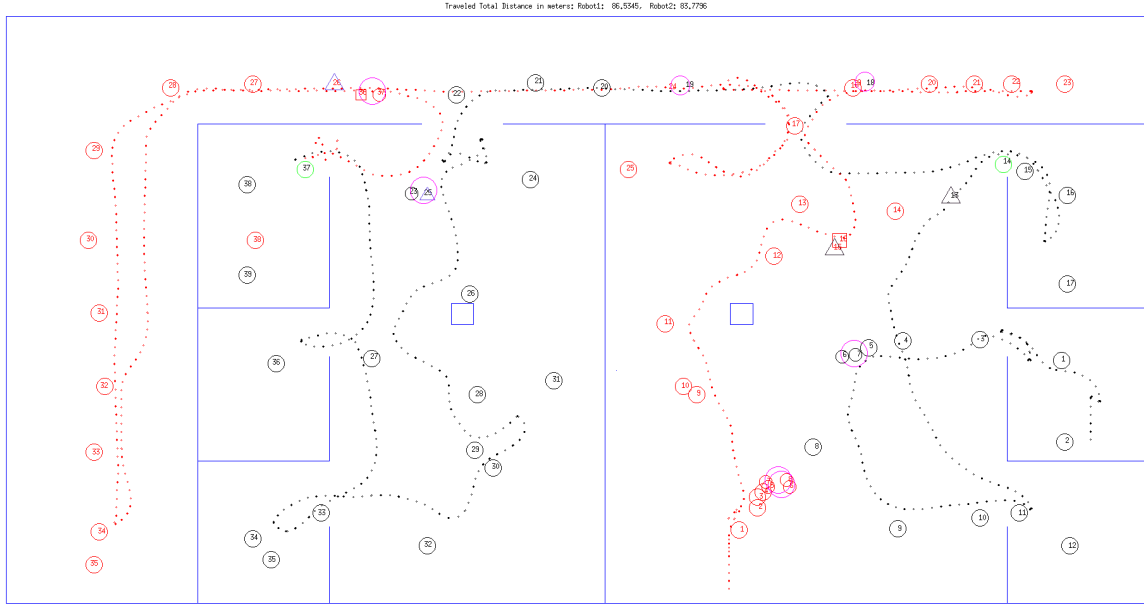
robot takımının verimliliği düşmektedir. Ayrıca, robotlar $hMenzil$ içinde olsalar dahi $tHarita$ parametresi 1 dakika olduğu için robotlar haritalarını birleştiremeden görev paylaşımı yapabilmektedirler. Şekil 5.12'de bu duruma bir örnek verilmektedir.



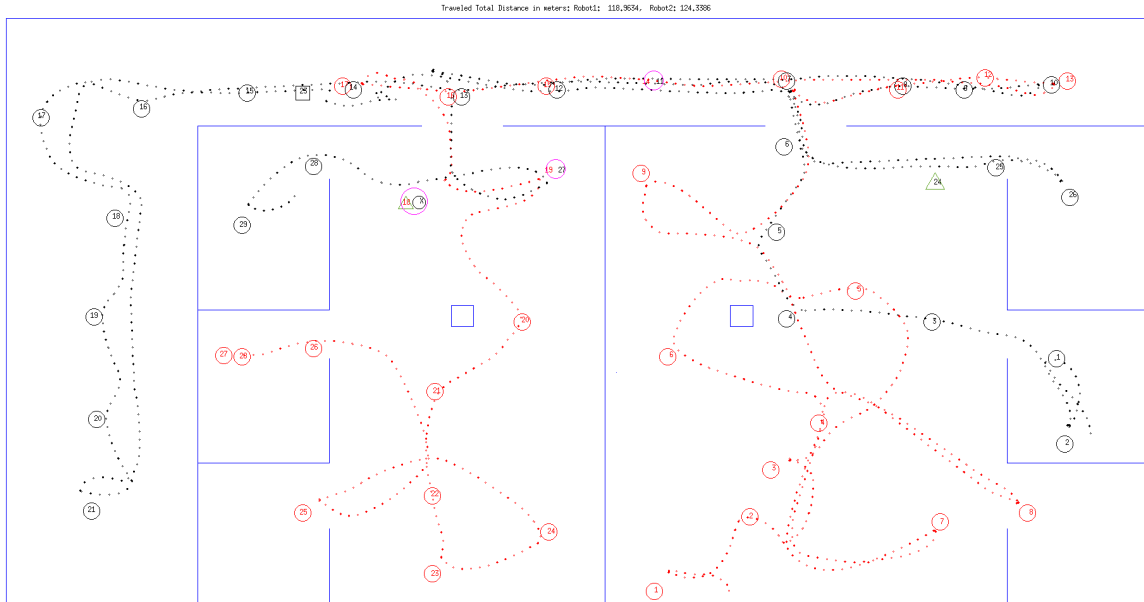
Şekil 5.12: Robotların $tHarita$ parametresi dolmadan hedef seçimi

Bu örnekte, birinci robotun 17. hedefine ve ikinci robotun ise 22. hedefine (Şekil 5.10) ulaştığı andaki doluluk ızgaraları verilmiştir. İkinci robot sahip olduğu eski bilgiler yüzünden birinci robotun hali hazırda keşfettiği ortadaki küçük odaya hareket etmektedir. Birinci robot ise ortadaki küçük odadan çıkarak aşağıdaki küçük odaya doğru ilerlerken yolda karşılaşmakta ve görev değişimi yapmaktadırlar. Dolayısıyla, birinci robot yine ortadaki küçük odaya, ikinci robot ise aşağıdaki küçük odaya gitmekte ve robot takımının verimliliği düşmektedir. Şekil 5.10'da aynı renkte gösterilen üçgenler hedef değişimlerini ifade etmektedir. Bu durumda, birinci robotun gitmek istediği 18. hedef, hedef değişiminden önce ikinci robotun vardığı 23. hedef olmaktadır. Benzer şekilde, ikinci robot gitmek istediği 23. hedef, hedef değişiminden önce birinci robotun 18. hedefi olmaktadır. Büyük pembe daireler içindeki hedefler, hedeflerin birebir aynı olduğu durumlarda kullanılmaktadır. Sonuç olarak, en çok kenar hücresine sahip sınır hücre kriteri kullanıldığında robotlar sürekli olarak $hMenzil$ içine girmekte ve dışına çıkmaktalar ve bu durumda robot koordine olmasını zorlaştırmaktadır.

Şekil 5.13 ve 5.14'te sırasıyla A* ve Dijkstra yol planları için birinci robotun 16,0.3,1.57 konumundan ve ikinci robotun ise 24,4,1.57 konumundan başladıkları durumdaki en yakın sınır hücre kriterinin yaptığı hedef seçimleri görünmektedir.

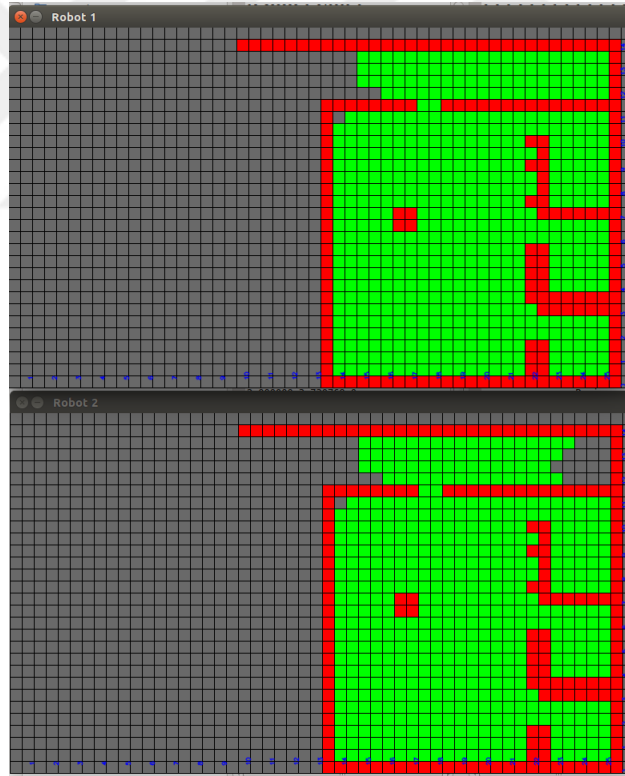


Şekil 5.13: A* en yakın sınır hücre kriteri



Şekil 5.14: Dijkstra en yakın sınır hücre kriteri

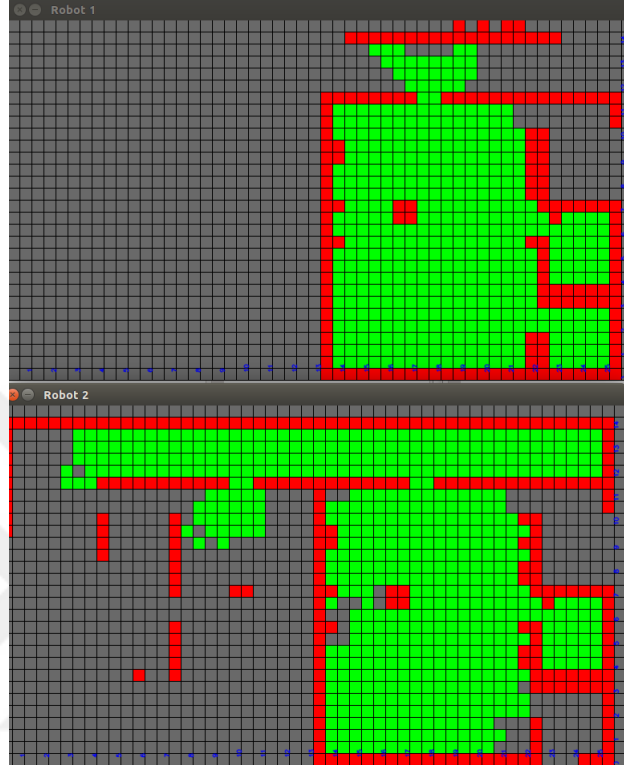
Robotların A* yol planını kullanıldığında 2 kere iş değiştirdikleri ve 2 kere bekledikleri görülmektedir. Benzer şekilde Dijkstra kullanıldığında ise 1 kere iş değiştirip, 1 kere beklemektedirler. Bu durum, en çok kenar hücrelerine sahip sınır hücre kriteri ile en yakın sınır hücre kriteri arasında her iki yol planı için de 20 dakika gibi çok önemli bir fark ortaya çıkarmaktadır. Bu kriterde robotlar sürekli en yakın hedefe gitmeyi amaçladıkları için ilk karşılaştıklarında haritalarını birleştirmekte ve koordine olarak birbirlerinden ayrılmaktadırlar. Bu durum, A* için birinci robotun 16. ikinci robotun 14. hedefinde (Şekil 5.13) açıkça gözükmemektedir. Bu noktadan sonra robotlar hala birbirlerinin haberleşme menzilleri içinde kalmakta ve haritalarını güncellemektedirler. İkinci robotun 18. hedefine ulaştığında haritası güncel olduğu için sol tarafa yönelmektedir. Böylece ikinci büyük odanın sağında kalan koridor parçasının sadece birinci robot tarafından keşfedilmesi açıklanmaktadır. Eğer ikinci robot haritası güncellenmemiş olsaydı ikinci robotta birinci robot gibi davranarak sağ tarafa yönelecek, diğer robot ile karşılaşacak ve zaman kaybedecekti. Şekil 5.15'te bu durum verilmektedir.



Şekil 5.15: Robotların h Menzil içinde olduklarında hedef seçimi

Dijkstra için ise birinci robot 5. ve ikinci robotun 4. hedefine ulaştığında robotlar ayrılmaktadır (Şekil 5.14). Bu noktadan sonra robotlar birbirlerinin haberleşme menzillerinden çıktıkları için haritalarını güncelleyememektedirler. Birinci robot 10.

hedefine geldiğinde haritası güncel olmadığı için sağ tarafa yönelmiştir. Oysaki aynı anlarda ikinci robot 15. hedefine doğru gitmekte ve çoktan ikinci büyük odanın sağında kalan koridor parçasını keşfetmişti. Şekil 5.16'da bu durum verilmektedir.

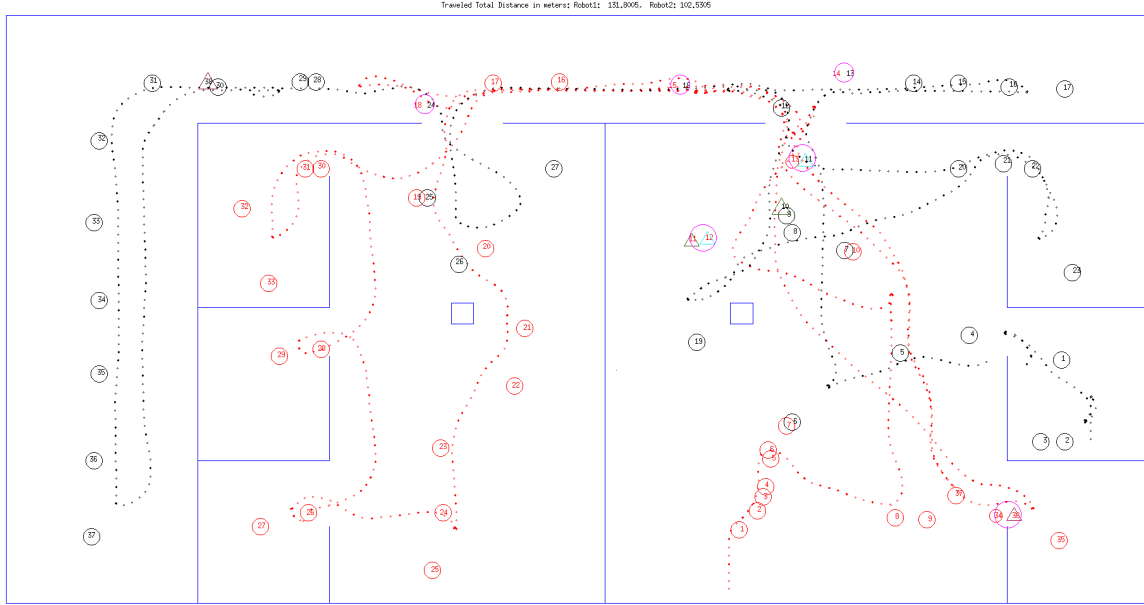


Şekil 5.16: Robotların $hMenzil$ dışında olduklarında hedef seçimi

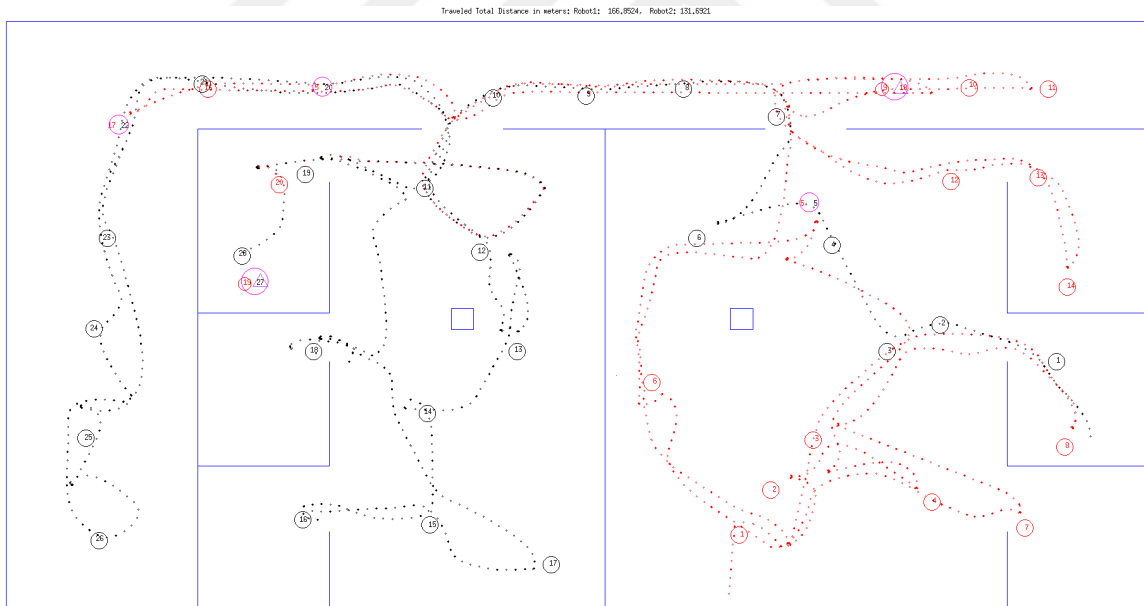
Sonuç olarak, en yakın sınır hücre kriteri kullanıldığında robotlar keşfin ilk anlarında $hMenzil$ içinde bulunmakta, koordine olmakta ve ayrılmaktadırlar. Keşfin büyük kısmı tamamlandıktan sonra tekrar karşılaşmakta ve koordine olarak görevlerini tamamlamaktadırlar. Böylece robotlar daha az mesafe katetmekte ve keşif süresi kısalmaktadır.

Şekil 5.17 ve 5.18'de sırasıyla A* ve Dijkstra yol planları için birinci robotun 16,0.3,1.57 konumundan ve ikinci robotun ise 24,4,1.57 konumundan başladıkları durumdaki market sınır hücre kriterinin yaptığı hedef seçimleri görülmektedir. Robotların A* yol planını kullanıldığında 3 kere iş değiştirdikleri görülmektedir. Benzer şekilde Dijkstra kullanıldığında ise 1 kere iş değiştirmektedirler.

Market sınır hücre kriteri ise bir sonraki hedefi seçerken hem mesafeyi hem de o hedefe gidildiğinde kazanılacak olan yeni bilgileri hesaba katarak kararını vermektedir. Şekiller incelendiğinde robotların yukarıda bahsedilen iki kriterin birleşimi gibi çalıştığı



Şekil 5.17: A* market sınır hücre kriteri



Şekil 5.18: Dijkstra market sınır hücre kriteri

gözlemlenmektedir. Robotlar bazı noktalarda güncel haritalarında diğer robotun çalıştığı alanda mesafe uzak olmasına rağmen daha kârlı işler bulabilmektedirler. Bu kriterde, en çok kenar hücrelerine sahip sınır hücre kriterinde olduğu gibi çok görev değişimi yaşanmamasının sebebi robotların diğer robotların çalışma alanlarına gittiklerinde o robotun başka bir

bölgeyi keşfetmesi ile açıklayabiliriz. Bu durumlarda genellikle robotlar tekrar haberleşme menziline girmekte ve bilgilerini güncelleyerek yeniden koordine olabilmektedirler.

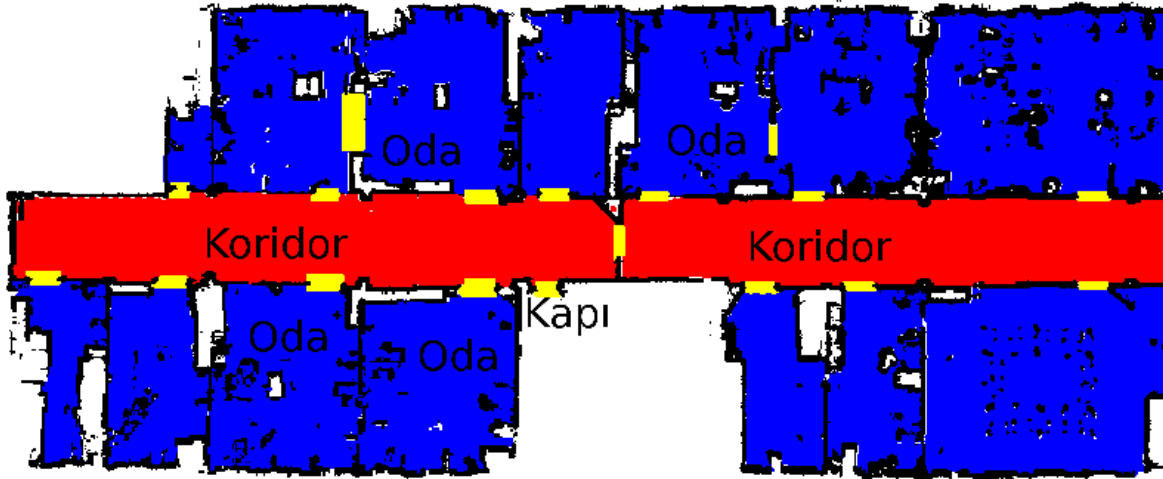
Son olarak robotların başlangıç konumlarının koordinasyonları üzerindeki etkiler tartışılmalıdır. Çizelge 5.1'de keşif süresi ve katedilen mesafe olarak verilen sonuçların yanı sıra robotların kaç kere bekledikleri veya hedef değiştirdikleri de bu konuda bazı fikirler vermektedir. Çizelge 5.2'de ise robotların keşif süresince bekleme ve hedef değiştirme miktarları verilmektedir. Konum 1 birinci robotun 2.5, 0.3, 1.57 ve ikinci robotun ise 24.8, 13.2, 3.14, konum 2 birinci robotun 16, 0.3, 1.57 ve ikinci robotun ise 24, 4, 1.57, konum 3 birinci robotun 9, 7, 1.57 ve ikinci robotun ise 11, 7, 1.57 konumundan başladığı durumları anlatmaktadır. Çizelge incelendiğinde market ve en çok kenar hücresi kriterlerinin robotların başlangıç noktalarından etkilendiği görülmüştür. Bu kriterler için robotların ya birbirinin sürekli haberleşme menzili içinde olması ya da birbirleriyle sadece son anda karşılaşmaları durumunda verim alındığı görülmüştür. Bununla birlikte, en yakın hücre kriteri robotların başlangıç konumlarından etkilenmemekte ve en iyi sonuçları vermektedir.

Çizelge 5.2: Çarpışma Önlem Protokolü A* ve Dijkstra Sonuçları

Fiyatlandırma Kriteri	Konum	A*		Dijk	
		Değişim	Bekleme	Değişim	Bekleme
En çok kenar hücresi	1	1	0	1	2
	2	5	3	6	3
	3	5	3	4	3
En yakın	1	1	2	1	1
	2	2	2	1	1
	3	3	0	0	1
Market	1	0	0	1	0
	2	3	0	1	0
	3	3	0	1	2

6. ANLAMSAL TABANLI KEŞİF YÖNTEMİ

Geçmiş yıllarda gezgin robotlar ile yapılan çalışmalarda robotun sağladığı ham algılayıcı verileri sadece konumlandırma, seyrüsefer ya da haritalama gibi temel görevleri yerine getirmek amacıyla kullanılmaktaydı. Bununla beraber, son yıllarda robotun etrafını tanımasını, bulunduğu konumu başka konumlardan anlamsal olarak ayırmasını amaçlayan çalışmalar yapılmaya başlanmıştır. Genel olarak, bina içi ortamlarda robotun bulunabileceği 3 temel anlamsal konum oda, koridor ve kapı olarak tanımlanabilir (Şekil 6.1). Robotun bu anlamsal konumlardan hangisinde olduğuna karar vermesi konumlandırma, seyrüsefer ve haritalama gibi temel problemlerin çözümünde verimliliği artıracak gibi servis görevlerinde, yaşlı ya da hasta bakımı ve arama kurtarma gibi uygulamalarda kullanılacak robotların insanlar tarafından yapılan istekleri daha iyi anlayıp yerine getirmelerini sağlayabilir.



Şekil 6.1: Üç anlamsal sınıf: Oda, Koridor ve Kapı (Mozos vd. 2005).

Algılayıcılardan elde edilen veriler ile nesnelerin bulunması bilgisayar ile görüş topluluğunun uzun yıllardır çalıştığı konulardan biridir. Robotik topluluğu ise ortam etiketleme ve anlamsal sınıflandırma konularını 2000'li yılların ortalarından itibaren yoğun olarak ele almıştır. Yapılan ilk çalışmalarda, genellikle kameradan alınan ve daha önceden etiketlenen veriler ile eğitim yapılmakta daha sonra bu eğitim kısmında üretilen bilgiler kullanılarak testler yapılmaktaydı (Pronobis vd. 2006). Benzer şekilde, Vasudevan ve diğerleri (Vasudevan vd. 2007) ise anlamsal konumları belirlemek için olasılıksal bir gösterim kullanmışlardır. Bunu yapabilmek için görüntü verisinden kitap, kupa ve sandalye

gibi nesnelere sınıflandıran ve bunların hangi odada (ofis, mutfak) bulunabileceği üzerine olasılıksal bir hipotez kullanan yöntem önermişlerdir. 2009 yılında Wu ve diğerleri yaptıkları çalışmada görsel yer kategorizasyon problemini otonom robot platformu ile toplanan görsel verilerden yararlanarak bir yerin anlamsal kimliğinin belirlenmesi olarak tanımlamışlardır. 6 farklı evden aldıkları verileri kullanarak bulunulan noktanın evin hangi bölümü olduğuna karar vermeye çalışmışlardır (Wu vd. 2009). Benzer bir çalışmada, bir konumda elde edilen görsel veride bulunan nesnelere sıklıkları olasılıksal olarak incelenmekte ve bu konuma benzer diğer konumlardan alınan veriler ile kümelenecek konumunu otomatik olarak belirlemektedir (Viswanathan vd. 2009). Son olarak, Eich ve diğerleri 3 boyutlu nokta bulutu kullanarak nesnelere biçimlerine karar vermeye çalışmaktadırlar (Eich vd. 2010). Görsel verileri kullanarak bir noktanın anlamsal sınıfını bulmaya çalışan yaklaşımlar, değişik ışıklandırma ve hava koşulları gibi verinin farklılaşacağı bir çok zorlu durumu göz önüne alarak geliştirmeler yapmışlardır. Bununla birlikte, afet sonrası bina içi ortamlarda hala bu görsel yaklaşımların verimli olması mümkün değildir. Örneğin, bu ortamlarda elektrik kesintisi olduğu için ışık olmayabilir. Benzer şekilde, toz ya da duman yüzünden kameranın görüş alanı azalabilir. Dolayısıyla, tez çalışmasının kapsamı göz önüne alındığında lazer mesafe algılayıcı ile anlamsal sınıfların belirlenmesi uygun olacaktır. Daha sonra, bulunan bu anlamsal sınıflar keşif hedeflerinin belirlenmesinde kullanılacaktır.

Bu bölümde, ilk olarak geçmiş çalışmalarda yer alan belli başlı anlamsal sınıflandırma yaklaşımları, kapı bulma yaklaşımları ve anlamsal sınıfları keşif ve haritalamada kullanan yaklaşımlar incelenecektir. Daha sonra, gerçekleştirilen yaklaşım alt bölümünde, K-ortalama, Öğrenen Vektör Nicelleme (ÖVN) (Learning Vector Quantization) ve Markov model kullanılarak bir noktanın anlamsal sınıfına nasıl karar verildiği detaylı olarak örneklerle verilecektir. Kapıların lazer mesafe algılayıcı ile tespit edilmesi için kullanılan kural tabanlı bir yaklaşım anlatılacaktır. Ayrıca, tek ve çok robotlu keşif yaklaşımında anlamsal sınıflandırma bilgilerinin kullanılabilmesi için yapılan çalışmalar verilecektir. Son olarak, anlamsal sınıflandırma yaklaşımının verimliliği, kapı bulma yaklaşımının etkileri ve anlamsal sınıflandırma kullanılarak yapılan hedef seçimleri sonunda keşif verimliliği tek ve çok robotlu sistemler için analiz edilecektir.

6.1 Geçmiş Çalışmalar

6.1.1 Anlamsal sınıflandırma ve haritalama yaklaşımları

Galindo ve diğerleri yaptıkları çalışmada bir renkli kamera ve lazer mesafe algılayıcısı kullanmışlardır. Bu yaklaşımda, lazer kullanılarak ortamın metrik haritası elde edilmiş, daha sonra bu harita gri tonlama (gray-scale) gibi düşünülüp büyük geniş alanlar

görüntü işleme yöntemleri ile belirlenmiştir. Belirlenen bu bölgeler, bölütleme yapılarak birbirinden ayrılmış ve oda olarak adlandırılmıştır. Bu odalar arasındaki ilişkiler kullanılarak ortamın topolojik haritası elde edilmiştir. Son aşamada, odada kamera yardımıyla bulunan nesnelere ile bulunabilecekleri odalar eşleştirilerek anlamsal harita çoklu hiyerarşik bir biçimde oluşturulmaktadır (Galindo vd. 2005).

Mozos ve diğerleri yaptıkları çalışmada sadece lazer mesafe algılayıcılarından aldıkları verileri kullanmışlardır. Bina içi ortamlarda temel olarak oda, koridor ve kapı anlamsal sınıflarının bulunabileceği varsayımını yapmışlardır. 360 adet lazer uzaklık ölçümünü kullanarak basit öznitelik (feature) kümelerini oluşturmuş, daha sonra bu kümeleri sırasal (sequential) AdaBoost adını verdikleri denetimli eğitim algoritmasıyla sınıflandırmışlardır. Önerdikleri yaklaşımı hem benzetim ortamında hem de gerçek robotlar ile birden çok ortamda test etmişler ve verimliliğini göstermişlerdir (Mozos vd. 2005). Bu noktadan sonra elde ettikleri anlamsal sınıflara olasılıksal gevşetme etiketlemesi (probabilistic relaxation labeling) tabanlı bir bölütleme uygulamışlardır. Elde edilen farklı anlamsal sınıftaki bölgeler ve onlar arasındaki ilişkileri kullanarak ortamın topolojik haritasını bir çizge olarak elde etmişlerdir (Mozos ve Burgard, 2006). Yazarlar, bir diğer çalışmalarında robotların belirli sınıflardan belirli sınıflara geçebileceklerini ve bu bilgiden yararlanılması gerektiğini vurgulamışlardır. Örneğin, robot bir odadan direkt olarak koridora geçememekte ve mutlaka kapı sınıfındaki bir noktadan geçmelidir. Bu bilgileri kullanmak için robotun ardışık hareketleri arasındaki ilişkiyi saklı Markov model kullanarak tanımlamışlardır (Martínez Mozos vd. 2007a). Yazarlar yaptıkları analiz sonucunda kapı sınıfının tanımlanmasının zor olduğuna vurgu yapmışlar ve bunun sebebi olarak da bu sınıftaki eğitim verisinin azlığı ve robotun kısmen oda ya da koridorda olduğu durumları göstermişlerdir.

Sousa ve diğerleri oda ve koridor anlamsal sınıflarını birbirinden ayırmak için 240 derecelik tarama yapabilen bir lazer mesafe algılayıcı kullanmışlardır. Bu algılayıcı 0.36 derecelik bir çözünürlükle 632 tane okuma elde etmekte ve bunları 14 adet basit öznitelige çevirmektedir. Daha sonra bu özellikleri destek vektör makinesi (Support Vector Machine, SVM) kullanarak sınıflandırmış ve anlamsal sınıfların tespit edilmesinde bu sınıfları kullanmıştır (Sousa vd. 2007).

Zender ve diğerleri, düşük derecede robot kontrolü ve algılamasından kullanıcı ile konuşarak iletişime geçen yüksek seviyeye kadar farklı seviyelerde alt sistemler içerecek bir yaklaşım önermişlerdir. Bu yaklaşımda, lazer kullanılarak metrik harita oluşturulmaktadır. Bir üst seviyede, robot bir ortamda ilerlerken belirli aralıklarla düğümler oluşturmakta ve böylece boş alanlardan oluşan ve düğümlerle ifade edilen bir seyrüsefer çizgesi elde edilmektedir. Bu seviyede ayrıca, düğümlerin sınıflarına lazer verisi

kullanılarak karar verilmektedir. Kapı sınıfındaki düğümlere robotun bir kapıdan geçtiğinde karar verilmektedir. Koridor ve odalar için ise basit geometrik öznitelikler kullanılmıştır. Topolojik harita seviyesinde seyrüsefer çizgesinde bulunan bağlı düğümler, kapı düğümleri ile ayrılarak farklı alanlarda toplanmaktadırlar. En üst seviyede ise robotun kamerası ile bulunan nesnelere en yakın düğümler ile ilişkilendirilmektedir. Bu şekilde, odalar ve nesnelere arasındaki bağlantı kurulmakta ve daha soyutlanmış bir şekilde bir harita elde edilmektedir. Ayrıca bu seviyede soyutlanmış harita kullanıcı ile iletişime geçmek için kullanılmaktadır (Zender vd. 2008).

Bir diğer çalışmada 3 boyutlu lazer mesafe algılayıcısı kullanılmaktadır. Bu çalışmada, 3 boyutlu tarama ile elde edilen sahne üzerinden düzlem çıkarma ve etiketleme yöntemleri kullanarak sahnedeki düzlemlere karar verilmektedir. Bu düzlemler, yer, duvar, tavan, kapı ya da bilinmeyen olarak sınıflandırılmaktadır. Benzer şekilde kapılar ise paralel ve dik olarak sınıflara ayrılmaktadır. Bu aşamada, 3 boyutlu tarama 2 boyutlu resime dönüştürülmekte ve bu resim üzerinden nesnelere belirlenmekte ve konumlarına karar verilmektedir. Son olarak, nesnelere konumları tekrar 3 boyuta dönüştürülmektedir (Nüchter ve Hertzberg, 2008).

Goerke ve Braun yaptıkları çalışmada kapı, koridor, boş alan, oda ve bilinmeyen olmak üzere 5 sınıf kullanmışlardır. Bu sınıfları belirlemek için 270 derecelik bir alanı yarım derece çözünürlükle tarayan bir lazer mesafe algılayıcıdan yararlanmışlardır. Öte yandan, yazarlar iki çeşit sınıflandırıcı kullanmışlardır. Bunlardan ilkinin kendileri belirlerken diğeri AdaBoost kullanılarak belirlenmiştir. Bu sınıflandırıcılar yardımıyla her sınıf için sıfır ile bir arasında değişen bir olasılık değeri belirlenmektedir. Bununla birlikte, ortamdaki değişimler, algılayıcı gürültüleri gibi etkilerden dolayı bir hücrenin anlamsal sınıfı farklı zamanlarda farklı belirlenebilmektedir. Dolayısıyla, bir hücrenin anlamsal sınıfına karar vermek amacıyla iki yaklaşım kullanmışlardır. İlk olarak, her andaki en yüksek olasılığa sahip olan sınıf belirlenmekte ve görev tamamlandığında en çok karar verilen sınıf o hücrenin anlamsal sınıfı olarak kabul edilmektedir. Diğer yaklaşım ise, bütün anlamlardaki olasılık değerlerini kullanarak son durumdaki anlamsal sınıfa karar vermektedir (Goerke ve Braun, 2009).

Lazer mesafe verisi kullanan bir diğer çalışmada lazer ölçümleri bir öznitelik seti elde etmek için kullanılmaktadır. Daha sonra, bu öznitelik seti hesaplamalı regresyon (logistic regression) tabanlı ikili sınıflandırıcının girdisi olarak düşünülmüştür. Bu süreç sonunda 3 baskın öznitelik belirlenmekte ve sınıflandırma kullanılmaktadır. Bir noktanın hangi sınıfa ait olduğuna karar vermek için o noktada alınan ölçümler olasılıksal olarak değerlendirilmektedir. Bu çalışmada, oda ve koridor noktaları anlamsal olarak sınıflandırılırken, kapı noktalarını bulmak için basit bir sezgisel önerilmiştir (Shi vd. 2010).

Nieto ve diğerleri yaptıkları çalışmada metrik haritada bulunan hücrelerin anlamsal sınıflarını belirlemek için ortamı bölgelere ayırmışlardır. Her bölge bir anlamsal sınıf ve çok değişkenli Gauss dağılım ile modellenmiştir. Her bir Gauss dağılımı rehber tarafından anlamsal sınıf belirlendiği andaki robotun algılayıcı verilerine göre belirlenmektedir. Bu çalışmada, bir rehber, robot ile birlikte ortamı gezmekte ve bulunulan konumun anlamsal sınıfını robota öğretmektedir. Bu anlarda robot, lazerden aldığı verileri kullanarak Gauss dağılımının ortalamasını ve kovaryansını hesaplamaktadır. Robot, seyrüsefer halindeyken sürekli olarak oluşturduğu Gauss dağılımları ile şu andaki lazer verilerinden karar verdiği anlamsal sınıfın doğru olup olmadığını kontrol etmekte ve gerekirse rehberden tekrar anlamsal sınıfı öğretmesini istemektedir (Nieto-Granda vd. 2010).

Pronobis ve diğerleri yaptıkları çalışmada hem kameradan hem de lazerden aldıkları verileri kullanarak ortamdaki anlamsal sınıfları belirlemeye çalışmışlardır. Kameradan aldıkları verileri aylarca farklı ışıklandırma koşullarında ve küçük ortam değişiklikleriyle elde etmişlerdir. Ayrıca, bu resimleri hem yerel hem de küresel tanımlayıcılar kullanarak etiketlemişlerdir. Bu farklı verileri birleştirmek için her özneteliği büyük bir sınıflandırıcı setiyle eğitmişlerdir. Bu eğitimin sonunda seçilen öznetelikler için bu seçimin ne kadar iyi olduğunu gösteren puanlar elde edilmiştir. Son olarak, bu puanlar SVM'ye girdi olarak kullanılarak bu çok farklı verilerin birleştirilmesi sağlanmıştır. Önerilen yaklaşım, eğitilen ortamdaki farklı bir ortamda test edildiğinde birleştirilen verilerin anlamsal sınıfı bulmakta oldukça gürbüz olduğu görülmüştür (Pronobis vd. 2009). Daha sonra bu çalışmalarını ortamın 4 katmanlı bir gösterimle ifade edilmesi için genişletmişlerdir (Pronobis ve Jensfelt, 2012). Bu gösterimin ilk katmanında, metrik harita bulunmaktadır. Bu katmanın üstünde, yerler (places), yollar ve yer tutucular (placeholders) olarak tanımlanan 3 kavramla ilgili bilgiler yer almaktadır. Ayrık alanlardaki boşluklar yerler olarak tanımlanmıştır. Bu yerler arasında robotun seyrüseferi ile yollar oluşmaktadır. Son olarak, yer tutucular ise robotun ziyaret ettiği ve ortamdaki bilinmeyen kısımlara yakın yerleri tanımlamaktadır. Kategori katmanında nesnelere, nirengi noktalarını ortamın geometrik ve görsel özelliklerini saklayan kategorik modeller bulunmaktadır. Bu modeller kullanılarak kavram katmanı oluşturulmaktadır. Bu çalışmada, kapıları bulmak için lazer verilerini girdi alan bir algoritma kullanılmıştır. Yeni odaları ise kapı verisine dayanarak belirlemişlerdir.

Shi ve Kodagoda yaptıkları çalışmada, geçmiş çalışmalardaki yöntemlerin aynı ortamda eğitilip aynı ortamda test edildiklerini ve bunun önerilen yöntemlerin geliştirilmesi önünde büyük bir problem olduğunu savunmuşlardır. Bunun üstesinden gelmek için yarı denetimli, SVM, şartlı rassal alana (Conditional Random Field, CRF) ve Genelleştirilmiş Voronoi Diyagramı yöntemlerini birleştiren bir yaklaşım önermişlerdir. Yarı denetimli yaklaşımlarda etiketlenmiş verilerle birlikte etiketlenmemiş verilerde kullanılmaktadır. Etiketlenmemiş olan veriler ortamın gerçek veri dağılımı ile ilgili

yardımcı bilgiler içermektedir. Yazarlar, 2 boyutlu lazer verilerinden 24 adet basit geometrik öznitelik belirlemişlerdir. Bununla birlikte, GVD ile elde edilen çizgedeki düğüm ve ayrıtlarla ilgili de 6 öznitelik kullanılmasına karar vermişlerdir. Önerdikleri yöntemi 6 farklı ortamda test etmişler ve denetimli eğitim yaklaşımlarına göre yarı denetimli eğitim yaklaşımının daha verimli olduğunu göstermişlerdir (Shi ve Kodagoda, 2013).

Mozos ve diğerleri yaptıkları çalışmada, 2 boyutlu bir lazeri dikey eksen etrafında döndürerek panoramik ve 3 boyutlu taramalar yapmışlardır. Daha sonra, bu taramaları 10 derece aralıklarla parçalara ayırmışlardır. Ayrılan bu parçalardaki mesafe ile dönen ışının yoğunluğunu gösteren yansıma verilerini yerel ikili örüntü histogramlarına dönüştürmüşlerdir. 10 derece aralıkla ayrılan bütün parçalardan gelen bu histogramları SVM'de kullanarak tek bir öznitelik vektörüne dönüştürmüşlerdir. Daha sonra, belirledikleri 5 sınıfı (koridor, mutfak, laboratuvar, çalışma odası ve ofis) %93 başarı oranı ile sınıflandırmışlardır. Yazarlar, bununla birlikte elde ettikleri 3 boyutlu lazer taramasını 10 derecelik parçalara ayırmadan, bütün olarak aynı prosedürü işlettiklerinde %96 başarı oranına ulaşmışlardır. Ancak, robotun zamanının kısıtlı olduğu durumlarda taramayı parçalara ayırmanın oldukça hızlı sonuç ürettiği gözlemlenmiştir. Benzer bir çalışmanın kamera ile yapılması durumunda ışıklandırma koşulları gibi kamera verisinden kaynaklanacak problemler ile karşılaşılmamıştır (Mozos vd. 2013).

Bir diğer araştırmada, yazarlar çeşitli AdaBoost yaklaşımlarını karşılaştırmışlardır. AdaBoost ikili sınıflandırma yapmaktadır. Bununla birlikte, bu çalışmada, oda, koridor ve kapı sınıfları kullanılmıştır. Dolayısıyla, bu üç sınıfı ayırmak için olasılıksal karar listeleri kullanılmıştır. Bu şekilde listedeki her AdaBoost her sınıfın başarı oranına karar verecektir. Son olarak, olasılıksal gevşetme etiketlemesi sınıflandırma hatalarını azaltmak amacıyla kullanılmıştır (Soares ve Araújo, 2014). 2015 yılında yapılan bir çalışmada ise anlamsal sınıflandırma ve haritalama ile ilgili yapılan çalışmaları algılama tipi, topolojik harita çıkartıp, çıkartmaması, iç veya dış ortamda yapılması ve hangi ölçekte ortamların kullanıldığı gibi ana başlıklara ayırıp incelenmektedir (Kostavelis ve Gasteratos, 2015).

6.1.2 Kapı bulma yaklaşımları

Kapı bulma konusundaki mevcut çalışmalar genellikle görsel yöntemler üzerine yoğunlaşmaktadır. Bu çalışmalarda kullanılan yöntemler piksel tabanlı ölçümlere dayanmaktadır ve genellikle robotun kamerası ile kapı konumu arasındaki mesafe ve yönelime karşı hassastır. Ayrıca robotun yöneliminin de kapılara paralel olması ve genellikle kapıların kapalı olması gerekmektedir. Monasterio ve diğerleri kapıları bulabilmek için Sobel süzgeci kullanmışlardır. Bu süzgeç sayesinde dik düz doğrular arasındaki mesafeyi ölçerek kapıları bulmayı amaçlamışlardır (Monasterio vd. 2002).

Benzer şekilde, Yang ve Tian, kenar ve köşe özelliklerini kullanarak kapıların geometrik karakterini ortaya çıkartmışlardır (Yang ve Tian, 2010).

Kapı konumlarını mesafe verisi kullanarak bulmayı amaçlayan bazı çalışmalar bulunmaktadır. Bu çalışmalar genellikle kapıların geometrik özelliklerinden faydalanmaktadırlar. Diaz ve diğerleri (Diaz vd. 2001) yaptıkları çalışmada, lazer verisi kullanarak kapı ve girişleri belirleyen algısal bir yaklaşım önermişlerdir. ElKaissi ve diğerleri (ElKaissi vd. 2007) yaptıkları çalışmada adına "Space and Edge Detection Laser Range Finder", SEDLRF, verdikleri bir kapı belirleme sistemi önermişlerdir. Hızlı kapı belirlemek amacıyla kapı kenarları arasındaki mesafeyi SICK LMS lazer algılayıcı ile algılamışlardır. Barber ve diğerleri (Barber vd. 2002) ise lazer mesafe verisinden kapı, duvar ve köşe gibi önemli bilgileri farkedebilmek için Hough dönüşümünü kullanmışlardır. Althaus ve Christensen (Althaus ve Christensen, 2003) ise bina içi ortamlarda topolojik harita çıkarmak için 16 ses üstü algılayıcıdan oluşan halka kullanmışlardır. Kapıları tanımlamak için önerdikleri yöntem kapı kenarları arasındaki boşluğu ses üstü mesafe algılayıcılar ile belirlemeyi temel almaktadır.

Başka bazı çalışmalarda kapı bulmak için hem görsel hem de mesafe verisi kullanılmıştır. Anguelov ve diğerleri (Anguelov vd. 2004) yaptıkları çalışmada hem lazer mesafe verisini hem de kameradan aldıkları görüntü verisini kullanmışlardır. Önerdikleri çalışmada, ortamı bölütlemek için kapı ve duvar nesnelere şekil, renk ve hareket özelliklerini içeren olasılıksal bir model kullanmışlardır. Benzer şekilde, Hensler ve diğerleri ise AdaBoost denetimli eğitim algoritmasını kullanarak Kapı ve kapı olmayanları ayırmayı amaçlamıştır. Bu çalışmada, dikey doğru çiftleri, kapı ve yer arasındaki aralık, duvarın renk ve dokusu ve kameranın hareketi gibi özellikler kullanılmıştır (Hensler vd. 2009). Bu çalışmaların hepsi duvarlardaki renk ve ışıklandırma koşullarının değişimlerine duyarlıdır.

Kapı bulma ayrıca ortamların anlamsal sınıflandırılması için geliştirilen yöntemler için de bir hedeftir. (Shi vd. 2010) çalışmasında lazer verisinden elde edilen anlamsal bilgi ile birlikte kapının genişliğini temel alan bir kapı algılayıcı gerçekleştirmişlerdir. Vasudevan ve diğerleri lazer algılayıcı verileri ile çizgileri tespit ederek kapı konumlarına karar vermeye çalışmışlardır (Vasudevan vd. 2007). Mozos ve diğerleri (Mozos vd. 2005) lazer mesafe ölçümlerinden basit geometrik özellikler çıkarmışlar ve bu özellikleri bir AdaBoost sınıflandırıcıyı eğitmek için kullanmışlardır. Çalışmalarında, kapı konumlarının doğru biçimde ayırt edilmesinin, hem eğitim verisinin kıtlığından hem de robotun kısmen oda veya koridorda, kısmen kapıda olduğu durumlarından ötürü zor bir iş olduğundan bahsetmişlerdir.

6.1.3 Anlamsal sınıflandırma tabanlı keşif yaklaşımları

Anlamsal sınıflardan yararlanarak keşif yapmayı amaçlayan ilk yaklaşımlardan biri Diaz ve diğerleri (Diaz vd. 2001) tarafından önerilmiştir. Bu çalışmada, ortamın yapısında bulunan kapı, koridor gibi yapılar sayesinde robotların koridorlarda gezinebilmesini, kapıları keşfedebilmesini ve odalara girebilmesini sağlamaktadır. Önerilen yaklaşımda iki temel davranış, koridor boyunca ilerle ve kapıya doğru git, tanımlanmaktadır. Bu davranışların birleştirilmesi ile robotlar bilinmeyen bir ortamın keşfini yapabilmektedir. Koridorların tanınmasında sadece koridorun genişliği ve açısına ihtiyaç duyulmakta ve bu bilgiler lazer ışınları kullanılarak elde edilmektedir. Kapıların tanınmasında kapının genişliği kullanılan robota göre seçilmektedir.

Stachniss ve diğerleri (Stachniss vd. 2006) binaların genelde koridorlardan yanyana dizilmiş odalar ve ofislere açılan yapılar olduğunu söyleyerek, keşif görevinde bu bilgiden yararlanılabileceğini savunmaktadır. Lazer uzaklık ölçerden alınan ölçümler yardımıyla bir noktanın koridor mu yoksa başka bir yer mi olduğuna saklı Markov modeli kullanılarak karar verilmektedir. Eğer bir hedef koridorda ise onun faydası odalardaki hedeflere göre daha yüksek olmaktadır. Böylece robotlar ortama daha iyi dağılmakta ve hızlıca birçok potansiyel hedef noktası buldukları için keşif süresi önemli ölçüde azalmaktadır.

Wurm ve diğerleri (Wurm vd. 2008) önerdikleri yaklaşımda robotlara atama yapmak amacıyla o ana kadar keşfedilmiş alanlar bölümlere ayrılmaktadır. Bu ayrışma sayesinde robotlar ortama verimli bir şekilde dağılmakta, tekrar gezilen alanlar azalmakta ve keşif süresi önemli ölçüde kısalmaktadır. Bununla beraber haritanın bölümlere ayrılması için bir çizgenin parçalanması yaklaşımı öne çıkmaktadır. Ortamın Voronoi çizgesi çıkarıldıktan sonra bu çizge kritik noktalar adı verilen ve kapı, dar koridor girişleri olarak tanımlanan yerlerden k ayrık kümeye parçalanmaktadır. Bu kümelerin herbiri robotlara atanabilecek bölümlerdeki düğümlerden oluşmaktadır. Aynı ana fikre sahip bir diğer çalışma ise (Stachniss vd. 2008) sunulmuştur. Bu çalışmada, hedeflerin seçimi esnasında hedefin anlamsal sınıfının ne olacağını tahmin etmek amacıyla saklı markov modeli kullanılmıştır. Hedef noktasına doğru sanal bir yörünge üretilmiş ve kısmi olarak bilinen haritada o noktada anlamsal sınıf tahmin edilmeye çalışılmıştır. Daha sonra, robotun odadan direk olarak koridora çıkamayacağı gibi genellikle bina içi ortamlarda kullanabileceğimiz bilgiler ile tahmin iyileştirilmiştir. Yazarlar, robot takımında 5 robottan az sayıda robot olduğunda standart koordinasyon yöntemleri ile aynı sonuçları elde ettiklerini belirtmişlerdir.

Galindo ve diğerleri yaptıkları çalışmada anlamsal bilginin robotların görev planlaması esnasında ne gibi faydalarının olabileceğini tartışmışlardır (Galindo vd. 2008).

İlk olarak, anlamsal bilgi, planlayıcının daha geniş bilgiye sahip olmasını ve daha karmaşık görevleri planlayabilmesini sağlamaktadır. Bununla birlikte, sahip olduğu anlamsal bilgi ile çıkarımlar yaparak bu bilgiyi kullanmayan bir planlayıcının yapamayacağı planları hazırlayabilmektedir. Son olarak, robotlar sahip oldukları anlamsal bilgi ve çıkarım yapma yeteneği ile kendilerine yeni görevler belirleyebilirler. Örneğin, havluların sadece banyoda olması gerektiği varsayımı yapılırsa, robot banyo dışında bir yerde havlu bulduğunda havluyu banyoya götürebilir.

Görüntü tabanlı keşif algoritması (Krishnan ve Krishna, 2010) melez bir anlamlandırma kullanmaktadır. Bu çalışmada, robot etrafındakileri metrik seviyeden anlamsal seviyeye kadar geniş bir kapsamda anlamlandırmaktadır. Robot bir bölgeyi keşfedip bu melez haritayı yaratmaktadır. Bu haritanın en üst katmanında bir çizge yer almaktadır. Bu çizgenin düğümleri laboratuvarlar ve koridorlar gibi anlamsal yapılardır. Bu çizgenin ayrıtları ise kapı ve birleşim noktaları gibi geçiş bölgelerini tanımlamaktadır. Orta katmanda ise anlamsal yapılar resimler kullanılarak belirlenmektedir. Keşif sırasında robot ilk önce bulunduğu anlamsal yapıyı tamamen keşfetmekte ve aynı zamanda diğer anlamsal yapıyı geçiş için kullanılacak olan geçiş bölgelerinin yerlerini belirlemektedir. Bu geçiş bölgeleri lazer mesafe algılayıcısı bilgileri ile belirlenmekte ve görüntü ile kontrol edilmektedir. Bir anlamsal yapının keşfi tamamlandığında robot diğer anlamsal yapıya geçiş bölgelerini kullanarak geçmektedir. Süreç gidilecek geçiş noktası kalmayınca sonlandırılmaktadır.

Aydemir ve diğerleri yaptıkları çalışmada esas olarak bilinmeyen bir ortamda belirli bir nesneyi aramaktadır (Aydemir vd. 2011). Bu nesneyi ararken aynı zamanda da ortamın keşfini sürdürmektedirler. Keşif sırasında hedefleri belirlerken 6 tane kriteri gözden geçirmektedirler. İlk olarak, hedef yeni bir yerde değil; hedef yeni bir yerde ama yeni bir odada değil; hedef yeni ve tek bir odada; hedef başka odalar ile bağlantılı olan bir odada; hedef direk olarak tek ve yeni odada; hedef direk olarak yeni bir odada ve bu odaya bağlı başka bir oda var. Bu kriterler sayesinde esasında hem ortamı keşfetmek hem de aradığı nesneyi bulmayı amaçlamaktadır.

Bu alt bölümde verilen geçmiş çalışmalar göz önüne alındığında anlamsal sınıfların belirlenmesinde genellikle görsel veri kullanılarak nesnelere bulunması ve bu nesnelere bulunabileceği yerlerin olasılıksal olarak değerlendirilmesi yolu izlenmiştir. Bununla birlikte, afet sonrası bina içi ortamlarda, elektrik kesintisi, toz ve duman bulutları gibi olumsuz koşullar yüzünden görsel tabanlı sınıflandırmanın yapılması zor olacaktır. Diğer yandan, lazer mesafe algılayıcısı bu olumsuz koşullardan etkilenmediği gibi görsel algılayıcılara göre daha uzun menzilli ve daha hassas ölçümler yapabilmektedir. Sonuç olarak, lazer mesafe algılayıcısı anlamsal sınıfların bulunması için veri sağlayacaktır.

Geçmiş çalışmalarda verilerin eğitimi için SVM, AdaBoost, CRF ve türevleri gibi denetimli yaklaşımlar kullanılmıştır. Bu çalışmaların tamamında ham lazer verileri belirli sayıda özniteliğin bulunması için kullanılmış ve denetimli eğitim yaklaşımlarına bu öznitelikler girdi olarak verilmiştir. Bu tez çalışması kapsamında K-ortalama denetimli olmayan sınıflandırma yöntemi ile denetimli Öğrenen Vektör Niceleme (ÖVN) sınıflandırma yöntemleri kullanılmıştır. Bu yöntemler ile bir ortamda eğitilen veriler hem aynı ortamda hem de başka ortamlarda test edilmiştir.

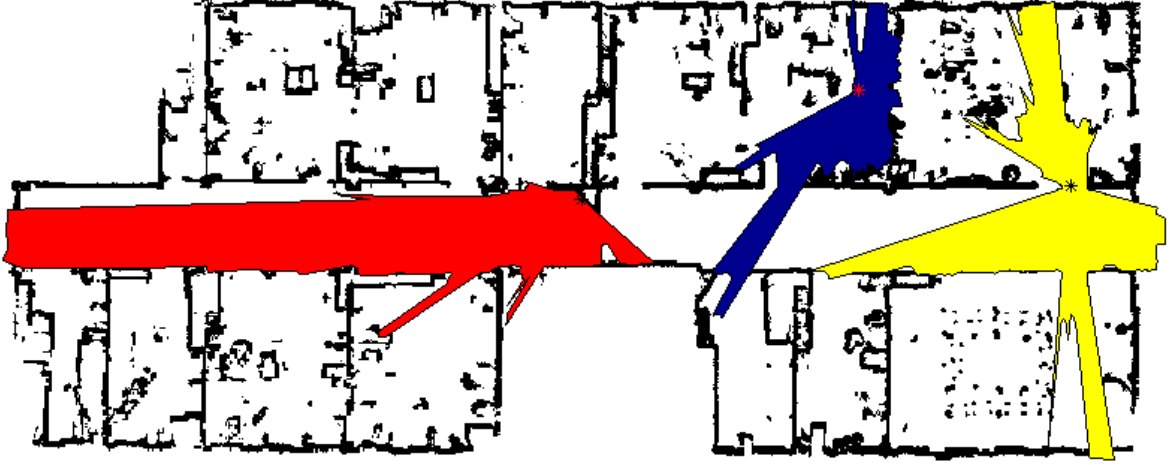
Geçmiş çalışmalar incelendiğinde lazer mesafe verisi kullanılarak kapıların bulunmasının zor bir problem olduğu açıktır. Denetimli ya da denetimli olmayan eğitim yaklaşımları eğitim verisinin azlığı ve robotun kısmen oda ya da koridorda olduğu durumlar nedeniyle kapıları yeteri kadar tanımlayamamaktadır. Bu durumun üstesinden gelmek için ilk önce yönelim normalizasyonu yapılmış ve daha sonra kural tabanlı bir kapı bulma algoritması gerçekleştirilmiştir.

Geçmiş çalışmalar incelendiğinde, elde edilen anlamsal konum bilgilerinin çok az yayında keşif görevinde kullanıldığı görülmektedir. Bu çalışmalarda da genellikle sadece oda ve koridorlar sınıflandırılmakta ve robotların başta koridorları gezmesi istenmektedir. Ayrıca robot keşif görevindeyken belirlenen hedefler bilinmeyen alanın içinde olduğu için bu konumun anlamsal sınıfının bilinmesi mümkün değildir. Bu noktada genellikle robotun o noktaya gittiği varsayılarak bilinen ortamın haritası kullanılarak hedefin bulunduğu noktanın anlamsal sınıfı belirlenmektedir. Bu tez çalışması kapsamında elde edilen anlamsal bilgiler keşif görevinde hedef seçiminde kullanılmıştır. Bu yaklaşımda, hedef noktaların anlamsal sınıflarına gizli Markov modeli kullanılarak karar verilmiştir. Daha sonra, robotlara oda veya koridor öncelikli keşif yaptırılması sağlanmıştır. Bu alt bölümden sonra gerçekleştirilen yaklaşım ve deneysel sonuçlar detaylı olarak verilecektir.

6.2 Gerçeklenen Yaklaşım

Bu çalışmada, bina içi ortamlarda lazer mesafe verisi kullanarak anlamsal sınıflandırma yapılması amaçlanmaktadır (Şekil 6.2). Oda, koridor ve kapıların boyutlarının, şekillerinin ortamdan ortama farklılık gösterebileceği bilinen bir gerçektir. Bu sebeple, genel olarak anlamsal sınıflandırmanın başarısı bu özelliklere bağlı olmaktadır. Dolayısıyla, robotun anlamsal konumunun belirlenmesinde bu özelliklerin hesaba katıldığı bir çok çalışma bulunmaktadır (Mozos vd. 2005) ve (Pronobis ve Jensfelt, 2012). Bununla birlikte, önerilen bu çalışmada, 362 ham lazer mesafe okuması sınıflandırmaya girdi olan özellikler olarak kullanılmıştır. Anlamsal sınıfların farklı şekilleri ve boyutlarını ele almak için yönelim normalizasyonu uygulanmıştır. Yönelim normalizasyonu, aynı konumda

bulunabilecek olan farklı anlamsal sınıfların elenmesine yardımcı olmaktadır. Bu şekilde, anlamsal sınıflandırmanın başarısının iyileşmesi beklenmektedir.



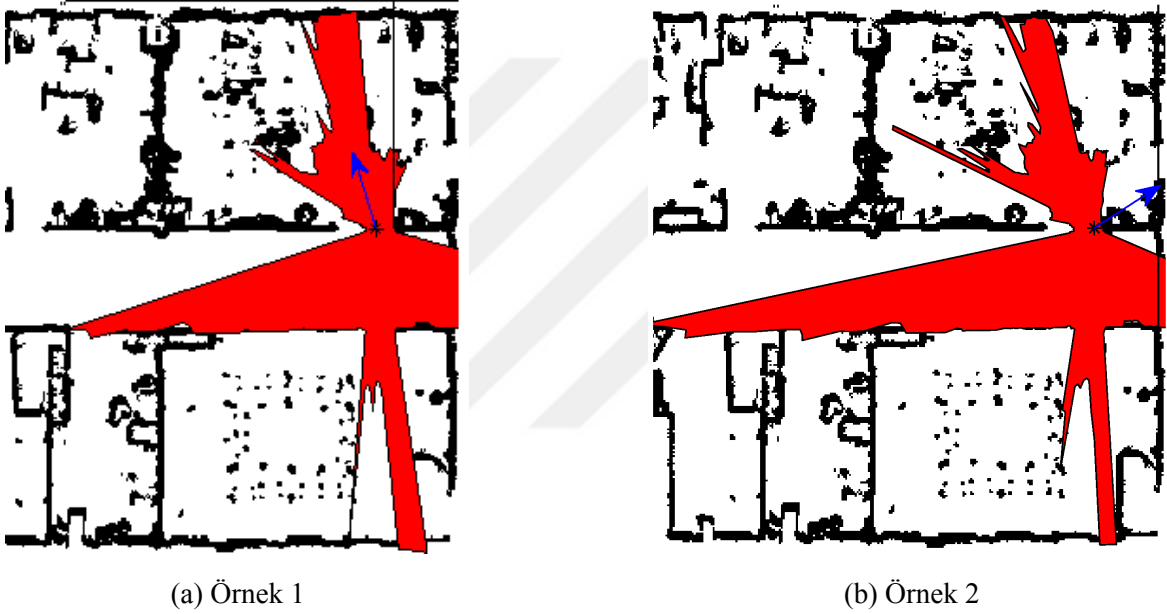
Şekil 6.2: Oda, Koridor ve Kapı sınıflarından lazer verilerine örnekler (Mozos vd. 2005).

Bu çalışmada, K-Ortalama ve Öğrenen Vektör Nicemleme (ÖVN) sınıflandırma yöntemleri küme merkezlerini elde etmek için kullanılmıştır. K-Ortalama (MacQueen, 1967), temel olarak veriyi kümelere bölmektedir. Bu noktadan sonra, gelen her yeni veri ile birlikte küme merkezleri güncellenmekte ve gelen veri kendine en yakın olan küme merkezine ait kümeye atanmakta ve onun kimliğini almaktadır. Bu kimlik daha sonra anlamsal sınıflandırmada kullanılmaktadır. Oda, koridor ve kapı sınıflarını belirleyebilmek amacıyla olasılıksal bir yöntem uygulanmaktadır. ÖVN'de benzer şekilde veriyi küme merkezleri etrafında toparlamaktadır. Ancak K-Ortalamadan farklı olarak, ÖVN'de küme merkezlerinin sınıfları önceden belirlenmektedir. Dolayısıyla, ÖVN daha önceden sınıfı belirlenmiş küme merkezlerini kullanarak sınıflandırma yapmaktadır. K-Ortalama ve ÖVN sınıflandırma yöntemleri oda ve koridor anlamsal sınıflarını başarılı bir şekilde belirlemektedir (Kaleci vd. 2015a). Kapıların başarı oranı istenilen seviyeden çok uzak olduğu için ilk önce yönelim normalizasyonu yapılmış ve daha sonra kural tabanlı bir kapı bulma algoritması gerçekleştirilmiştir.

Robotik uygulamalarda genellikle robotun oda veya koridorda çalıştığı gözlemlenmektedir. Dolayısıyla, bir anlamsal sınıftan diğerlerine geçişlere çok seyrek olarak rastlanmaktadır. Bu bilgi, anlamsal sınıflandırmanın başarısını artırmak amacıyla kullanılabilir. Markov model, birbirini takip eden anlamsal sınıflar arasındaki ilişkiyi tanımlamak için kullanılabilir. Benzer şekilde, koşullu olasılık kullanılarak da bu ilişki tanımlanabilmektedir. Önerilen yaklaşım, koşullu olasılık ve Markov model kullanılarak daha yüksek sınıflandırma başarısı elde etmek amacıyla genişletilmiştir.

6.2.1 Yönelim normalizasyonu

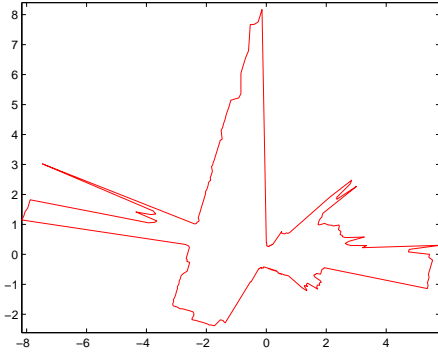
Önceki kısımda belirtildiği üzere (Mozos vd. 2005) eğitim verisinin azlığı ve robotun kısmen oda ya da koridorda olduğu durumlar, kapı anlamsal sınıfının oda ve koridor anlamsal sınıflarından ayrıştırılmasını zorlaştırmaktadır. Buna ek olarak, robotun farklı yönelimlere sahip olması kapı sınıfının bulunmasını daha da zor bir problem haline getirmektedir. Şekil 6.3a ve 6.3b robotun aynı konumda olduğu ancak farklı iki yönelime sahip olduğu durumdaki lazer okumalarını göstermektedir. Örnek 1'de robot yaklaşık olarak 100 derecelik bir yönelime sahipken Örnek 2'de robotun yönelimi yaklaşık olarak 30 derecedir.



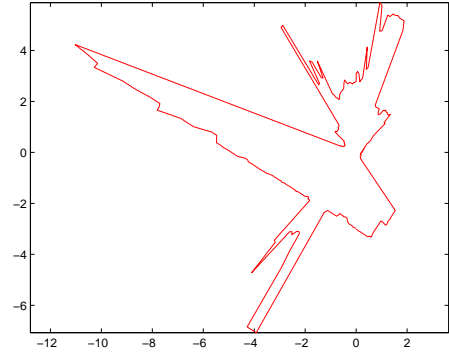
Şekil 6.3: Aynı konumda farklı yönelimde iki robot lazer okuması örnekleri

Şekil 6.4a ve 6.4b sırasıyla Örnek 1 ve Örnek 2 için robotun yerel koordinat sistemine göre lazer verisini göstermektedirler. Sonuç olarak, Örnek 1 ve Örnek 2 aynı konumda yer almaktadırlar ve küresel koordinat sistemi göz önüne alındığında (Şekil 6.3a ve 6.3b'ye bakıldığında) benzer lazer taramasına sahiptirler. Şekil 6.4c ve 6.4d ise lazer okumalarını lazer mesafe algılayıcının indeksine göre vermektedir. Bununla beraber, Şekil 6.4c ve 6.4d'de görüldüğü üzere yönelimlerinin farklı olması sebebiyle iki örnek iki farklı lazer taraması elde etmiş gibi gözükmemektedir.

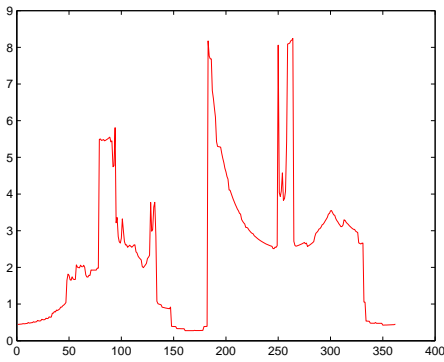
Açıktır ki, robot aynı konumda dahi olsa yöneliminin farklı olması lazer indeksine göre yapılan okumaların farklı olmasına sebep olmaktadır. Bununla birlikte, K-Ortalama ve ÖVN sınıflandırma yöntemleri veriyi kümelendirirken lazerin indeksine göre olan mesafe



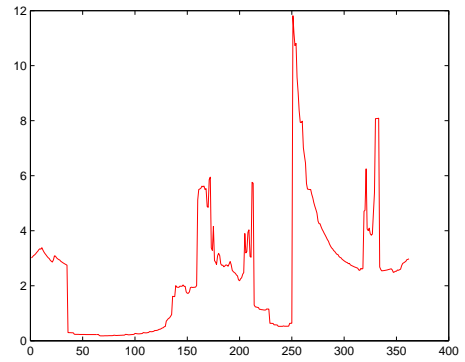
(a) Örnek 1 için lazer taraması



(b) Örnek 2 için lazer taraması



(c) Örnek 1 için indekse göre lazer okumaları



(d) Örnek 2 için indekse göre lazer okumaları

Şekil 6.4: Örnek 1 ve Örnek 2 için yerel ve indekse göre lazer verileri

ölçümlerini kullanmaktadırlar. Bu sebeple, kapı sınıfının bulunma başarısını artırmak amacıyla, yönelim normalizasyonu tüm lazer mesafe ölçümlerine dairesel kaydırma (circular shifting) yöntemi ile uygulanmıştır (Algoritma 11).

Algoritma 11: Dairesel Kaydırma Algoritması

Girdi: Lazer mesafe verisi $laserData[362]$ and robotun yönelimi $angle$

Çıktı: Normalize edilmiş lazer mesafe verisi $shiftedLaserData$

$index \leftarrow 0;$

forall the $i : 362 - angle$ to 362 do

$circularShiftedLaserData[index] \leftarrow laserData[i];$

$index \leftarrow index + 1$

end

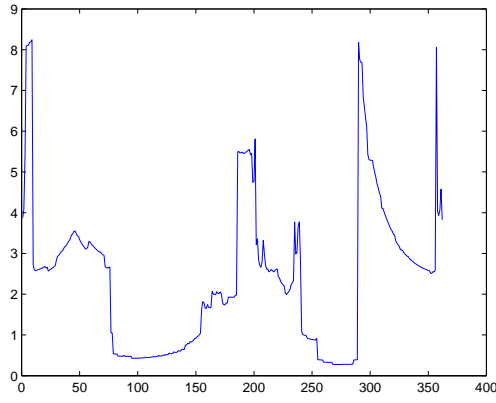
forall the $i : 1$ to $362 - angle$ do

$circularShiftedLaserData[index] \leftarrow laserData[i];$

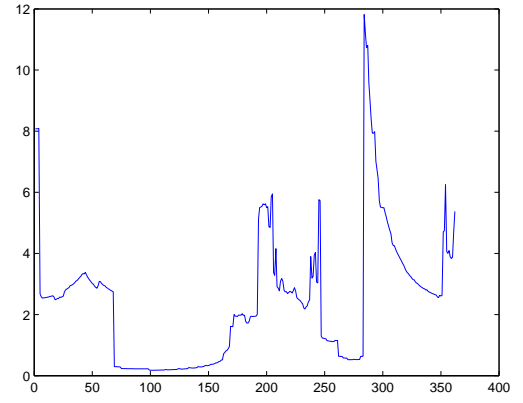
$index \leftarrow index + 1$

end

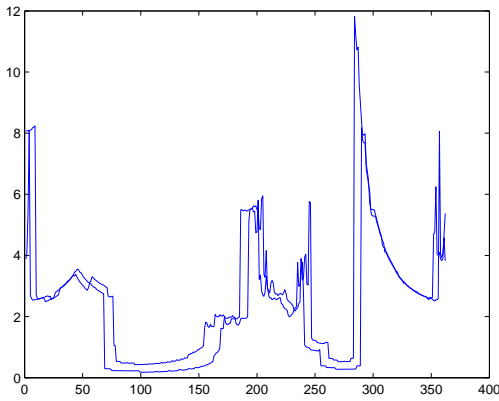
Bu algoritmanın nasıl çalıştığını anlamak amacıyla Şekil 6.4c-6.4d ve Şekil 6.5a-6.5b'yi gözden geçirelim. Örnek 1 için robot yöneliminin yaklaşık olarak 100 derece olduğu bilinmektedir. Dolayısıyla, bu örnek için son 100 lazer indeksi okuması yeni oluşturulan dizinin ilk 100 elemanı olmaktadır. Daha sonra kalan indekslerdeki okumalarda yeni dizinin ilk 100 elemanından sonraki elemanlara sırasıyla yerleştirilmektedir. Benzer prosedür Örnek 2'deki lazer okumalarına uygulanarak normalize edilmiş lazer verileri elde edilmektedir. Yönelim normalizasyonu uygulandıktan sonra robotun yöneliminin sanal olarak sıfıra getirildiği düşünülmektedir. Sonuç olarak, Şekil 6.5c ve 6.5d'de verilen durumlarda robotun yönelimi hemen hemen aynı olacaktır. Dolayısıyla, aynı lazer indeksine sahip okumalarda yaklaşık olarak aynı mesafe ölçümlerine sahip olacak ve yönelim normalizasyonu daha kolay bir kapı sınıflandırması sağlayacaktır.



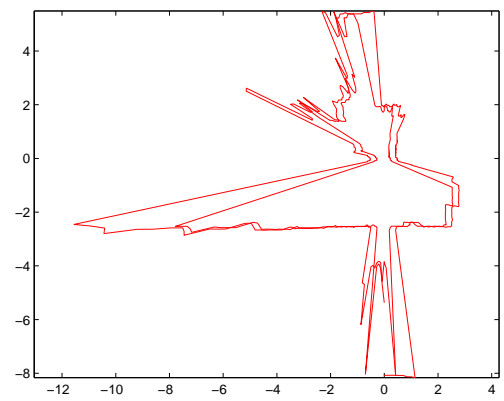
(a) Örnek 1 normalize edilmiş



(b) Örnek 2 normalize edilmiş



(c) Normalize edilmiş lazer verileri



(d) Normalize edilmiş lazer taramaları

Şekil 6.5: Yönelim Normalizasyonu

6.2.2 K-Ortalama (KO) ile sınıflandırma

KO sınıflandırma yöntemi veriyi K kümeye ayırmayı amaçlamakta ve bunu yaparken her veriyi kendisine en yakın olan kümeye yerleştirmektedir (MacQueen, 1967). KO tekrarlamalı olarak küme merkezlerine karar vermektedir. Küme merkezleri yakınsadığında küme merkezleri ile her verinin ait olduğu küme kimliği bilgileri elde edilmektedir. Üç anlamsal sınıf için küme merkezleri örnekleri Şekil 6.6'da verilmiştir. Her sınıfa ait küme merkezi diğer iki sınıfa ait olanlara göre göreceli olarak yüksek olasılığa sahip olanlardan seçilmiştir. Şekilde görüldüğü gibi oda ve kapı anlamsal sınıflarına ait olan küme merkezleri birbirine benzemektedir. Bu benzerlik oda ve kapı sınıflarının birbirinden ayrılmasını zorlaştırmaktadır.

KO yönteminden elde edilen bilgiler Bayes tabanlı olasılıksal bir yöntem ile değerlendirilerek robotun üç anlamsal sınıftan hangisinde bulunduğuna karar verilmektedir. Şimdi bu durumu Bayes kuralını kullanarak matematiksel olarak ifade etmeye çalışalım.

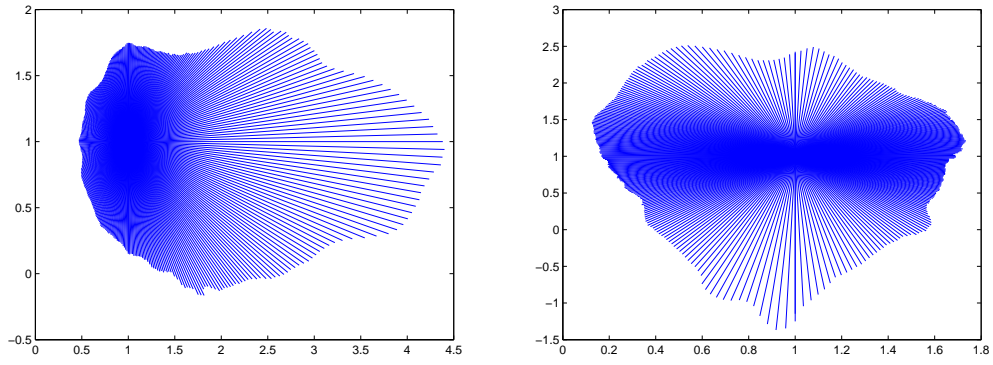
$$P(x_i|c_k^i) = \frac{P(c_k^i|x_i)P(x_i)}{P(c_k^i)} \quad (6.1)$$

Denklem 6.1'de x_i robotun i 'inci anda bulunabileceği üç anlamsal sınıftan birini göstermektedir. Denklem'deki c_k^i ifadesi ise i 'inci anda robottan elde edilen lazer mesafe verilerinin k 'inci küme merkezine en yakın olduğunu anlatmaktadır. Sonuç olarak, $P(x_i|c_k^i)$, i 'inci anda lazer mesafe verilerinin en yakınında k 'inci küme merkezinin olduğu biliniyorsa bu anda robotun bulunabileceği anlamsal sınıfların olasılıklarını ifade etmektedir.

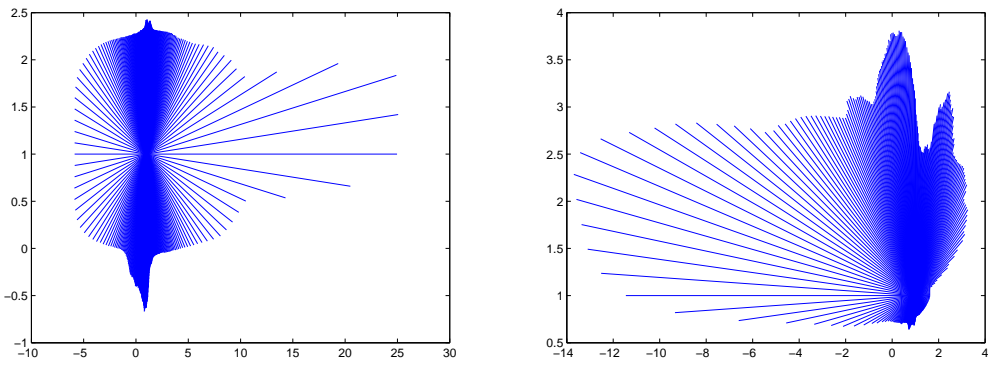
$P(x_i|c_k^i)$ olasılıklarını bulabilmek için $P(c_k^i|x_i)$ ve $P(c_k^i)$ olasılıkları önceden KO sınıflandırma yönteminde küme verilerini oluştururken kullanılan eğitim verisinden hesaplanmaktadır. $P(x_i)$ önsel (prior) olasılığı ise başlangıç durumuna göre belirlenmektedir. Genellikle, robotun ortamla ilgili herhangi bir bilgiye sahip olmadığı varsayılarak düzgün dağılımla belirlenir.

6.2.3 Öğrenen vektör nicemleme (ÖVN) ile sınıflandırma

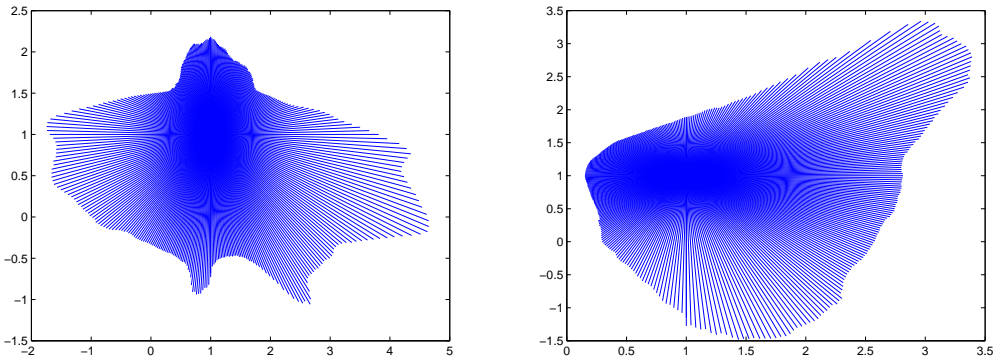
Öğrenen Vektör Nicemleme yöntemi, veriyi belli merkezler etrafında kümelemeyi hedeflemektedir. KO'nun aksine ÖVN denetimli eğitim yapan bir yaklaşımdır. ÖVN'de kümelemeye başlanmadan önce her grup merkezi belli bir sınıfa atanmıştır. Merkezler güncellenirken, eldeki eğitim verisinin sınıfına göre merkez veriye yaklaştırılmakta ya da uzaklaştırılmaktadır. Küme merkezleri yakınsadığında her merkezin anlamsal sınıfı belli olmakta ve ek bir işlem yapmaya gerek kalmamaktadır. Test aşamasında, lazer verisine en



(a) Oda



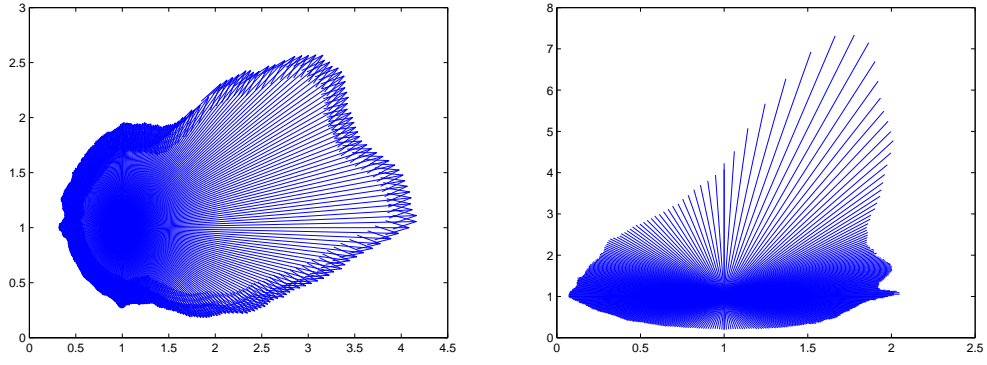
(b) Koridor



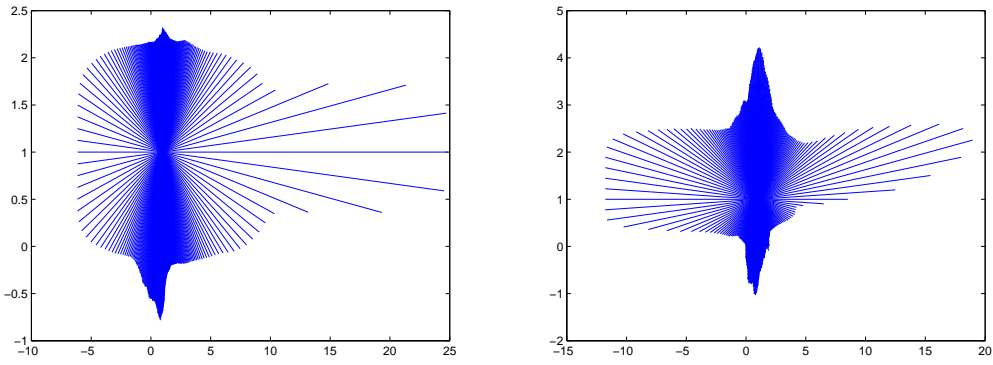
(c) Kapı

Şekil 6.6: K-Ortalama yöntemi için küme merkezi örnekleri

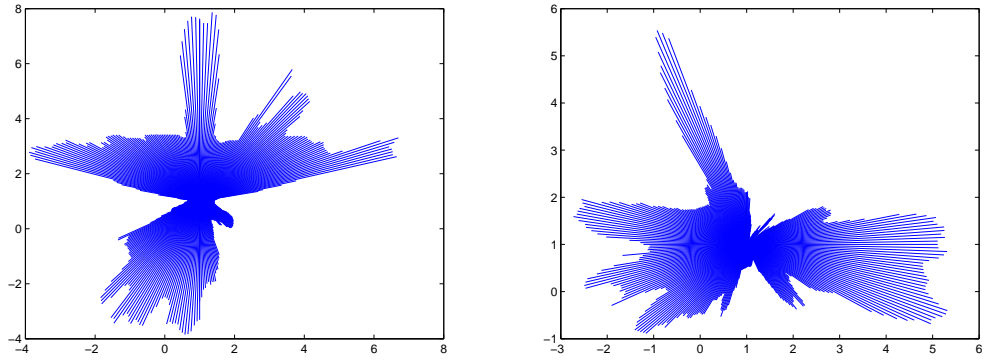
yakın merkezin atandığı sınıf, o lazer verisinin anlamsal sınıfı olarak kabul edilir. Üç anlamsal sınıf için küme merkezleri örnekleri Şekil 6.7'de verilmiştir. Şekilde görüldüğü gibi KO'nun aksine küme merkezleri ait oldukları sınıfların geometrik karakteristiklerini yansıtmaktadırlar.



(a) Oda



(b) Koridor



(c) Kapı

Şekil 6.7: ÖVN yöntemi için küme merkezi örnekleri

6.2.4 Markov model

Robotik uygulamalarda genellikle robotun oda veya koridorda çalıştığı gözlemlenmektedir. Dolayısıyla, bir anlamsal sınıftan diğerlerine geçişlere çok seyrek olarak rastlanmaktadır. Bu bilgi, anlamsal sınıflandırmanın başarısını artırmak amacıyla kullanılabilir. Markov model, birbirini takip eden anlamsal sınıflar arasındaki ilişkiyi tanımlamak için kullanılabilir. Markov model 1966 yılında L. E. Baum (Baum, 1966)

tarafından geliştirilmiş ve konuşma, el yazısı, jest, mimik tanımlamadan robotun anlamsal konumunu tahmin etmeye kadar geniş bir alanda kullanılmıştır (Martínez Mozos vd. 2007b).

Robotun anlamsal konumunu tanımlamak için bir ardsal (posterior) $Bel(x_i)$ aşağıda verildiği gibi hesaplanmaktadır:

$$Bel(x_i) = \alpha P(c_k|x_i) \sum_{x_{i-1}} P(x_i|x_{i-1}) Bel(x_{i-1}) \quad (6.2)$$

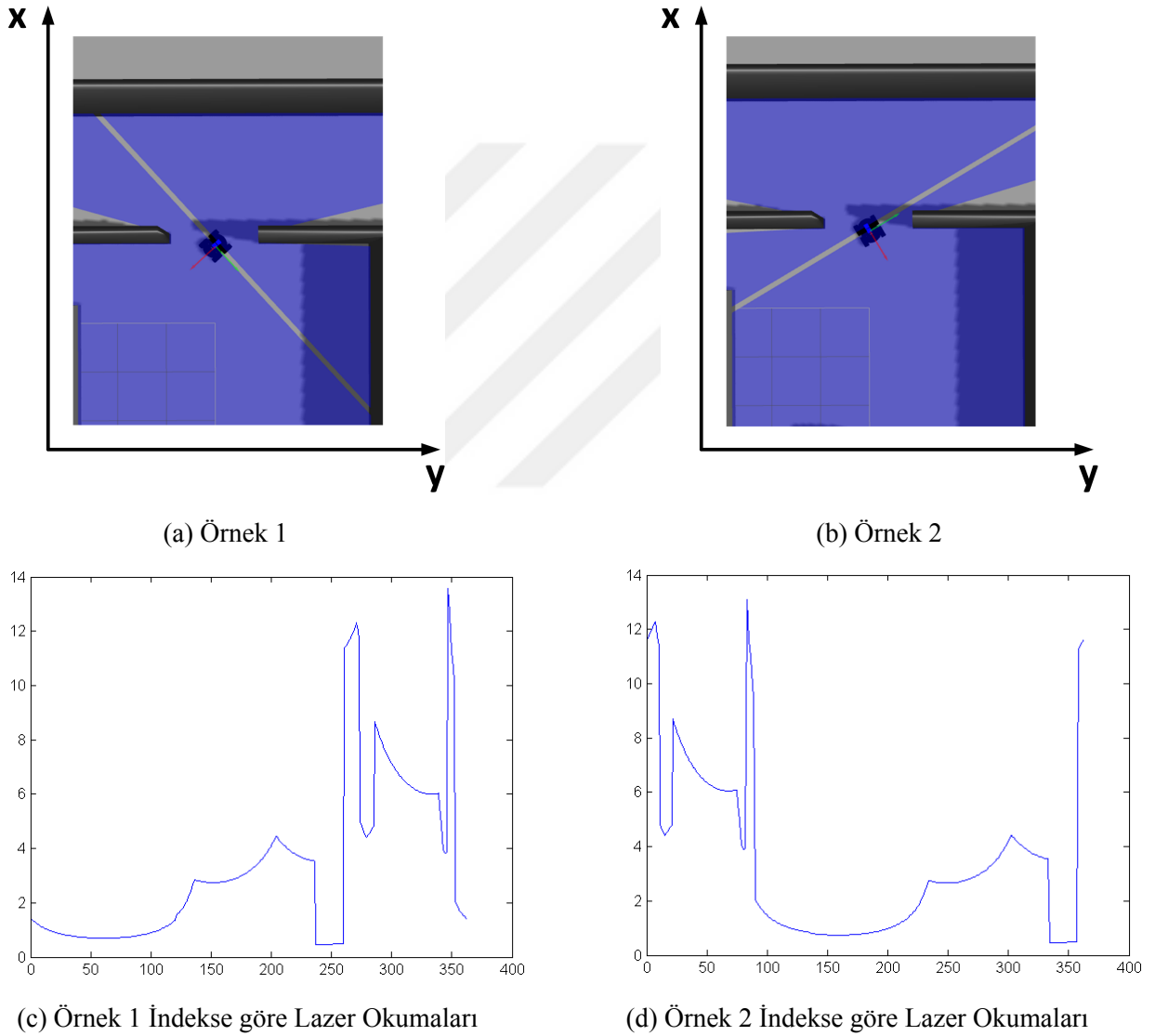
Bu modelde, x_i robotun i 'inci anda bulunabileceği üç anlamsal sınıftan birini göstermektedir. Denklem'deki c_k ifadesi ise robottan elde edilen lazer mesafe verilerinin k 'inci küme merkezine en yakın olduğunu anlatmaktadır. Küme merkezleri ve her lazer verisinin ait olduğu küme merkezi bilgileri K-Ortalama ve ÖVN sınıflandırma yöntemlerinden elde edilmektedir. $P(c_k^i|x_i)$ olasılığı eğitim verisinden elde edilen küme merkezleri kullanılarak hesaplanmaktadır. $P(x_i|x_{i-1})$ olasılığı ise geçiş olasılığı olarak adlandırılmakta ve bir anlamsal sınıftan diğerine geçiş olasılığını göstermektedir. Bu yaklaşımda, ardsalların ilk değerleri normal dağılım kullanılarak atanmıştır. Son olarak, α normalizasyon katsayısı olarak adlandırılmakta ve üç anlamsal sınıfın ardsallarının toplamının 1 olmasını sağlamaktadır. Her adımda ardsalların $Bel(x_i)$ değerleri güncellenmekte ve en büyük ardsal değerine sahip sınıf o anda robotun anlamsal sınıfı olarak kabul edilmektedir. Her adımda, bu şekilde elde edilen anlamsal sınıflar, daha sonra eldeki gerçek referanslar ile (ground truth) karşılaştırılarak her sınıfın bulunma başarısı hesaplanmaktadır.

6.2.5 Kural tabanlı kapı bulma yaklaşımı

Bina içi ortamlarda, oda ve koridor gibi binanın çeşitli alanları kapılar ile birbirinden ayrılmaktadır. Açık kapıların bulunması, konumlandırma, seyrüsefer ve keşif gibi temel seviyedeki görevlere katkıda bulunmaktadır. Ayrıca arama-kurtarma ve yaşlı bakımı gibi daha karmaşık görevler için de robota ortam hakkında anlamsal farkındalık sağlamaktadır.

Geçmiş çalışmalarda genellikle görsel yöntemler üzerine yoğunlaşmıştır. Bu çalışmalarda kapıların kapalı olduğu varsayımı yapılmakta ve robotlar koridor boyunca ilerlerken kamerasının görüş alanında bulunan kapıları saptamak amaçlanmaktadır. Bununla birlikte, bir çok uygulamada robotların oda ve koridorlar arasında seyrüsefer yaparken açık kapılardan geçtiklerinin farkında olmaları gerekmektedir. Bu çalışmada, sadece lazer mesafe verisi kullanan kural tabanlı bir kapı bulma yöntemi önerilmektedir.

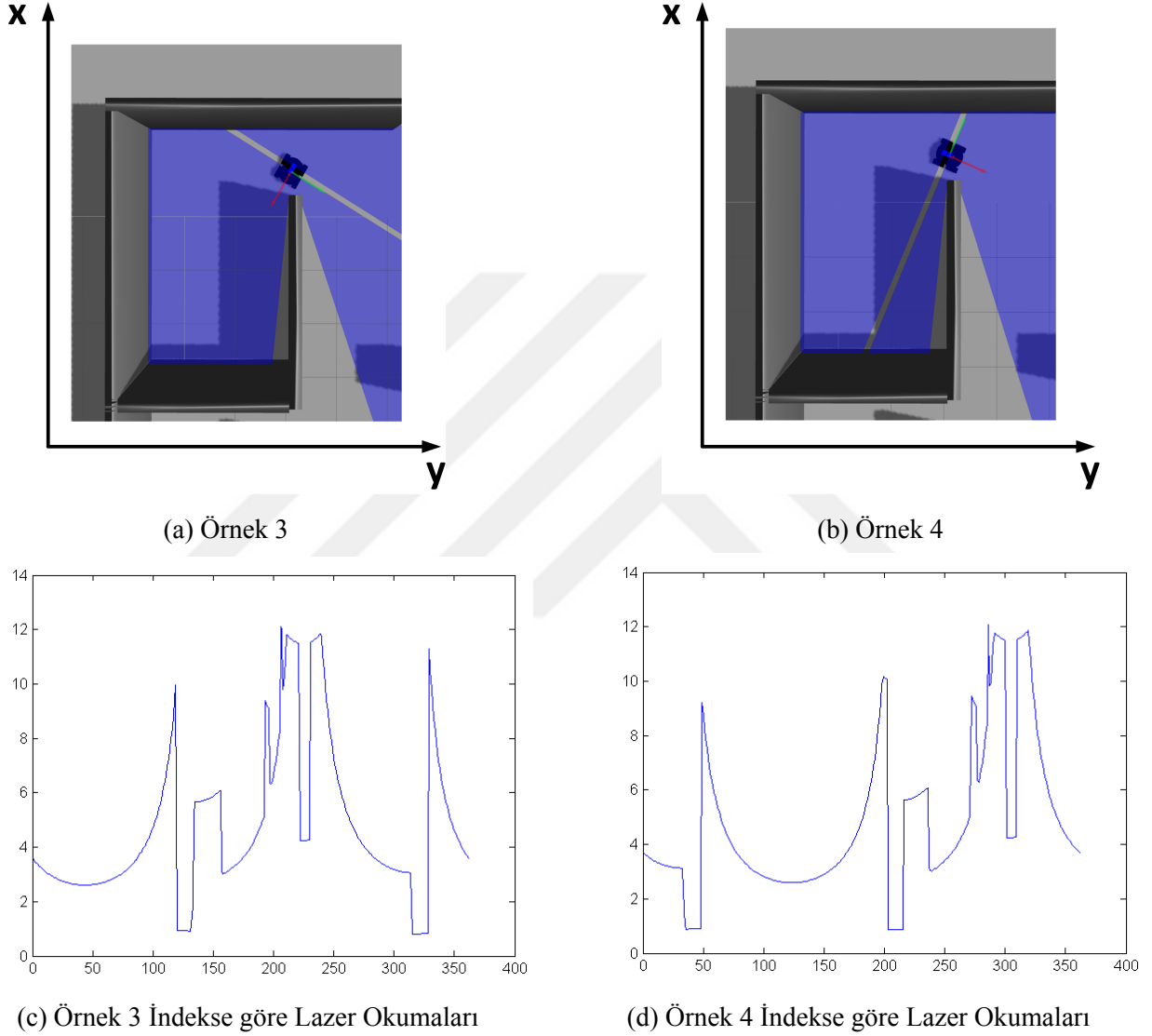
Önerilen yöntem, üç varsayım üzerine kurulmuştur. İlk olarak, kapıların açık olması ve robotun kapılardan geçebiliyor olması gerekmektedir. İkincisi, kapıların yatay veya dikey olması, yani, referans koordinat sisteminin eksenlerine paralel durumda olması gerekmektedir. Bu durum genellikle çoğu binada gözlemlenebilir. Şekil 6.8a ve 6.8b'de yatay olarak tanımladığımız kapılar yer almaktadır. Şekilde robotlar hemen hemen aynı konumda ancak farklı yönelimlere sahiptirler. Şekil 6.8c ve 6.8d'de ise indekse bağlı olarak lazer ölçümlerini göstermektedir.



Şekil 6.8: Yatay kapıda farklı yönelimlerle lazer ölçümleri

Benzer şekilde, Şekil 6.9a ve 6.9b'de dikey olarak tanımladığımız kapılar, Şekil 6.9c ve 6.9d'de ise indekse bağlı olarak lazer ölçümlerini göstermektedir. Şekillerde görüldüğü gibi kapıların yatay ya da dikey olması ve robotun yöneliminin farklı olması indeks bazında

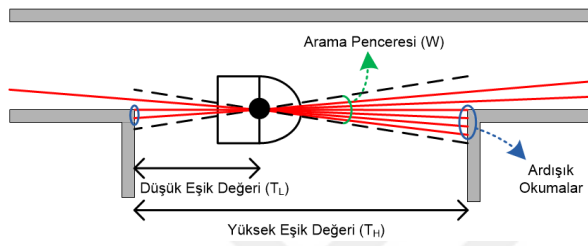
lazer ölçümlerini farklılaştırmakta ve kapı bulma problemini daha da zorlaştırmaktadır. Bu noktada, lazer verilerine yönelim normalizasyonu yapılmaktadır. Yönelim normalizasyonu sonucunda robotun yönelimi sanal olarak sıfır dereceye sabitlenmektedir. Son varsayım ise robotun hemen hemen kapı çerçevesi içinde bulunması gerekliliğidir.



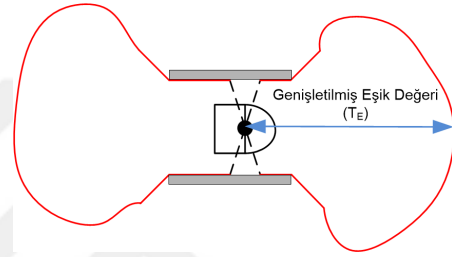
Şekil 6.9: Dikey kapıda farklı yönelimlerle lazer ölçümleri

Önerilen yöntem, robot kapı bölgesinde bulunduğu lazer taramasının darboğaz şeklinde olacağı gözlemine dayanmaktadır. Kapı bölgeleri kapı çerçevesinin yakını ve/veya arasında kalan alanları tanımlamaktadır. Bu bölgeleri tespit etmek için darboğaz karakteristiğini anlatan bir dizi kural geliştirilmiştir. Yönelim normalizasyonu sayesinde yatay ve dikey kapıları bulabilmek için bir takım kurallar daha önceden belirlenmiş açılarını kullanarak belirlenebilmektedir. Bunu yapabilmek için, darboğazı betimleyen bazı

tanımlamalara ihtiyaç duyulmaktadır. Kapı bulmak için tanımlanan parametreler Şekil 6.10'a da gösterilmektedir. Düşük (T_L) ve yüksek (T_H) eşik değerleri robot ve kapı çerçeveleri arasında bulunan minimum ve maksimum uzaklıkları tanımlamaktadır. Arama penceresi (W) darboğazın bulunması için kontrol edilen lazer ışınlarının aralığını göstermektedir. W , darboğaza karşılık gelen ardışık lazer ışınlarının sayısı da ardışık okumalar parametresini ifade etmektedir. Genişletilmiş eşik değeri parametresi (T_E) Şekil 6.10b'de gösterilmiştir. Bir konumun darboğaz karakteristiği gösterdiğine karar verilmesi için, ardışık okumalar parametresinin belli bir eşik değerin üstünde olması ve W dışında kalan lazer okumalarının maksimumunun da T_E parametresini geçmesi gerekmektedir.



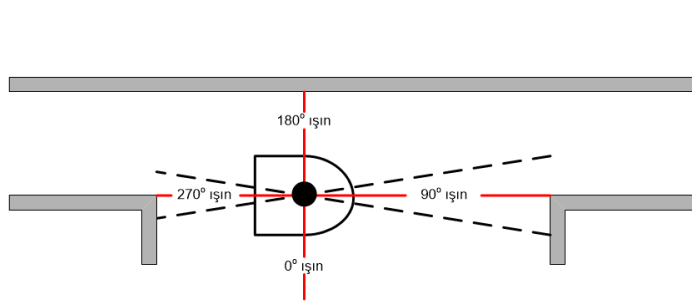
(a) Paralel nesnelere bulmak için gereken parametreler



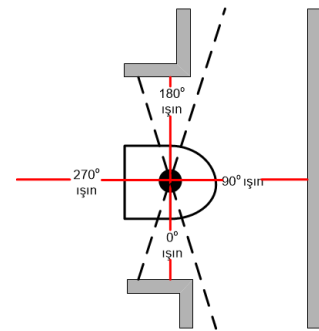
(b) Paralel nesnelere kapıdan ayırmak için gereken parametresi

Şekil 6.10: Kapı bulma için tanımlanan parametreler

Kapıların dikey veya yatay konumda oldukları, yani referans koordinat sisteminin eksenlerine paralel oldukları kabul edilmektedir (Şekil 6.11).



(a) Yatay Kapı



(b) Dikey Kapı

Şekil 6.11: Yatay ve dikey kapı durumları

Eğer robot bir yatay kapı konumundaysa, 90 derece komşuluğundaki lazer ışınları kapının çerçevesinin bir parçasını vurabilir. Bu lazer ışınlarını ayırt edebilmek için, W içerisinde kalan lazer ışınlarından, mesafeleri T_L ve T_H değerleri arasında olanlar

aranmaktadır. Bu lazer ışınları aday ışınlar olarak adlandırılmaktadır. Eğer ardışık aday ışınların sayısı ardışık okumalar parametresinden fazla ise, algoritma kapı çerçevesinin karşı tarafına bakmaktadır. Eğer karşı taraftaki ardışık aday ışınların sayısı da ardışık okumalar parametresinden fazla ise, robot, birbirine mesafesi yaklaşık olarak T_H kadar olan iki paralel nesne arasında bulunmaktadır (duvar, engel veya kapı çerçevesi). Ancak, bu bilgi kendi başına bir konumun kapı çerçeveleri arasında mı yoksa sadece paralel iki nesne arasında mı olduğunu ayırt etmek için yeterli değildir. Gerçek bir darboğaz karakteristiğinin oluşması için, iki paralel tarafın dışında kalan lazer ışını mesafelerinin en büyüğünün de T_E parametresinden büyük olması gerekmektedir. Bahsedilen bu özellikler darboğaz karakteristiğini, dolayısıyla da bir kapı konumunun tanınmasında kullanılan kuralları oluşturmaktadır. Dikey kapılar için ise, kapı çerçeveleri için 180 derecedeki lazer ışınının yakın çevresini aramak gerekmektedir. Daha sonra da yatay kapılar için uygulanan aynı adımlar takip edilerek, eğer kriterler sağlanırsa, robotun bir kapı konumunda bulunduğuna karar verilmektedir (Kaleci vd. 2015c). Algoritma 12'de kapı bulma yöntemi detaylı olarak verilmiştir.

6.2.6 Anlamsal sınıflandırma tabanlı keşif yaklaşımı

Keşifte anlamsal sınıflandırma bilgilerinin kullanılabilmesi için sistem mimarisinde bulunan erkinlerde bazı düzenlemeler yapılmıştır. Bu keşif yaklaşımında anlamsal bilgilerin düğümlere yüklenmesi istenmiştir. Bu şekilde, metrik haritadaki her hücre için bu hesaplamanın önüne geçilmek istenmiştir. Dolayısıyla, anlamsal sınıflandırma tabanlı keşif sırasında topolojik harita kullanılacaktır. Bununla birlikte, anlamsal bilgilerin sadece hedef seçiminde kullanılması planlanmıştır. Izgara erkini haritaları oluşturmak, güncellemek, arayüzde göstermek ve istenildiğinde diğer erkinlerle paylaşmak ile sorumludur. Dolayısıyla bu keşif algoritmasında bu erkinde bir değişikliğe ihtiyaç duyulmamaktadır. Benzer şekilde, üretici erkin belirli aralıklarla yapılması gereken görevleri belirlemekte, bu görevleri fiyatlandırmakta ve robot için en faydalı görevi seçmektedir. Görevlerin belirlenmesinde sınır hücre yaklaşımı kullanılmaya devam edilmiştir. Topolojik harita kullanılırken belirlenen, görevi tanımlayan sınır hücre alanının çevresi fiyat olarak kabul edilmektedir. Macar algoritması görevin seçilmesinde eski rolüne devam etmektedir.

Tüketici erkin, belirlenen görevler için maliyeti hesaplamaktadır. Anlamsal sınıflandırma ile elde edilen bilgilerin robotun hedef seçiminde kullanılabilmesi için görev maliyetleri üzerinde değişiklikler yapılmıştır. Topolojik harita ile keşif yapıldığında robot görevine gitmek için bulunduğu konum ile görevi arasındaki düğümleri kullanmakta ve görevin maliyeti düğümler arası mesafe olarak hesaplanmaktaydı. Anlamsal sınıflandırma tabanlı keşifte, topolojik haritada yer alan düğümler oda, koridor, kapı ve bilinmeyen olarak anlamsal bilgilere sahip olmaktadır. Her yeni üretilmiş düğüm bilinmeyen anlamsal sınıfı

Algoritma 12: Kapı Bulma Algoritması

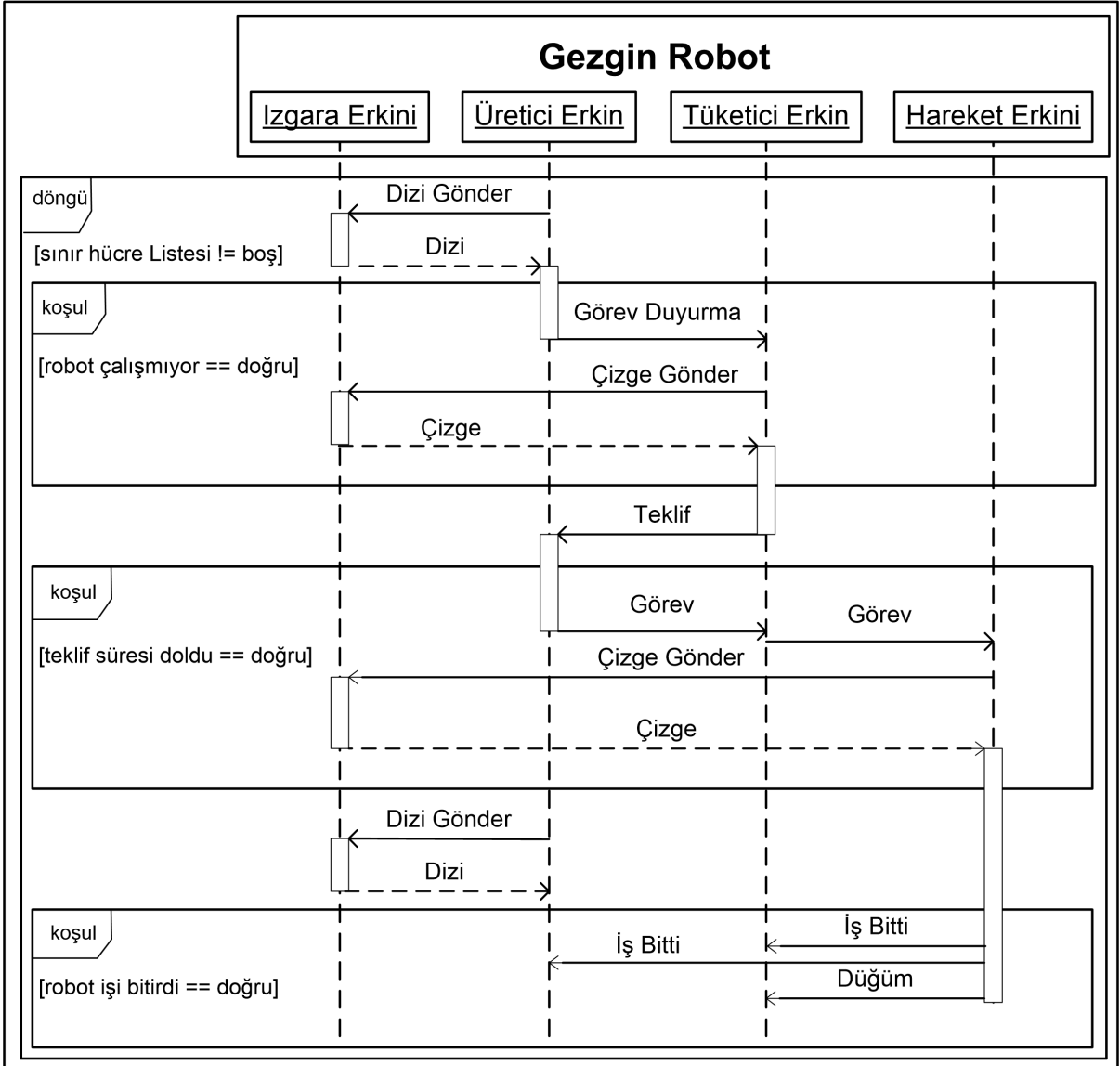
Girdi: Normalize edilmiş lazer mesafe verisi $shiftedLaserData[362]$
Çıktı: Lazer mesafe verisi kapı sınıfına mı ait yoksa değil mi
Veri: W, T_L, T_H, T_E , Ardışık okuma parametresi SI
 $k \leftarrow 0; l \leftarrow 0;$
forall the $i : 90 - W/2$ **to** $90 + W/2$ **do**
 if $shiftedLaserData[i] > T_L$ **and** $shiftedLaserData[i] < T_H$ **then**
 $k \leftarrow k + 1;$
 end
end
if $k > SI$ **then**
 forall the $i : 270 - W/2$ **to** $270 + W/2$ **do**
 if $shiftedLaserData[i] > T_L$ **and** $shiftedLaserData[i] < T_H$ **then**
 $l \leftarrow l + 1;$
 end
 end
end
if $l > SI$ **then**
 $sum \leftarrow 0; index \leftarrow 0;$
 forall the $i \notin (90 - W/2, 90 + W/2)$ **and** $(270 - W/2, 270 + W/2)$ **do**
 $sum \leftarrow sum + shiftedLaserData[i];$
 $index \leftarrow index + 1;$
 end
 if $sum/index > T_E$ **then**
 return lazer mesafe verisi kapıya ait
 end
end
 $k \leftarrow 0; l \leftarrow 0;$
forall the $i : 361 - W/2$ **to** $0 + W/2$ **do**
 if $shiftedLaserData[i] > T_L$ **and** $shiftedLaserData[i] < T_H$ **then**
 $k \leftarrow k + 1;$
 end
end
if $k > SI$ **then**
 forall the $i : 180 - W/2$ **to** $181 + W/2$ **do**
 if $shiftedLaserData[i] > T_L$ **and** $shiftedLaserData[i] < T_H$ **then**
 $l \leftarrow l + 1;$
 end
 end
end
if $l > SI$ **then**
 $sum \leftarrow 0; index \leftarrow 0;$
 forall the $i \notin (361 - W, 0 + W)$ **and** $(180 - W, 181 + w)$ **do**
 $sum \leftarrow sum + shiftedLaserData[i];$
 $index \leftarrow index + 1;$
 end
 if $sum/index > T_E$ **then**
 return lazer mesafe verisi kapıya ait
 end
end

ile işaretlendirilmektedir. Topolojik harita ile keşif yapıldığı gibi görevin maliyeti hesaplanmaktadır. Bu noktada, robotun hedefinin dolayısıyla ziyaret edeceği son düğümün anlamsal sınıfının robotun seçeceği görev üzerinde önemli etkisi bulunmaktadır. Bu noktada, robotlara oda öncelikli robot ya da koridor öncelikli robot olarak iki rol atanmaktadır. Oda öncelikli bir robotun aday listesinde bulunan oda anlamsal sınıfına ait hedefler tamamlanmadan koridor anlamsal sınıfına ait bir hedefe gitmesi istenmemektedir. Benzer durum koridor öncelikli robot için de geçerlidir. Bunu gerçekleştirebilmek için yol planındaki son düğümün anlamsal sınıfı kontrol edilmektedir. Bu düğüm oda öncelikli robot için koridor anlamsal sınıfına ait ise hesaplanan maliyet çok büyük bir sayı ile çarpılmaktadır. Benzer durum koridor öncelikli robot içinde gerçekleştirilmektedir. Bu aşamada, sınır hücre keşif yönteminde hedefler bilinen ve bilinmeyen sınırların bölgeleri olduğu için çoğu zaman yeni üretilmiş düğümler bu hedeflere en yakın olmaktadır. Bu noktada, bilinmeyen durumunda olan anlamsal sınıfı diğer üç anlamsal sınıftan birine dönüştürmek gerekmektedir. Bunu yapabilmek için Markov modeli kullanılmıştır. Bir anlamsal sınıftan diğerine geçiş olasılıkları robotun ürettiği topolojik haritalar kullanılarak elde edilmiştir. Bilinmeyen bir düğüm bağlı olduğu diğer düğümlerin anlamsal sınıflarını kullanarak kendi anlamsal sınıfına karar vermektedir.

Hareket erkini iki temel görevden sorumludur. Bunların ilki, robotun bulunduğu noktadan belirlenen hedefine gidebilmesi için yol planının hazırlanmasıdır. Yol planı oluştuktan sonra robotun engellerden sakınarak güvenli bir şekilde hedefine varabilmesi için VAH kullanılmıştır. Robot, topolojik haritada bir düğüme vardığında robotun yönelimi ve 362 derecelik lazer taramasını kullanarak bu düğümün anlamsal sınıfına karar vermektedir. Robot hedefine vardığında tekrar anlamsal sınıfına karar vermekte ve sonra bu anlamsal sınıfı ziyaret ettiği en son düğüme atamaktadır. Robot görevi tamamladığında, düğümler hakkındaki anlamsal bilgileri tüketici erkine göndermektedir. Bu sayede tüketici erkin yeni görev atamaları için bu bilgileri kullanabilmektedir. Şekil 6.12'de erkinler arası iletişimi gösteren sıralama diyagramı keşifte anlamsal sınıf bilgilerinin kullanıldığı sistem için verilmiştir.

Çok robotlu anlamsal tabanlı keşif yaklaşımı temel olarak tek robotlu anlamsal tabanlı keşif ve çok robotlu topolojik keşif yaklaşımlarında kullanılmıştır. Üretici erkinlerde çok robotlu topolojik keşif yaklaşımından farklı herhangi bir değişiklik yapılmamıştır.

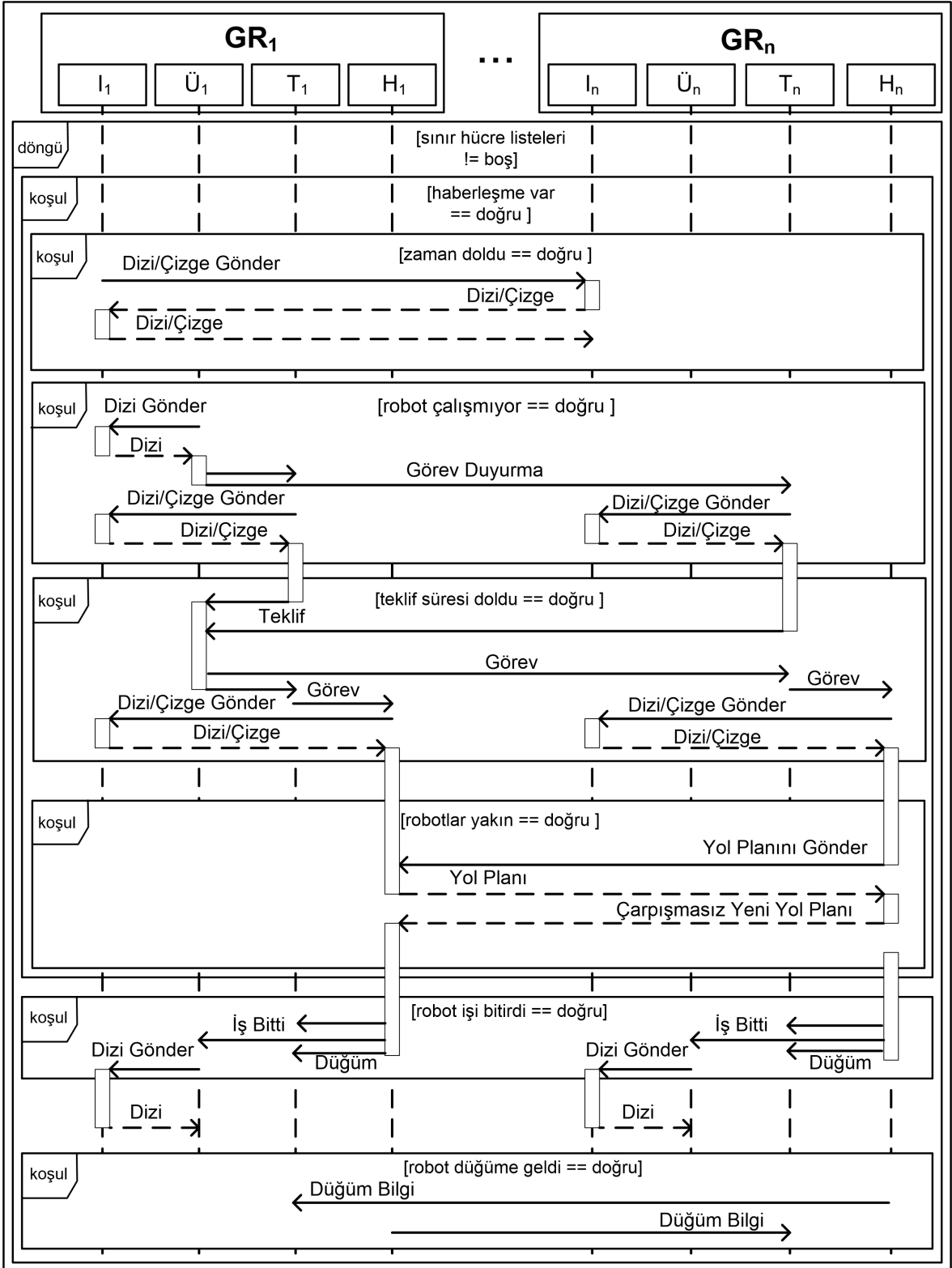
Izgara düğümü topolojik haritaların birleştirilmesinden sorumludur. Bu aşamada, robotların topolojik haritaları birleştirilirken düğümlerin anlamsal bilgileri de robotlar tarafından birbirlerine gönderilmektedir. Bir robotun sahip olduğu bir düğüm bilinmeyen durumunda iken diğer robottan gelen ilgili düğüme ait anlamsal bilgi direk bu düğüme atanmaktadır. Bununla birlikte, eğer bir robot bilinmeyen dışındaki bir anlamsal sınıfa ait



Şekil 6.12: Anlamsal Tabanlı Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)

düğümüne sahipse ve diğer robottan bu düğümüne ait anlamsal bilgi bilinmeyen olarak geldiyse kendi bilgisini korumaktadır. Eğer her iki robotunda bilinmeyen dışında anlamsal sınıfa ait olan karşılıklı düğümleri varsa bu düğümlerin anlamsal sınıflarına karar vermek için Markov modeli kullanılmaktadır.

Hareket erkini her düğümüne vardığında diğer robotun tüketici erkinine bu düğümün anlamsal sınıfını göndermektedir. Diğer robotun haberleşme menzili içindeyse bu bilgiyi almakta ve kendi topolojik haritasında bu düğümüne karşılık gelen düğümün anlamsal bilgisini yukarıda anlatılan şekilde güncellemektedir. Eğer robotlar haberleşme menzili içinde değilse bu bilgi kullanılamamaktadır. Şekil 6.13'te erkinler arası iletişimi gösteren sıralama diyagramı çok robotlu keşif için verilmiştir.



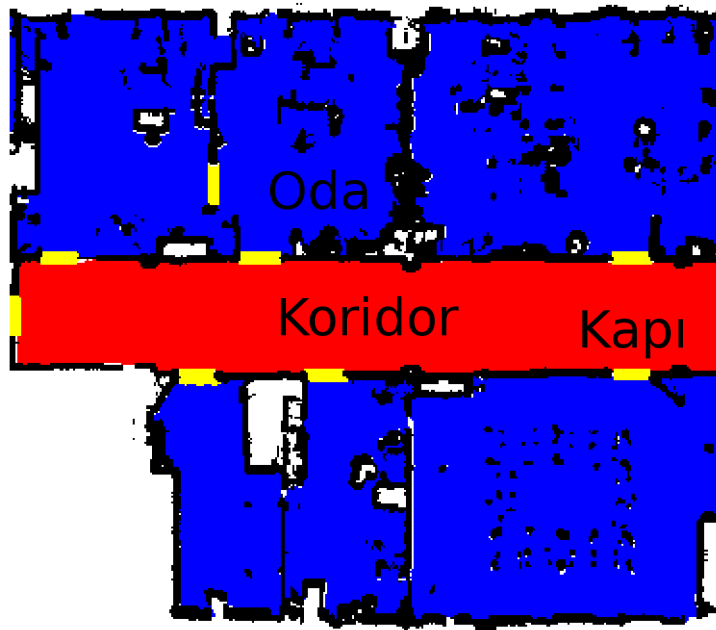
Şekil 6.13: Çok Robotlu Anlamsal Tabanlı Keşifte Erkinler Arası İletişim (Sıralama Diyagramı)

6.3 Sonular

Bu alt b6l6mde ilk olarak anlamsal sınıflandırma yaklaşımının başarısı hem Freiburg 79 hem de ESOGÜ binalarından elde edilen lazer mesafe verileri kullanılarak deęerlendirilecektir. Daha sonra, kapı bulma algoritmasının başarısını etkileyen parametrelerin etkisi analiz edilecektir. Kapı bulma ve anlamsal sınıflandırma yaklaşımlarının birleştirilmesi ile elde edilen yöntemin ESOGÜ binasındaki veriler için ürettięi sonuçlar verilecektir. Son olarak, tek ve çok robot ile keşifte anlamsal sınıflandırma bilgilerin kullanılmasının olumlu ve olumsuz yönler incelenecektir.

6.3.1 Freiburg 79 test verisi sonuçları

Önerilen yöntem, Freiburg Üniversitesi'nin 79 no'lu binasından elde edilen lazer mesafe verileri kullanılarak test edilmiştir (Mozos vd. 2005). Ortam tam ortasından ikiye ayrılmış, sağ tarafı eğitim verisi olarak, sol tarafı ise test için kullanılmıştır. Eğitim verisinde 43624 oda, 16687 koridor ve 1449 kapı örneęi bulunmaktadır. Eğitim verisi KO ve ÖVN yöntemleri ile sınıflandırılmış, elde edilen bilgiler test verisinin anlamsal sınıflarını belirlemek için kullanılmıştır. Test verisinde 50180 oda, 15516 koridor ve 701 kapı örneęi bulunmaktadır. $P(c_k^i|x_i)$ ve $P(c_k)$ olasılıkları K-Ortalama sınıflandırma yönteminden elde edilen küme merkezi verileri kullanılarak hesaplanmıştır. Bununla birlikte, ÖVN başlangıta belirlenen küme merkezlerini kullanarak anlamsal sınıflara karar vermektedir. Yöntemlerin verimlilięini ölçmek için kullanılan gerek referans (ground truth) Şekil 6.14'te gösterilmiştir.



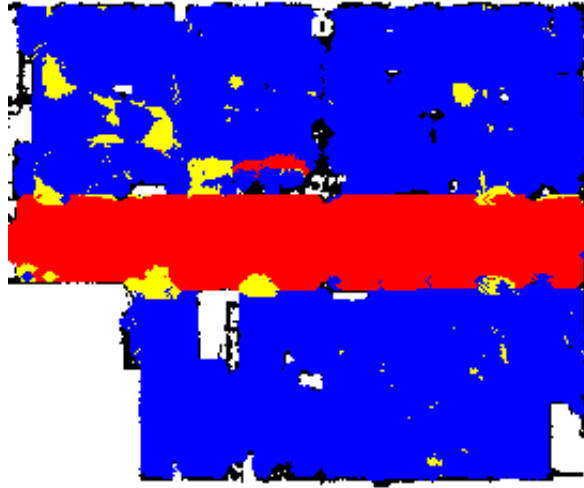
Şekil 6.14: Freiburg 79 gerek referans verisi

Çizelge 6.1: Test Verisi Sonuçları

	Yöntem	Oda	Koridor	Kapı	Ortalama	Sınıf Ortalaması
K =40	KO,EO	78.77	78.65	50.93	78.45	69.45
	KO,V	100	98.70	0	98.64	66.23
	ÖVN	87.83	98.11	55.92	89.89	80.62
K =100	KO,EO	77.33	97.51	53.64	81.80	76.16
	KO,V	98.79	98.33	7.99	97.72	68.37
	ÖVN	94.36	95.81	54.49	94.28	81.56
K =400	KO,EO	89.90	96.23	41.63	90.86	75.92
	KO,V	98.81	98.75	20.63	97.98	72.73
	ÖVN	95.02	98.28	54.07	95.35	82.46
K =1000	KO,EO	92.25	95.81	36.95	92.50	75.00
	KO,V	97.96	98.60	27.25	97.36	74.60
	ÖVN	96.38	98.09	59.91	96.39	84.79

Sonuçlar Çizelge 6.1'de verilmiştir. Çizelgede, K küme sayısını, EO (Eşit olasılık) ve V (veriden elde edilen olasılık) önsel olasılık değerlerini ifade etmektedir. Eşit olasılık kullanıldığında tüm sınıfların önsel olasılıkları 1/3 olarak kabul edilmektedir. Diğer yandan veriden elde edilen olasılıklar için her sınıfın verinin içindeki örnek sayısı verinin tamamındaki örnek sayısına bölünmektedir. 2. 3. ve 4. sütunlar sırasıyla oda, koridor ve kapı örnekleri için sınıflandırma başarısını, 5. ve 6. sütunlar ise sırasıyla ortalama ve sınıf bazlı ortalamayı göstermektedir. Çizelgede görüldüğü üzere KO yönteminde EO önsel olasılık olarak seçildiğinde kapı bulma oranları yükselmektedir. Bununla beraber, V önsel olasılıkları kullanıldığında kapı sınıfının önsel olasılığı çok düşük olduğu için kapı sınıfının bulunma başarısı çok azalmakta ve oda ve koridor bulma başarısı artmaktadır. ÖVN yöntemi ise üç anlamsal sınıfın başarısını birlikte artırmayı amaçladığı için oda ve koridor anlamsal sınıflarının başarısını çok fazla etkilemeden kapı bulma oranını yüzde 60'a kadar artırmıştır.

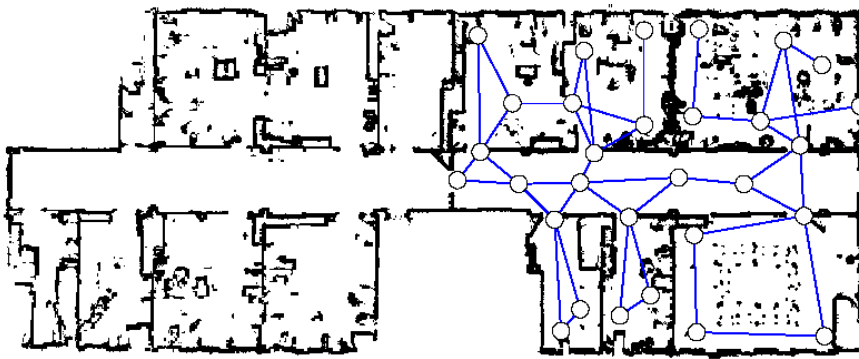
Şekil 6.15'te ÖVN yönteminin 1000 küme merkezi kullandığında elde edilen görsel sonuç verilmiştir. Şekilde görüldüğü gibi ÖVN yöntemi oda ve koridor anlamsal sınıflarını rahatlıkla birbirinden ayırabilmektedir. Bununla birlikte, kapı anlamsal sınıfına ait olması gereken bazı noktalar oda veya koridor anlamsal sınıflarına ait gözükmektedirler. Ayrıca, odaların içindeki bazı noktalarında kapı olarak belirlendiği gözükmektedir. Sonuç olarak, ÖVN genellikle oda ve kapı anlamsal sınıflarını birbiri ile karıştırmaktadır.



Şekil 6.15: ÖVN Görsel Sonuç

6.3.2 Freiburg 79 yörünge verisi sonuçları

Markov modeli gerçekleyebilmek için verinin hem eğitim kısmında hem de test kısmında yörüngeler üretilmiştir. İlk olarak, seçilen bazı noktalar düğüm olarak tanımlanmıştır. Daha sonra, bu düğümler arasındaki her olası bağlantı belirlenmiştir. Bu çalışmada, yörüngelerin başlangıç ve bitiş düğümleri rastsal olarak belirlenmiştir. Dijkstra en kısa yol bulma algoritması bu düğümler arasında yörünge üretmek amacıyla kullanılmıştır. Ortamın test kısmı için belirlenen düğümler ve onlar arasındaki olası bağlantılar Şekil 6.16'da verilmiştir.



Şekil 6.16: Düğüm Ağı

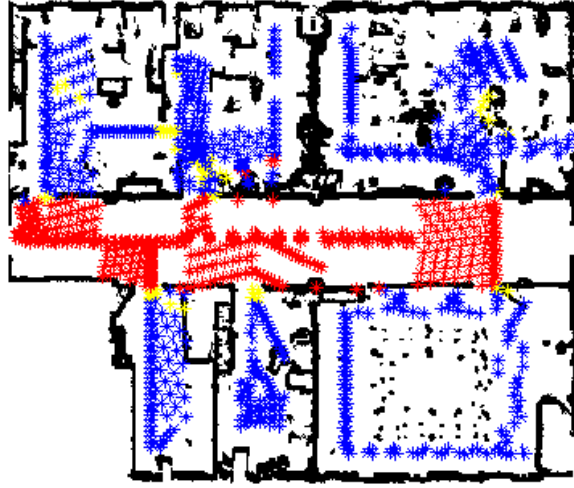
Ortamın test kısmında 1000 yörünge ve 98405 anlamsal nokta önerilen yaklaşımların test edilmesi için kullanılmıştır. Bu test verisinde 55128 koridor, 40242 oda ve 3035 kapı örneği bulunmaktadır. KO, ÖVN ve Markov model kullanılarak yörünge verisi ile elde edilen sınıflandırma sonuçları Çizelge 6.2'de verilmiştir. Markov model kapı

sınıfının başarısını artırmak amacıyla gerçekleştirilmiştir. Ancak, Markov model oda ve koridor sınıflarının başarısını artırmıştır. Bunun sebebi odadan kapıya ve koridordan kapıya geçişlerin çok seyrek olmasıdır. Bir oda ya da koridorda kalma olasılığı her zaman bir kapıya gitme ya da kapıda kalma olasılığından daha büyüktür. Yörünge ile elde edilmiş veri için ÖVN hem kapı sınıfının başarısında hem de ortalama başarıda KO yönteminden daha iyi sonuçlar üretmiştir. Bunun sebebi, ÖVN denetimli bir öğrenme yöntemi olduğu için sınıfların geometrik karakterlerini daha iyi yansıtmaktadır.

Çizelge 6.2: Yörünge Verisi Sonuçlar

	Yöntem	Oda	Koridor	Kapı	Ortalama	Sınıf Ortalaması
K =40	KO,EO	65.84	89.77	36.28	78.34	63.96
	KO,V	100	99.95	0	96.89	66.65
	KO Markov	99.49	99.09	12.13	96.58	70.24
	ÖVN	81.53	99.16	80.85	91.39	87.19
	ÖVN Markov	98.50	99.71	27.45	96.99	75.22
K =100	KO,EO	63.54	98.82	65.83	83.38	76.07
	KO,V	99.22	99.52	0	96.32	66.24
	KO Markov	97.80	99.24	13.17	96.02	70.27
	ÖVN	92.57	98.93	76.74	95.64	89.41
	ÖVN Markov	93.01	99.85	47.97	95.46	80.28
K =400	KO,EO	86.38	97.57	48.70	91.48	77.55
	KO,V	98.54	99.95	15.58	96.77	71.36
	KO Markov	96.01	99.76	30.87	96.10	75.55
	ÖVN	94.97	99.35	69.42	96.64	87.91
	ÖVN Markov	96.26	99.95	51.66	96.95	82.62
K =1000	KO,EO	91.63	97.20	51.43	93.51	80.09
	KO,V	97.16	99.81	33.97	96.70	76.98
	KO Markov	96.97	99.65	45.67	96.91	80.77
	ÖVN	93.47	99.95	69.42	96.36	87.62
	ÖVN Markov	94.61	99.95	57.59	96.46	84.05

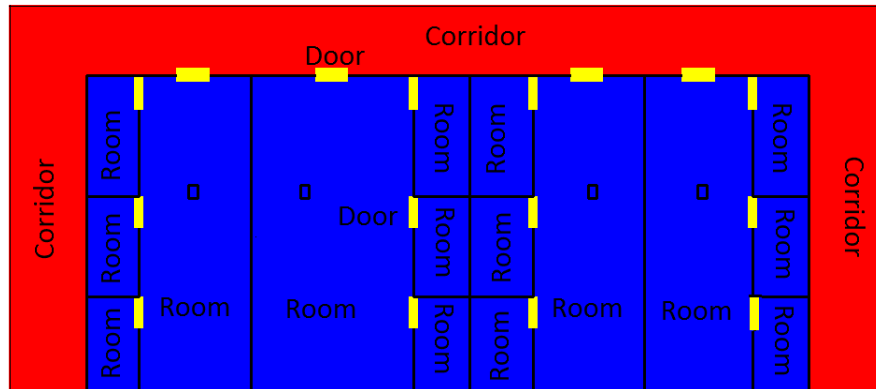
Şekil 6.17'de ÖVN Markov yönteminin 1000 küme merkezi kullandığında elde edilen görsel sonuç verilmiştir. Şekilde görüldüğü gibi ÖVN yöntemi oda ve koridor anlamsal sınıflarını rahatlıkla birbirinden ayırabilmektedir. Bununla birlikte, test verisi sonuçlarında olduğu gibi yörünge verisi sonuçlarında da genellikle oda ve kapı anlamsal sınıflarını birbiri ile karıştırmaktadır.



Şekil 6.17: ÖVN Görsel Sonuç

6.3.3 ESOGÜ yörünge verisi sonuçları

Önerilen yöntemin farklı bir ortamda test edilmesi amacıyla ESOGÜ Laboratuvar binası Gazebo benzetim ortamında modellenmiştir. Benzetimlerde iki adet lazer mesafe algılayıcısı bulunan bir Pioneer P3-AT robotu kullanılmıştır. Robotun kontrolü için Robot İşletim Sistemi (ROS) kullanılmıştır. Önerilen yöntemler C++ programlama dili kullanılarak kodlanmıştır. İlk olarak, seçilen bazı noktalar düğüm olarak tanımlanmış ve bu düğümler arasındaki ilişkiler kullanılarak ortam bir çizge ile ifade edilmiştir. Daha sonra bu çizge üzerinde rassal başlangıç ve bitiş düğümleri belirlenmiştir. Bu düğümler arasında Dijkstra en kısa yol bulma algoritması kullanılarak yol planı yapılmış ve robota bu yörünge izletilmiştir. Robot seyrüseferi sırasında veri toplamıştır. Veri kümesi, oluşturulan yörüngelerden elde edilmiş 11815 oda, 14561 koridor ve 1541 kapı örneği içermektedir. Yöntemlerin verimliliğini ölçmek için kullanılan gerçek referans Şekil 6.18'de gösterilmiştir.



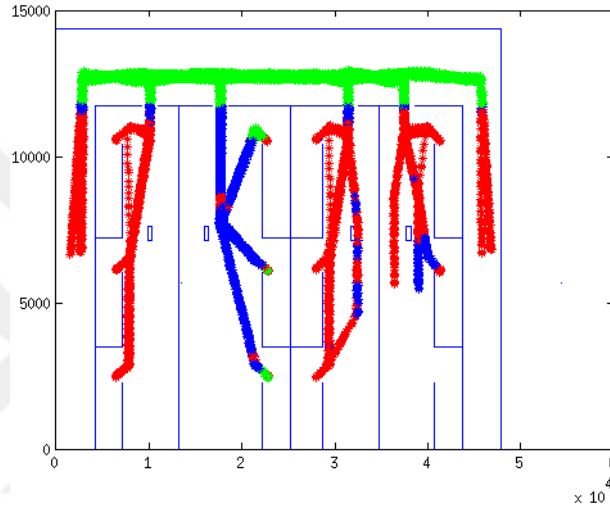
Şekil 6.18: ESOGÜ gerçek referans verisi

Çizelge 6.3: ESOGÜ Yörünge Verisi Sonuçları

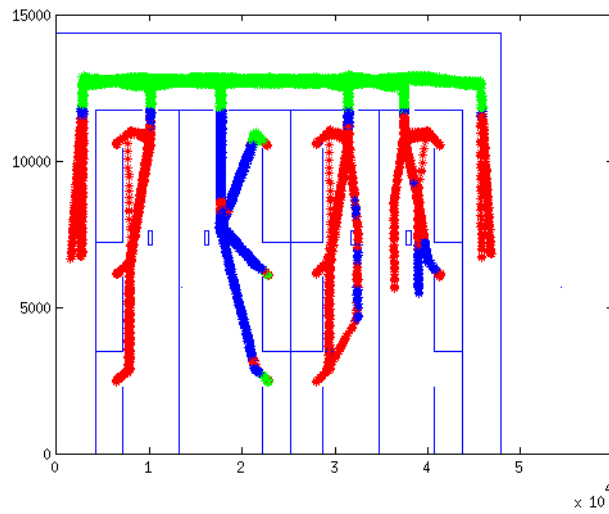
	Yöntem	Oda	Koridor	Kapı	Ortalama	Sınıf Ortalaması
K =40	KO,EO	43.04	89.54	38.08	67.43	56.89
	KO,V	100	92.46	0	94.58	64.15
	KO,Mar.	52.98	92.48	21.87	73.31	55.78
	ÖVN	75.54	92.40	38.08	83.86	68.67
	ÖVN,Mar.	75.52	92.37	42.01	83.90	96.97
K =100	KO,EO	56.76	91.41	8.60	74.28	52.26
	KO,V	100	91.41	6.88	94.13	66.10
	KO,Mar.	65.45	91.39	7.37	78.25	54.74
	ÖVN	86.72	92.38	46.93	89.12	75.34
	ÖVN,Mar.	91.27	92.45	35.14	91.07	72.95
K =400	KO,EO	89.92	92.23	20.39	90.12	67.51
	KO,V	95.82	92.25	9.34	92.68	65.80
	KO,Mar.	91.45	92.26	10.32	90.69	64.68
	ÖVN	99.00	92.35	37.59	94.61	76.32
	ÖVN,Mar.	99.03	92.43	24.57	94.47	72.01
K =1000	KO,EO	89.46	92.37	28.01	90.10	69.95
	KO,V	96.42	92.51	10.81	93.12	66.58
	KO,Mar.	90.74	92.39	14	90.49	65.71
	ÖVN	97.00	92.46	29.98	93.63	73.15
	ÖVN,Mar.	99.84	92.44	27.03	94.88	73.10

Elde edilen sonuçlar Çizelge 6.3'te verilmiştir. Çizelgedeki sonuçlar incelendiğinde hem KO hem de ÖVN yöntemleri için bütün K değerleri için oda ve koridor sınıflarının başarılı bir şekilde sınıflandırıldığı görülmektedir. KO yönteminde önceki sonuçlarda olduğu gibi ilk olasılık değerleri performansı önemli ölçüde etkilemiştir. Markov modelinde etkisi kapı başarısını artırmaktan ziyade oda ve koridor sınıflarının başarısını artırmak yönünde olmuştur. Sonuçlarda özellikle göze çarpan nokta kapı sınıfının başarı oranının oldukça düşük olmasıdır. Bunun, sebebi ise Freiburg 79 binasında eğitilen veri kullanılarak ESOGÜ ortamında test yapılmasıdır. Binalarda genellikle oda ve koridorlar birbirlerine benzemektedir. Ancak, kapılar farklılık gösterebilmektedir.

KO ve ÖVN yöntemleri için elde edilen en iyi sonuçların görselleri ise sırasıyla Şekil 6.19 ve Şekil 6.20'de verilmiştir. Şekillerde kırmızı renkteki örnekler oda sınıfını, yeşil renkteki örnekler koridor sınıfını ve mavi renkteki örnekler ise kapı sınıfını temsil etmektedir. Şekiller incelendiğinde ortamın sağında ve solunda kalan koridorların her iki yöntem için de oda olarak bulunmuştur. Ayrıca, kapı sınıfına ait olması beklenen noktaların sıklıkla oda sınıfına ait bulunduğu her iki şekilde de gözükmemektedir. Bunların sebebi ise bu ortam için herhangi bir eğitim yapılmamış olup Freiburg 79 ortamında eğitilerek elde edilen bilgilerin kullanılmasıdır.



Şekil 6.19: KO K =1000, EO için görsel sonuç

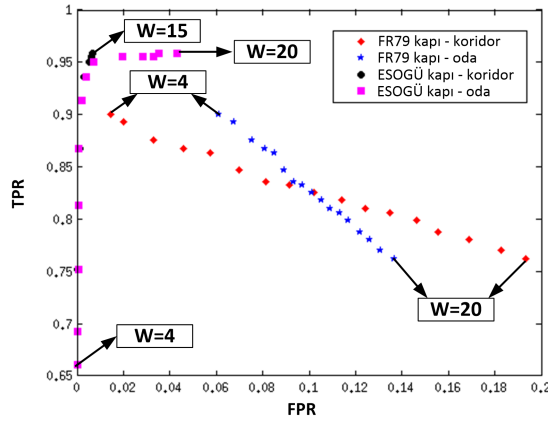


Şekil 6.20: ÖVN K =1000 için görsel sonuç

6.3.4 Kural tabanlı kapı bulma algoritması parametre analizi

Kural tabanlı kapı bulma yöntemi Freiburg 79 binasının ve ESOGÜ Laboratuvar binasının lazer mesafe verisi kullanılarak test edilmiştir. İlk olarak, parametrelerin kapı bulma yöntemi üzerindeki etkisi incelenecektir. T_L parametresi bütün deneylerde sıfır olarak seçilmiştir. Bu varsayım Freiburg verisi için kabul edilebilir olmakla birlikte, Gazebo benzetimleri ve gerçek zamanlı uygulamalarda, T_L parametresi robotun boyutlarına göre değişebilmektedir. Kapı-oda ve kapı-koridor konumlarının sınıflandırılmasının doğruluğu için deneyler yapılmıştır. Algoritmanın verimliliğinin ve kullanılan parametrelerin verimlilik üzerinde etkilerinin gözlemlenebilmesi için Karar Değerlendirme Grafiği (KDG) kullanılmıştır.

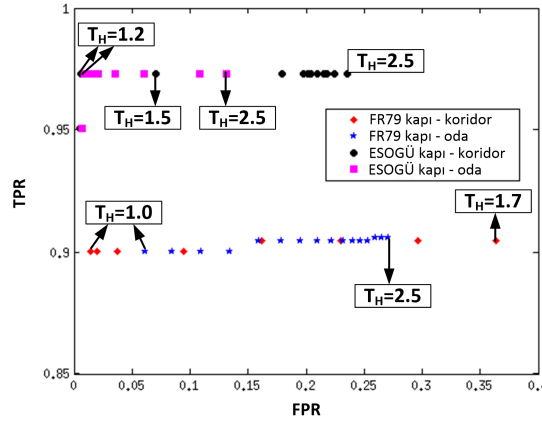
Şekil 6.21, 6.22, ve 6.23'te, bir seferde sadece bir parametre değiştirilerek elde edilen KDG eğrileri gösterilmiştir. Dikey eksen doğru pozitif oranını (TPR), yatay eksen de yanlış pozitif oranını (FPR) göstermektedir. Hem kapı-koridor hem de kapı-oda durumları için iki ayrı KDG eğrisi ile her parametrenin etkisi incelenmiştir. Kırmızı dörtgenler ve mavi yıldızlar Freiburg 79 veri kümesi için kapı-koridor ve kapı-oda durumlarının sonuçlarını göstermektedir. Benzer şekilde, siyah çemberler ile pembe kareler ESOGÜ ortamı için olan sonuçları ifade etmektedir.



Şekil 6.21: W parametresinin etkisi

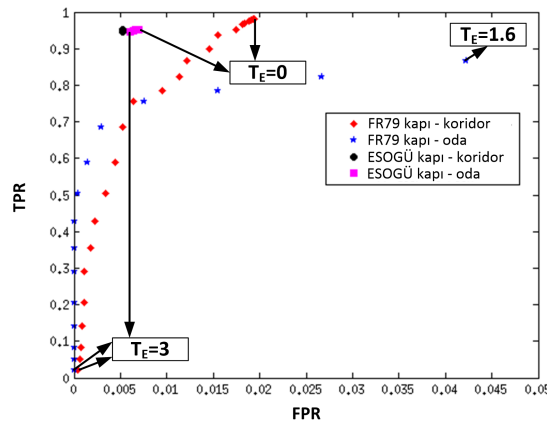
Şekil 6.21'de W parametresi değiştirilerek elde edilen KDG eğrisi gösterilmiştir. T_H ve T_E , Freiburg 79 veri kümesi için, sırasıyla 1 ve 1.5 metre olarak ayarlanmıştır. ESOGÜ ortamı için ise, T_H ve T_E parametreleri 1.2 ve 2 metreye ayarlanarak KDG eğrisi elde edilmiştir. W değerinin etkisi, bir konumun kapı olarak sınıflandırılabilmesi için kapı çerçevesine ne kadar yakın olduğu ile belirlenmektedir. Freiburg veri kümesi için, örnekler kapı çerçevesi içinde ise kapı örneği olarak atanmıştır. Şekil 6.21'de görüldüğü üzere,

Freiburg 79 ortamı için en yüksek başarımlar W değeri 4 olduğunda alınmıştır. ESOĞÜ ortamının gerçek referans değerinde, kapı çerçevelerinin 0.5 metre önünde ve arkasında olan her örnek kapı örneği olarak alınmıştır. En iyi sonuçlar W değeri 15 olduğunda elde edilmiştir.



Şekil 6.22: T_H parametresinin etkisi

Şekil 6.22'de T_H için olan KDG eğrisi verilmiştir. T_H parametresi kapının çerçeveleri arasındaki yaklaşık mesafeyi (kapı genişliği) tanımlamak için kullanılmıştır. Bina içi ortamlarda genellikle bu uzaklık 0.7 ile 1.3 metre arasındadır. Şekilde görüleceği gibi bu parametrenin aynı ortamda olduğunda TPR üzerinde herhangi bir etkisi yoktur. Bununla birlikte, her iki ortamda da T_H artarken FPR değeri de artmaktadır. En düşük FPR değeri Freiburg 79 ve ESOĞÜ ortamları için T_H parametresinin 1 ve 1.2 metre olduğunda elde edilmiştir. Kapı genişliği farklı ortamlar için değişiklik gösterebileceğinden, T_H değeri Freiburg 79 ve ESOĞÜ ortamları için sırasıyla 1 ve 1.2 metre olarak seçilmiştir.



Şekil 6.23: T_E parametresinin etkisi

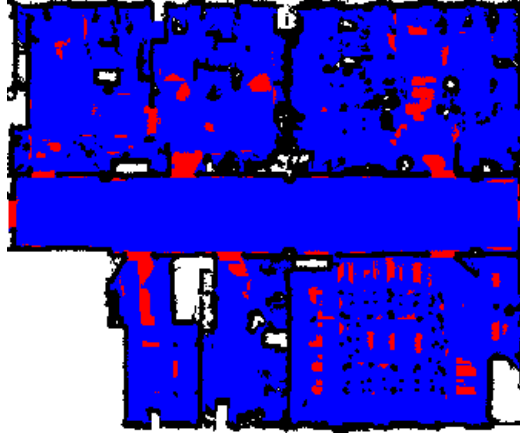
T_E parametresi paralel doğrular elde edildikten sonra kapının darboğaz özelliğini taşıyıp taşımadığını belirlemek için kullanılmıştır. Bu parametre ile robotun binanın köşeleri civarında hareket etmesi durumunda oluşan yanlış pozitif sonuçlar elenmiştir. Şekil 6.23'te görüldüğü üzere, parametre ESOGÜ ortamında kural tabanlı kapı bulma algoritmasının verimliliğini etkilememektedir. Bunun nedeni de robotun çarpışmalardan sakınmak amacıyla duvar veya diğer nesnelere fazla yaklaşmamasıdır. Freiburg 79 veri kümesinde ise duvar veya kapı çerçevesi olmayan diğer nesnelere çok yakın örnekler bulunmaktadır. Bu nedenle, T_E parametresi Freiburg 79 verisi için sonuçları etkilemektedir.

Çizelge 6.4'te önerilen kural tabanlı kapı bulma yönteminin sonuçları Freiburg 79 ortamı için özetlenmiştir. Test sonuçları kapı-koridor ve kapı-oda durumları için ayrı ayrı verilmiştir. Oda FPR, kapı-oda durumu için yanlış pozitif oranını, Kor FPR ise kapı-koridor durumu için yanlış pozitif oranını ifade etmektedir. TPR de her iki durum için doğru pozitif oranlarını göstermektedir.

Çizelge 6.4: Kapı Bulma Algoritması Freiburg 79 Sonuçları

W	T_H	T_E	Oda FPR	Kor FPR	TPR
15	1	1.5	11.3	13.5	80.6
20	1	1.5	13.7	19.3	76.2
4	1.2	1.5	10.9	3.8	90
4	1.5	1.5	17.9	22.9	90.4
4	1	1	21.4	1.2	97.4
4	1	2	0.3	0.5	68.5
4	1	1.5	6.1	1.5	90

Freiburg 79 veri kümesi için, W artıkça TPR azalmakta, FPR artmaktadır. Kapı-koridor durumu için, W artıkça yöntemimiz kapı konumlarına karşılık gelmeyen daha fazla sayıda paralel yapı bulmaktadır, sonuç olarak da kapı-oda durumuna göre daha düşük bir başarımla elde edilmektedir. T_H artırıldığında, TPR etkilenmemekle birlikte FPR keskin biçimde düşmektedir. Daha belirgin bir etki kapı-koridor durumu için gözlenebilmektedir. Son olarak, T_E artırıldığı zaman oda örneklerindeki yanlış pozitifler elenmektedir. Çizelge 6.4'ün son satırında %90 TPR, düşük Oda FPR ve Kor FPR olan makul bir sonuç verilmiştir. Önerilen yöntem, oda ve koridor örnekleri için düşük yanlış pozitif oranlarıyla birlikte %90 doğru kapı bulma başarısı göstermektedir. Şekil 6.24'te Freiburg 79 ortamı için görsel sonuçlar verilmiştir. Şekilde kırmızı noktalar önerilen yaklaşımın bulduğu kapı konumlarını göstermektedir.

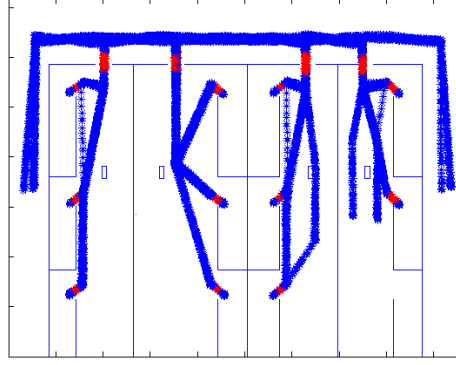


Şekil 6.24: Freiburg 79 $W = 4$, $T_H = 1$, $T_E = 1.5$ için görsel sonuç

Çizelge 6.5: Kapı Bulma Algoritması ESOGÜ Sonuçları

W	T_H	T_E	Oda FPR	Kor FPR	TPR
5	1.2	2	0	0	60.1
20	1.2	2	4.3	0.7	95.8
15	1.5	2	1.4	7.1	97.3
15	1.8	2	2	20.2	97.3
15	1.2	1	0.7	0.5	95.1
15	1.2	3	0.6	0.5	94.6
15	1.2	2	0.7	0.5	95.1

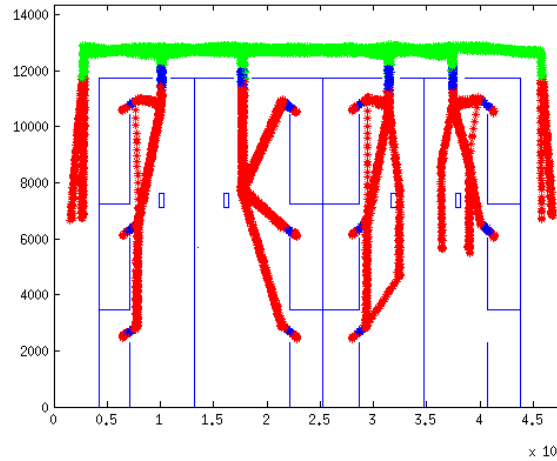
Çizelge 6.5'te önerilen kural tabanlı kapı bulma yönteminin sonuçları ESOGÜ ortamı için özetlenmiştir. ESOGÜ ortamında, daha önce de bahsedildiği üzere, T_E parametresinin önerilen yöntemin başarımı üzerinde bir etkisi bulunmamaktadır. T_H parametresinin Freiburg 79 veri kümesindeki benzer bir etkisi olmaktadır: T_H arttıkça TPR değişmemekte; ancak, özellikle kapı-koridor durumu için, koridor verisindeki paralel doğrular yanlış kapı algılamalarına, dolayısıyla da yüksek FPR oranına neden olmaktadır. Bu ortam için Freiburg 79 ortamından daha yüksek bir W değerine ihtiyaç duyulmaktadır. Bunun nedeni de kapı çerçevelerinin 0.5 metre önü ve arkasındaki alanda bulunan bütün örneklerin, gerçek referans değerinde kapı örneği olarak tanımlanmasıdır. Freiburg 79 ortamından farklı olarak, W değeri azaldıkça TPR oranı azalmaktadır. Bunun nedeni de bazı kapı örneklerinin algılanamamış olmasıdır. Çizelge 6.5'in son satırında, hem oda hem de koridor örnekleri için düşük yanlış pozitif oranları ile %95 doğru kapı algılama başarısı gösteren makul bir sonuç verilmiştir. Şekil 6.25'te ESOGÜ ortamı için görsel sonuçlar verilmiştir. Şekilde kırmızı noktalar önerilen yaklaşımın bulduğu kapı konumlarını göstermektedir.



Şekil 6.25: ESOGU $W = 15$, $T_H = 1.2$, $T_E = 2$ için görsel sonuç

6.3.5 ESOĞÜ yörünge verisi anlamsal sınıflandırma ve kapı bulma sonuçları

Bu noktadan sonra anlamsal sınıflandırma ve kapı bulma yöntemleri birleştirilmiş ve ESOĞÜ Laboratuvar binasının lazer mesafe verisi kullanılarak test edilmiştir. Elde edilen sayısal sonuçlar Çizelge 6.6'da verilmiştir. Çizelgedeki sonuçlar incelendiğinde kapı bulma yönteminin oda ve koridor sınıflarının başarı oranlarını etkilemediği gözükmemektedir. Bununla birlikte, kapı sınıfının başarı oranının %95'in üzerine çıktığı görülmektedir.

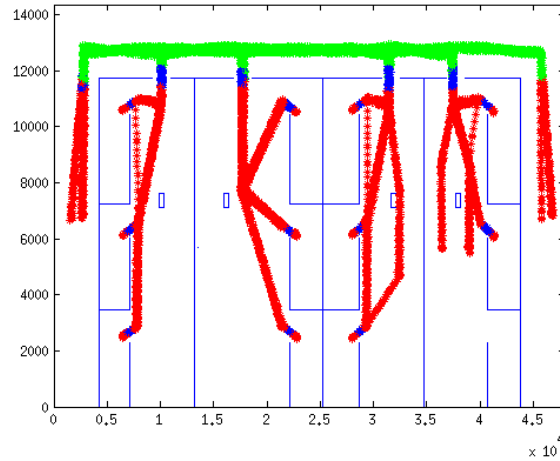


Şekil 6.26: KO $K = 40$, V için görsel sonuç

KO ve ÖVN yöntemleri için elde edilen en iyi sonuçların görselleri ise sırasıyla Şekil 6.26 ve Şekil 6.27'de verilmiştir. Şekiller incelendiğinde kapı sınıfına ait olması gereken noktaların kapı olarak bulunduğu gözükmemektedir. Bununla birlikte ortamın sağında ve solunda kalan koridorların her iki yöntem için de oda olarak bulunması problemi devam etmektedir. Ortamın bu kısımları için ek olarak bir çalışma yapılmamıştır.

Çizelge 6.6: ESOGÜ Yörünge Verisi Kapı Bulma ve Anlamsal Sınıflandırma Sonuçları

	Yöntem	Oda	Koridor	Kapı	Ortalama	Sınıf Ortalaması
K =40	KO,EO	42.78	89.12	98.28	66.58	76.73
	KO,V	99.31	91.94	95.09	93.98	95.44
	KO,Mar.	51.88	91.86	95.09	72.15	79.61
	ÖVN	75.27	91.88	95.58	82.92	87.58
	ÖVN,Mar.	79.32	91.94	95.58	84.81	88.95
K =100	KO,EO	56.20	90.89	95.09	73.63	80.73
	KO,V	99.31	90.89	95.09	93.43	95.09
	KO,Mar.	64.61	90.87	95.09	77.48	83.52
	ÖVN	96.73	91.71	95.09	92.68	94.51
	ÖVN,Mar.	96.27	91.77	95.09	92.50	94.38
K =400	KO,EO	89.35	91.71	95.33	89.29	92.13
	KO,V	95.19	91.73	95.33	91.98	94.08
	KO,Mar.	90.86	91.73	95.09	90	92.56
	ÖVN	99.17	91.93	95.33	93.92	95.48
	ÖVN,Mar.	98.81	91.81	95.09	93.68	95.23
K =1000	KO,EO	88.86	91.92	97.05	89.20	92.61
	KO,V	95.73	91.99	95.33	92.37	94.35
	KO,Mar.	90.09	91.94	95.09	89.75	92.37
	ÖVN	99.30	91.93	95.58	93.99	95.61
	ÖVN,Mar.	99.18	91.94	95.09	93.93	95.40



Şekil 6.27: ÖVN K =1000 için görsel sonuç

6.3.6 ESOGÜ tek robotlu anlamsal sınıflandırma tabanlı keşif sonuçları

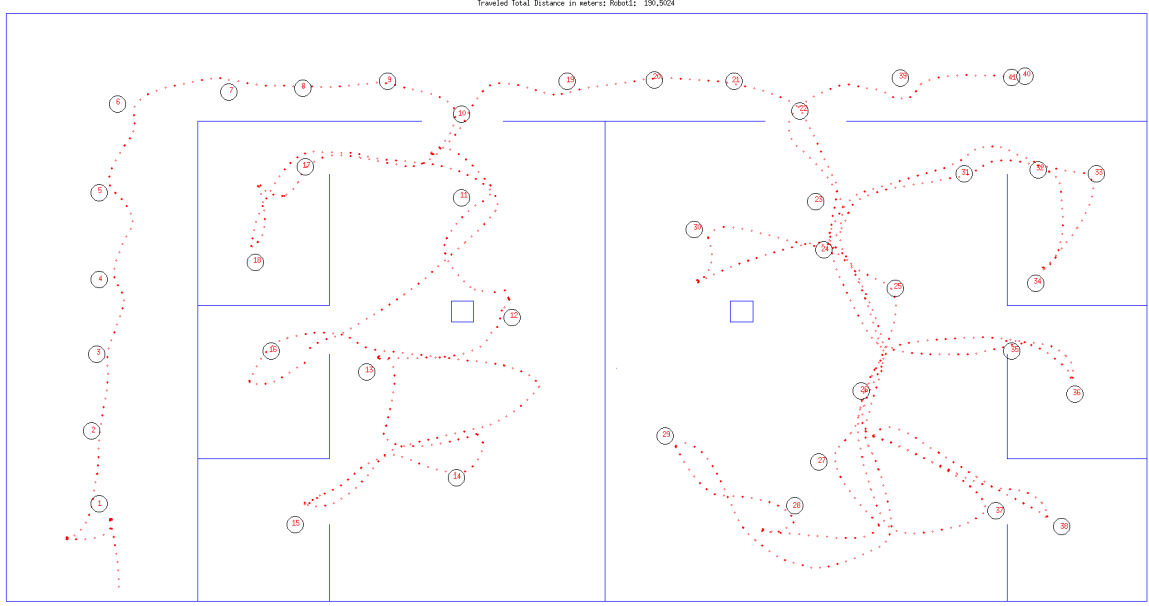
Önerilen yöntem, Gazebo benzetim ortamında modellenen Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Laboratuvar binasında test edilmiştir. Ortamın boyutları 26×15 metredir ve 52×30 'luk metrik ızgara ile ifade edilmiştir. Uygulamalar gerçekleştirilirken robotun oda öncelikli rolde olduğu kabul edilmiştir. Robot, diğer tek robotlu keşif yöntemleri ile aynı olan üç farklı yerden görevine başlamakta ve bütün ortamın keşfi tamamlanana kadar görevini sürdürmektedir.

Çizelge 6.7'de hedeflerin anlamsal sınıflandırma bilgisi ile belirlendiği keşif yaklaşımı ve Dijkstra yol planı ile üç farklı fiyatlandırma kriteri kullanılarak gerçekleştirilen keşif yaklaşımlarının sayısal sonuçları verilmiştir. Sonuçlar süre açısından incelendiğinde anlamsal sınıflandırma bilgileri kullanarak yapılan keşfin en iyi sonuçları ürettiği açıkça görülmektedir. Bunun sebebi, robotun tekrar geçtiği alanların azalmasıdır. Bununla birlikte, robotun katettiği mesafenin ortamın topolojik haritası ile doğrudan alakalı olduğu önceki bölümlerde vurgulanmıştır. Bu test sonuçlarında da, mesafe ile ilgili bir genelleme yapılamamaktadır.

Çizelge 6.7: Tek Robot Anlamsal Sınıf Bilgisi Sonuçları

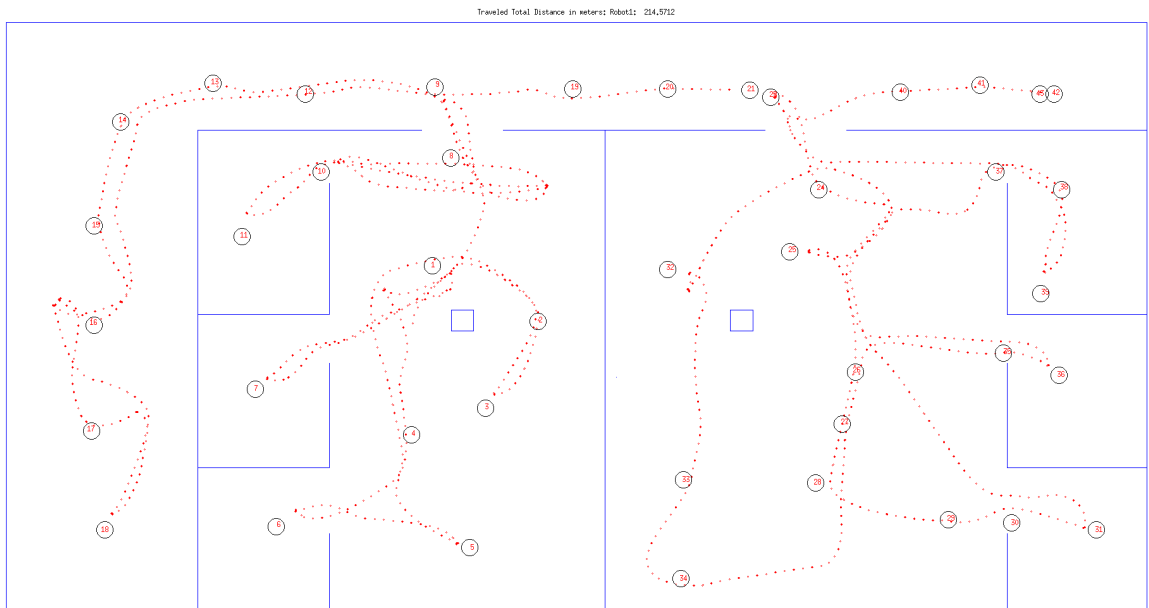
Başlangıç Konumu	Hedef Seçme Kriteri	Mesafe (m)	Süre (Dk)
2.5, 0.3, 1.57	En çok kenar hücresi	225.1777	72
	En yakın	188.3614	67
	Market	201.4353	68
	R1 Oda	190.5024	55
9, 7, 1.57	En çok kenar hücresi	276.1921	82
	En yakın	220.8951	71
	Market	197.6729	66
	R1 Oda	214.5712	62
24, 4, 1.57	En çok kenar hücresi	286.2189	80
	En yakın	231.8223	72
	Market	241.0084	74
	R1 Oda	220.6191	60

Şekil 6.28'de robotun 24, 4, 1.57 konumundan başladığı durumdaki anlamsal sınıflandırma bilgisini ile yaptığı hedef seçimleri görünmektedir. Robot oda öncelikli rolünde olduğu için başta ilk büyük odayı bütün küçük odaları ile keşfetmekte ve daha sonra ikinci büyük oda da aynı süreci işletip en son koridorun sonunu keşfetmektedir.



Şekil 6.28: Anlamsal sınıflandırma bilgisini hedef seçme kriteri Konum (1)

Şekil 6.29'da ise robotun 9,7,1.57 konumundan başladığı durumdaki anlamsal sınıflandırma bilgisini ile yaptığı hedef seçimleri görünmektedir. Robotun 8. hedefi büyük odanın kapısının önündedir. Sonraki hedef (9. hedef) koridordadır. Robot buraya çıkmakta ve buranın koridor olduğunu anlamakta ve tekrar büyük odaya (10. hedef) dönmektedir. Sonuç olarak, robotun başladığı üç durumda da robot başta odaları keşfetmekte ve daha sonra koridor hedeflerine yönelmektedir.



Şekil 6.29: Anlamsal sınıflandırma bilgisini hedef seçme kriteri Konum (2)

6.3.7 ESOGÜ çok robotlu anlamsal sınıflandırma tabanlı keşif sonuçları

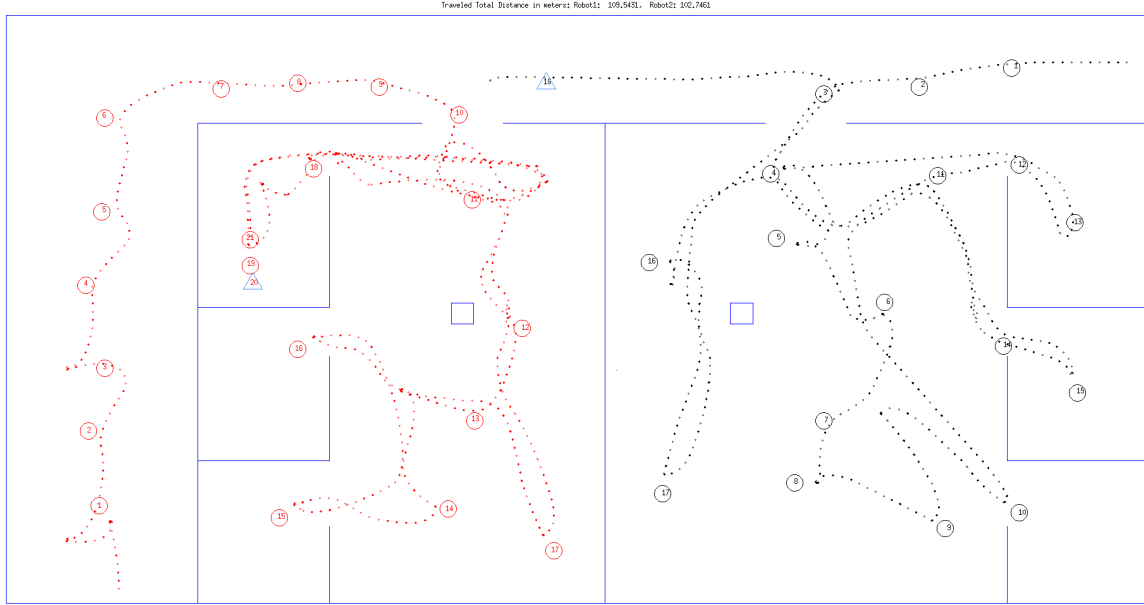
Uygulamalar gerçekleştirilirken ilk olarak her iki robotada oda öncelikli robot rolü verilmiştir. Daha sonra birinci robotun koridor öncelikli ve ikinci robotun ise oda öncelikli robot rolünde olduğu testler gerçekleştirilmiştir. Son olarak, ikinci durumdaki roller değiştirilmiştir. Robotlar, diğer çok robotlu keşif yöntemleri ile aynı olan üç farklı yerden görevine başlamaktadır.

Çizelge 6.8: İki Robot Anlamsal Sınıf Bilgisi Sonuçları

Fiyatlandırma Kriteri	Konum		Süre (Dk)	Mesafe(m)	
	R1	R2		R1	R2
En çok kenar hücresi			52	152.02	183.37
En yakın	2.5, 0.3, 1.57	24.8, 13.2, 3.14	34	101.37	119.07
Market			35	107.54	116.78
R1 Oda R2 Oda			31	109.54	102.75
R1 Koridor R2 Oda			42	120.35	138.43
R1 Oda R2 Koridor			42	130.74	160.71
En çok kenar hücresi					57
En yakın	16, 0.3, 1.57	24, 4, 1.57	37	118.96	124.33
Market			44	166.85	131.69
R1 Oda R2 Oda			34	119.56	131.21
R1 Koridor R2 Oda			40	138.75	144.96
R1 Oda R2 Koridor			40	138.87	115.53
En çok kenar hücresi					67
En yakın	9, 7, 1.57	11, 7, 1.57	38	121.46	114.69
Market			41	128.30	135.55
R1 Oda R2 Oda			52	169.07	167.20
R1 Koridor R2 Oda			38	126.12	136.77
R1 Oda R2 Koridor			40	131.44	137.44

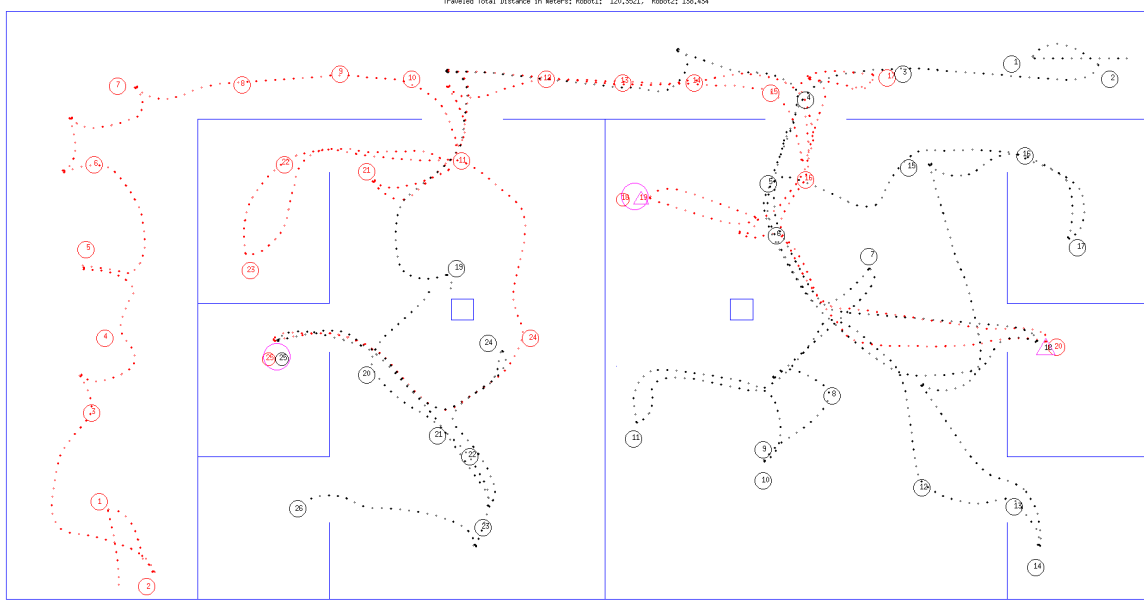
Çizelge 6.8'de hedeflerin yukarıda anlatılan üç durum ile belirlendiği keşif yaklaşımı ve üç farklı fiyatlandırma kriteri kullanılarak gerçekleştirilen keşif yaklaşımlarının sayısal sonuçları verilmiştir. Konum 1, birinci robotun 2.5, 0.3, 1.57 ve ikinci robotun ise 24.8, 13.2, 3.14; konum 2, birinci robotun 16, 0.3, 1.57 ve ikinci robotun ise 24, 4, 1.57; konum 3, birinci robotun 9, 7, 1.57 ve ikinci robotun ise 11, 7, 1.57 konumundan başladığı durumları anlatmaktadır. Çizelgede görüldüğü gibi hem mesafe hem de süre açısından her başlangıç konumu için en kötü sonuçları En çok kenar hücresi kriteri üretmiştir.

Konum 1 için iki robotunda oda öncelikli olduğu durum en iyi sonucu vermektedir. Şekil 6.30'da robotların hedef seçimleri görünmektedir. Her iki robotta başlangıçta koridordadır. Sececekleri başka hedef olmadıkları için ilk oda hedefini bulana kadar koridorda ilerlemektedirler. Daha sonra her ikisi de farklı büyük odaya girmekte ve odayı tamamen keşfedene kadar burada kalmaktadır. Keşif sona ererken ilk büyük odanın kapısına yakın bir yerde karşılaşmakta ve keşfi tamamlamaktadırlar.



Şekil 6.30: Konum 1 R1 ve R2 oda öncelikli kriteri

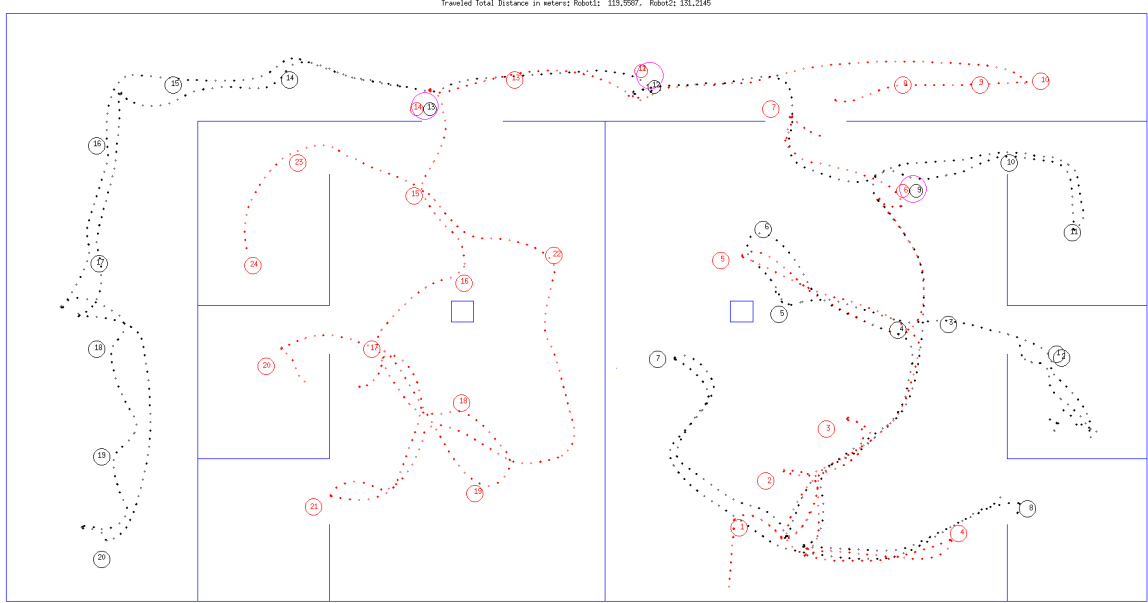
Birinci robotun koridor ikinci robotun ise oda öncelikli olduğu durumda ise market ve en yakın kriterlerinden daha kötü sonuç üretmiştir. Şekil 6.31'de robotların hedef seçimleri görünmektedir. Şekilde görüldüğü gibi birinci robot 10. hedefine geldiğinde içeri girmekte (11. hedef) ancak önceliği koridor olduğu için 12. hedefinde tekrar koridora dönmektedir. Benzer bir durum ikinci büyük odanın kapısında 15, 16 ve 17. hedeflerde de görünmektedir. Robot 17. hedefteyken diğer robotla haberleşmiş ve koridorun kalan kısmının keşfedildiğini öğrenmiş ve en yakın hedefine gitmiştir. Koridor öncelikli robotlar eğer koridor hedefi yoksa maliyeti azaltmak için en yakın oda hedefini seçmektedir. Benzer durum oda robotları içinde geçerlidir. Diğer yandan, oda öncelikli robot oda hedefleri arasından en kârlı olanı seçmektedir. Benzer durum koridor robotları içinde geçerlidir. Bu sayede robotlar tamamen aynı hedef adaylarına sahip olsalar dahi farklı görevleri seçmektedirler. Buna bir örnek birinci robotun 20. ve ikinci robotun 18. hedeflerine vardıklarında gerçekleşmektedir. Koridor öncelikli robot daha yakın olan 21. hedefi seçerken oda öncelikli robot daha kârlı olan 19. hedefi seçmiştir.



Şekil 6.31: Konum 1 R1 koridor ve R2 oda öncelikli kriteri

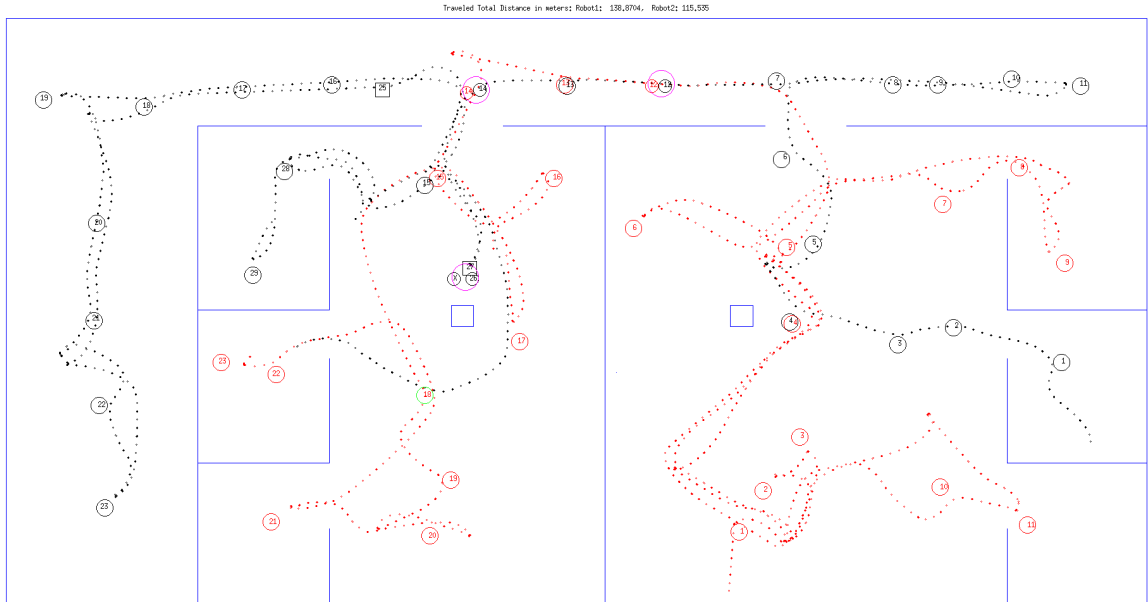
Konum 2 için iki robotunda oda öncelikli olduğu durum en iyi sonucu vermektedir. Şekil 6.32'de robotların hedef seçimleri görünmektedir. Her iki robotta başlangıçta odadadırlar. Dolayısıyla kendilerine en çok kâr getirecek hedefleri seçmektedirler. Birinci robot 6. hedefindeyken ikinci robotta 9. hedefine doğru yol almakta ve robotlar birbirlerine yakın oldukları için haritaları hemen hemen aynı olmaktadır. Bu şekilde ikinci robot küçük odaya (10. hedef) giderken birinci robotta koridora (7. hedef) çıkmıştır. Bu anlarda robotlar sürekli haberleşme menzili içinde oldukları için görevler paylaşılıp robotlar ayrılmaktadır. Soldaki büyük odanın girişinde benzer şekilde robotlar bir daha ayrılmaktadırlar. Bu noktada haberleşmeleri kesildiği için ikinci robot odaya dönmemiştir.

İkinci robotun koridor birinci robotun ise oda öncelikli olduğu durumda ise market kriterinden daha iyi ve en yakın kriterinden daha kötü sonuç üretmiştir. Şekil 6.33'te robotların hedef seçimleri görünmektedir. Şekilde görüldüğü gibi ikinci robot en yakın hedefleri seçerek koridora çıkmaktadır. Bu anlarda robotlar haberleşme menziline oldukları için oda öncelikli robot koridorlardaki hedeflerden haberdar olmuştur. Dolayısıyla, 9. hedefinden sonra odada bulunan 10 ve 11. hedefleri seçmiştir. Koridor öncelikli robot ise önce koridorun sağ tarafını sonra da sol tarafını soldaki büyük odanın kapısına (14. hedef) gelene kadar keşfetmektedir. Bu noktada içeri girmiş (15. hedef) ve bu hedefin odada bulunduğunu anlayınca tekrar koridora çıkmıştır (16. hedef). Koridor öncelikli robot sol taraftaki koridoru keşfedip tekrar büyük odanın kapısına X ile gösterilen hedefe gitmeyi planlamaktaydı. Aynı sırada oda öncelikli robotta soldaki büyük odanın kapısına gelmişti (14. hedef). Çarpışma protokolü uygulandığında Durum 5 ortaya çıktı ve



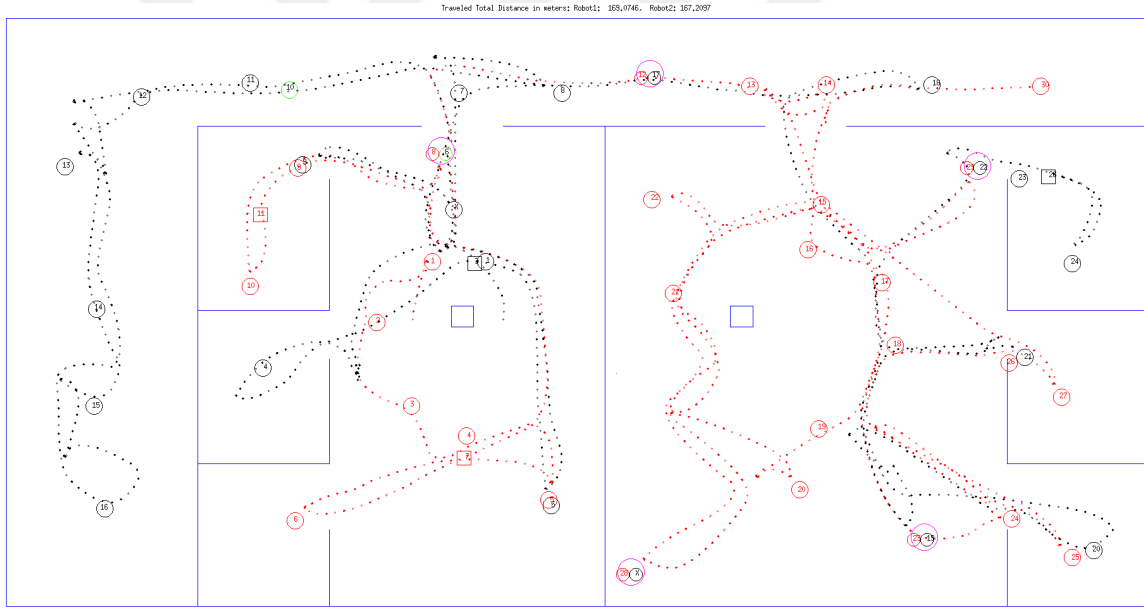
Şekil 6.32: Konum 2 R1 ve R2 oda öncelikli kriteri

koridor öncelikli robot beklerken (25. hedef) oda öncelikli robot 15. hedefine doğru hareket etmektedir. Daha önce anlatıldığı gibi farklı rollerdeki robotlar farklı hedefler seçtiği için robotlar büyük odaya dağılmış ve görevlerini tamamlamışlardır.



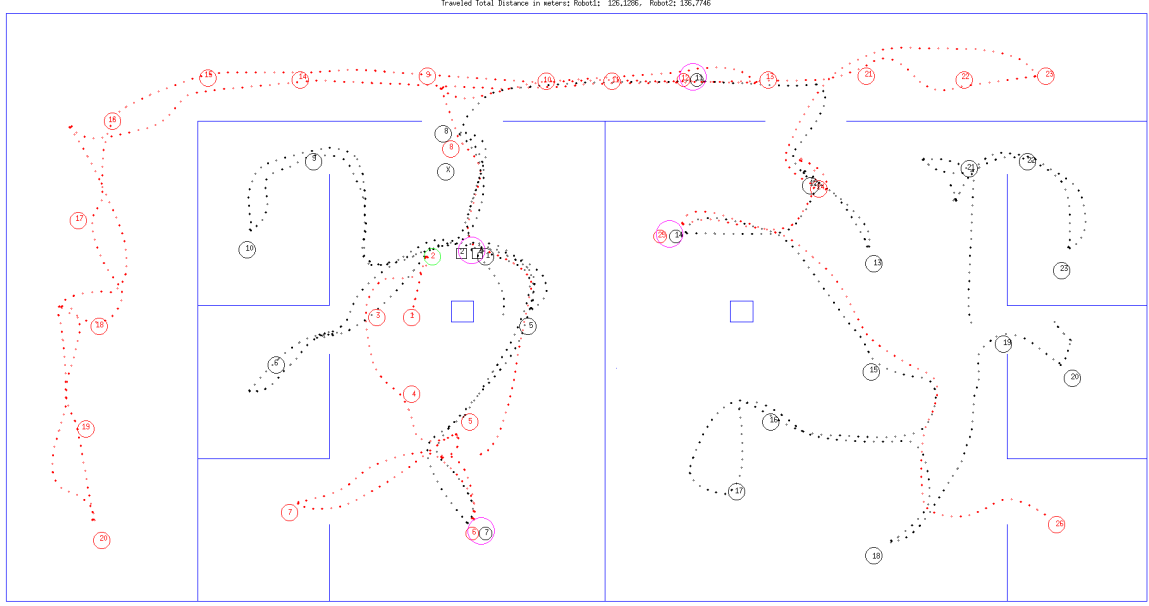
Şekil 6.33: Konum 2 R1 oda ve R2 koridor öncelikli kriteri

Konum 3 için iki robotunda oda öncelikli olduğu durum market ve en yakın kriterlerinden daha kötü sonuç üretmiştir. Şekil 6.34'te robotların hedef seçimleri görünmektedir. Her iki robotta başlangıçta odada ve birbirlerine çok yakındırlar. Dolayısıyla her ikisi de aynı haritaya sahip ve en kârlı hedefleri seçmek istemektedirler. Bu durumda robotlar sürekli olarak ayrılıp tekrar bir araya gelmektedir. Örneğin ikinci robot 8. hedefine vardığında 9. hedefle birlikte tekrar odaya dönmekte ve tekrar diğer robotun yanına gitmektedir. Bu sırada haberleşme koptuğu için birbirlerinden haberleri olmadan aynı hedefleri seçmektedirler. Örneğin, birinci robot 10. hedefinden yola çıkarken ikinci robotta 9. hedefinden yola çıkmak istemektedir. İkiside aynı hedefi (yeşille gösterilen 10. hedef) seçmişlerdir ve çarpışma protokolündeki Durum 2 oluşmuştur. Bu durumda birinci robot 11. hedefinde beklemekte ve ikinci robotta ortak hedefe gitmektedir. Bu gibi sürekli ayrılıp biraraya gelmeler keşif boyunca sürmekte ve robotlar koordine olmakta zorluk yaşamaktadır.



Şekil 6.34: Konum 3 R1 ve R2 oda öncelikli kriteri

Birinci robotun koridor ikinci robotun ise oda öncelikli olduğu durumda ise market ve en yakın kriterleriyle hemen hemen aynı sonuç üretmiştir. Şekil 6.35'te robotların hedef seçimleri görünmektedir. Her iki robotta başlangıçta odada ve birbirlerine çok yakındırlar. Keşfin ilk anlarında birbirleri ile yanyana olmakla beraber farklı rollere sahip oldukları için kısa sürede birbirlerinden ayrılmaktadırlar. Birinci robot koridorları keşfederken ikinci robot ise odalara odaklanmaktadır. Bu sayede birbirleri ile aynı yolu daha az katetmekte ve daha verimli bir keşif gerçekleştirmektedirler.



Şekil 6.35: Konum 3 R1 koridor ve R2 oda öncelikli kriteri

Son olarak robotların başlangıç konumlarının ve aldığı rollerin koordinasyonları üzerindeki etkiler tartışılmalıdır. Çizelge 6.8'de keşif süresi ve katedilen mesafe olarak verilen sonuçların yanı sıra robotların kaç kere bekledikleri veya hedef değiştirdikleri de bu konuda bazı fikirler vermektedir. Çizelge 6.9'da ise robotların keşif süresince bekleme ve hedef değiştirme miktarları verilmektedir. Konum 1 gibi robotların birbirinden uzakta başladığı durumlarda her iki robota da aynı rolün verilmesi en iyi sonucu vermektedir. Benzer şekilde Konum 2'de de robotlar birbirinden haberleşme menziline daha fazla bir mesafeye sahiptir. Dolayısıyla robotların aynı rolde olması verimliliği arttıracaktır. Bununla beraber, Konum 3'te olduğu gibi robotlar birbirlerine çok yakın konumlardan keşfe başladıklarında robotların farklı roller alması hızlı bir şekilde birbirlerinden ayrılmasını sağlayarak koordinasyonu iyileştirmektedir. Anlamsal sınıflandırma ile keşif yapmanın süre, mesafe ve koordinasyon verimliliğinin yanı sıra keşif stratejisinin belirlenebilmesidir. Ortamın geometrik yapısı, robotların başlayacakları noktalar göz önüne alınarak istenilen strateji robotlara uygun roller verilerek sağlanabilir.

Çizelge 6.9: Çarpışma Önlem Protokolü Anlamsal Sınıf Bilgisi Sonuçları

Fiyatlandırma Kriteri	Konum	Değişim	Bekleme
En çok kenar hücresi		1	2
En yakın	2.5, 0.3, 1.57	1	1
Market	24.8, 13.2, 3.14	1	0
R1 Oda R2 Oda		1	0
R1 Koridor R2 Oda		1	0
R1 Oda R2 Koridor		3	0
En çok kenar hücresi		6	3
En yakın	16, 0.3, 1.57	1	1
Market	24, 4, 1.57	1	0
R1 Oda R2 Oda		0	0
R1 Koridor R2 Oda		4	2
R1 Oda R2 Koridor		0	2
En çok kenar hücresi		4	3
En yakın	9, 7, 1.57	0	1
Market	11, 7, 1.57	1	2
R1 Oda R2 Oda		0	4
R1 Koridor R2 Oda		0	2
R1 Oda R2 Koridor		3	3

7. AFET ORTAMINDA ANLAMSAL TABANLI KEŞİF YÖNTEMİ

Afet sonrası bina içi ortamlarda robotlar tarafından yapılması istenen görev en genelde ortamın keşfini yapmak ve bütünlüğü bozulmuş bölümler, afetzeler gibi önemli unsurları haritalamaktır. Robot takımının haberleşmenin kısıtlı olduğu bu ortamlarda olabildiğince iyi şekilde koordine olması istenmektedir. Bunun sağlanması için robotların yerel haritalarını birbirleri ile paylaşmaları, ortamın farklı bölgelerinde hedefler seçmeleri gibi konular ele alınmaktadır (Liu ve Nejat, 2013). Bu tez çalışması kapsamında robotların koordinasyonu ve hedef seçmeleri ile ilgili konular önceki bölümlerde yer almaktadır. Bu noktada, afet ortamında ortaya iki problem çıkmaktadır. Birincisi, algılama problemidir. Afet ortamında keşif yapılırken robotun lazer mesafe algılayıcı ve seyrüsefer planlanırken derinlik ve renk algılayıcısından yararlanılmaktadır. Robot eğimli, düz olmayan zeminlerde seyrüsefer yaparken bu algılayıcılardan elde ettiği veriler istenilenden farklı olmaktadır. Örneğin robot seyrüseferi sırasında yukarı doğru eğimlendiğinde lazer mesafe verileri ışınların duvarları aşması sebebiyle olması gerekenden daha uzun mesafeler ölçmektedir. Diğer yandan, robot aşağı doğru eğimlendiğinde ise yere baktığı için olması gerekenden daha kısa mesafeler ölçmektedir. Bu problemi aşmak için 3. Bölüm'de anlatıldığı gibi lazer ışınlarının uç noktasının hesaplanmasında yuvarlama ve yunuslama açıları da hesaba katılmaktadır. Buna ek olarak ışınların xy düzlemine izdüşümleri alınmaktadır. Afet ortamında karşılaşılan ikinci problem ise seyrüseferdir. Robotun rampaların bulunduğu bir ortamda nasıl seyrüsefer yapacağı bu bölümde verilmektedir.

Bu bölümde, ilk olarak geçmiş çalışmalarda yer alan belli başlı afet ortamında keşif, seyrüsefer ve haritalama yaklaşımları incelenecektir. Daha sonra, gerçekleştirilen yaklaşım alt bölümünde, rampaların algılanması ve ara noktanın belirlenmesi ve belirlenen ara noktaların seyrüseferde nasıl kullanılacağı anlatılmaktadır. Son olarak, afet ortamında anlamsal sınıflandırma kullanılarak hedefin belirlendiği keşif verimliliği tek ve çok robotlu sistemler için analiz edilecektir.

7.1 Geçmiş Çalışmalar

Ziparo ve diğerleri afet ortamında keşfin koordinasyonunu artırmak için RFID algılayıcısı kullanmışlardır. Önerdikleri yaklaşımda, robotların seyrüseferi sırasında bu algılayıcılar ile direk olmayan haberleşme sağlanmaktadır. Robotlar RFID algılayıcılara konumlarını ve ziyaret ettikleri konumları yazmaktadırlar. Bu şekilde aynı hedefe gitme

olasılıkları azalmaktadır. Küresel seviyede planlama için ise RFID çizgesi çıkarılmaktadır. Bu çizgede düğümler RFID pozisyonlarını ayırtlar ise düğümler arasındaki yolları göstermektedir. Bu çizge kullanılarak planlama yapıldığında hem robotların birbirleri ile karşılaşmaları önlenmekte hem de planlamanın zaman karmaşıklığı azalmaktadır (Ziparo vd. 2007).

Calisi ve diğerleri 2007 yılında yaptıkları çalışmada, çok amaçlı keşif yapılması gereken arama kurtarma görevlerinde geçmiş çalışmalarda yer alan tek amaçlı keşif yaklaşımlarının yetersiz olduğunu vurgulamışlardır. Birden çok kriteri göz önüne alan çalışmalarında da verilen karar sonucu tek bir hedefe odaklanıldığı belirtilmiştir. Bununla beraber, arama kurtarma görevlerinde bazı görevlerin birbiriyle çelişmesi gibi durumlar oluşmaktadır. Örneğin, harita oluşturmak için uzak bir noktadan veri toplamak yeterli olurken, afetzelerin belirlenmesi için afetzedenin yakınında olmak gerekmektedir. Ayrıca, keşif ve haritalama yapmak için bir hedefe doğru gidilirken bir afetzede ile karşılaştığında, robottan yapması beklenen afetzedenin yakınına giderek durumu ile ilgili bilgiyi en kısa sürede toplamasıdır. Bu gibi durumları ele almak, yüksek seviyede bir koordinasyon sağlamak amacıyla Petri Net kullanılarak planlama yapılmaktadır (Calisi vd. 2007).

Wirth ve Pellenz yaptıkları çalışmada afet ortamında kullanılacak bir yol planı yaklaşımı önermişlerdir. Bu ortamlarda, robotun dar geçitlerden geçmesinin yaratacağı seyrüsefer sorunları ve geniş bölümlerde algılayıcıdan sağlanan verilerin bilgi içermemesi durumları ele alınmıştır. Bu durumların önüne geçmek için mesafe dönüşümünü temel alan bir yol planlaması kullanılmıştır. Bu planda, robot ile hedef arasındaki en kısa mesafe ile birlikte robotun dar bölgelerden veya geniş bölgelerden geçmemesi de amaçlanmaktadır (Wirth ve Pellenz, 2007).

(Taillandier ve Stinckwich, 2011) ve (Basilico ve Amigoni, 2011) yaptıkları çalışmalarda çok kriterli karar verme yöntemlerini kullanmışlardır. Hedefin belirlenmesi sırasında robotun hedefe olan uzaklığı, hedefe gidildiğinde elde edilecek bilgi miktarı ve hedefe vardığında komuta merkezi ile haberleşebilir olma olasılığı kriterleri kullanılmıştır. Bu kriterler belirli katsayılar ile test edilmiştir. Yayınların birbirinden tek farkları ise kullandıkları çok kriterli karar verme yöntemleridir. Taillandier ve Stinckwich yaptıkları çalışmada PROMETHEE yöntemini kullanırken Basilico ve Amigoni ise Choquet integralini tercih etmiştir.

(Cipolleschi vd. 2013) ve (Quattrini Li vd. 2016) yaptıkları çalışmada, kullanıcı tarafından sağlanan bir bilgiye dayanarak ilgili bölgelerin hızlıca keşfedilmesi için anlamsal sınıflandırma bilgisini kullanmışlardır. Örneğin, afet çalışma saatlerinde meydana geldiyse afetzedeler çoğunlukla ofislerde bulunacaktır. Dolayısıyla ilk önce küçük odaların

keşfedilmesi gerekmekte ilgili bölgeler bu odalar olarak tanımlanmaktadır. Diğer yandan, eğer afet ögüle yemeđi saatinde gerçekteştiyse kantin gibi büyük odalara bakılması uygun olacaktır. Hedefin belirlenmesi sırasında robotun hedefe olan uzaklığı, hedefe gidildiğinde elde edilecek bilgi miktarı, hedefe vardığında komuta merkezi ile haberleşebilir olma olasılığı, gidilecek noktanın anlamsal sınıfının ilgili bölge olma olasılığı ve gidilecek noktadaki kapı sayısı kriterleri kullanılmıştır. Bu kriterler çok kriterli karar verme yöntemleri ile farklı katsayılarla test edilmiştir.

Geçmiş çalışmalar incelendiğinde afet ortamında keşif ve haritalama görevleri için çok çeşitli çözüm yollarının olduğu görülmektedir. Tez çalışması kapsamında, afet ortamında seyrüsefer yapabilmek amacıyla derinlik ve renk algılayıcı kullanılarak uygun ara noktaların bulunması ve robotun küresel hedefi ile birlikte kullanılması amaçlanmıştır. Keşif görevinde ise anlamsal sınıf bilgileri kullanılacaktır.

7.2 Gerçeklenen Yaklaşım

Afet sonrası bina içi ortamlar ile ilgili konularda yapılan araştırmalarda ve yarışmalarda genelde NIST'in standartlarını belirlediği rampa, merdiven ve küpler gibi malzemeler kullanılmaktadır (NISTTestAlanları, 2016). Bununla beraber, robotların hareket ve algılama yetenekleri farklı seviyelerde sınıyabilecek üç test alanı belirlemiştir (Şekil 7.1). Sarı, turuncu ve kırmızı olarak adlandırdıkları bu test alanlarının temel özellikleri önceki bölümlerde verilmiştir.



Şekil 7.1: Sarı, turuncu ve kırmızı test alanları

Şekil incelendiğinde, rampalar, merdivenler ve küpler ile oluşturulmuş ortamın robotun hareketini zorlaştıracığı görülecektir. Bu tez çalışması kapsamında iniş, çıkış ve düz rampalar ele alınacaktır. Bu rampaların belirlenmesinde görsel verilerin kullanılmasının deprem ortamı için uygun olmayacağı düşünülmüştür. Bu noktada, akla ilk gelen çözüm lazer mesafe algılayıcının kullanılmasıdır. Ancak, rampalar lazerin taradığı düzlemin altında kaldığı için kullanılamamaktadır. Sonuç olarak, rampaların belirlenmesi için renk ve derinlik (rgb-d) algılayıcıdan yararlanılmıştır.

Özellikle son yıllarda, gezgin robotlar da dahil olmak üzere çeşitli platformlarda bir takım görevler için renk ve derinlik algılayıcı kullanılması yaygınlaşmıştır. (Suarez ve Murphy, 2012)'daki çalışmada, renk ve derinlik algılayıcının robot seyrüseferi, nesne manipülasyonu, ortam haritalaması gibi popüler görevler için kullanıldığına da değinilmiştir. Son olarak, bina içi ortamlarda arama ve kurtarma görevleri için de kullanılabilir olduğu belirtilmiştir. Bu tez çalışması kapsamında afet robotun başarıyla seyrüsefer yapabilmesine yardımcı olması amacıyla ara noktalar hesaplanmıştır. Ön aşama olarak, renk ve derinlik algılayıcıdan elde edilen derinlik verileri nokta bulutuna çevrilmiştir. İlk olarak, nokta bulutundaki düzlemler ve tipleri belirlenmiştir. Daha sonra, uygun ara noktalara karar verilmiştir. Bu aşamaları gerçekleştirmek için PCL (Point Cloud Library) kullanılmıştır. Son olarak, robot ara nokta ve küresel hedefi göz önüne bulundurarak seyrüseferini planlamaktadır.

İlk olarak nokta bulutundaki düzlemlerin bölütlenmesi gerekmektedir. Bunun için, RANSAC (Random Sample Consensus) algoritmasından faydalanılmaktadır. RANSAC algoritması, Fischler ve Bolles tarafından 1981 yılında önerilmiştir (Fischler ve Bolles, 1981). Bu algoritma, veri kümesindeki uyumlu elemanlar için bir model oluşturmaktadır. Her adımda, uygun noktaların oluşturduğu bir küme bir düzlemi ifade etmektedir. Düzlemler belirlendikten sonra bu düzlemlerin sınıfına karar verilmektedir. Bu çalışmada, iniş rampası, çıkış rampası, düz rampa, duvar düzlemi ve yer düzlemi olmak üzere beş sınıf belirlenmiştir. Bu sınıflar için verilen örnekler Şekil 7.2'de gösterilmiştir. Eğimli bir rampanın iniş ya da çıkış rampası olduğuna karar verilirken robotun bakış açısı kullanılmaktadır. Bunun sebebi, iniş ve çıkış rampaları farklı yönlerden bakıldığında tam ters şekilde çıkış ve iniş olarak belirlenebilir.

Bir düzlemin sınıfına karar vermek için, o düzlemin z eksenindeki değerinin xy düzlemiyle olan açısına bakılmaktadır. Eğer bu açı 90° 'nin belirli bir komşuluğunda bir değere sahipse, bu düzlem duvar düzlemi olarak sınıflandırılmaktadır. Eğer düzlemin açısı 0° 'nin belirli bir komşuluğunda bir değere sahipse bu düzlem iki sınıftan (yer ya da düz rampa) birine aittir. Düzlemin hangi sınıfta olduğuna karar vermek için z eksenindeki değerinin xy düzlemiyle olan açısına bakılmaktadır. Eğer bu açı pozitifse düzlem yer



Şekil 7.2: Robotun bakış açısına göre rampaların sınıflandırılması

olarak, negatifse düz rampa olarak sınıflandırılmaktadır. Eğimli rampaları belirlemek için rampanın xy düzlemiyle yaptığı açının belirli bir aralıkta olup olmadığına bakılmaktadır. Eğer düzlem eğimli ise xz düzlemine yansıması alınmakta ve bu yansıma sonucu elde edilen doğru parçasının x eksenine arasındaki açıya bakılmaktadır. Eğer bu açı 90° 'den büyükse düzlem çıkış rampası, küçükse iniş rampası olarak sınıflandırılmaktadır.

Bütün düzlemlerin sınıfları belirlendikten sonra, hangi düzlemlerin ara nokta belirlemede kullanılacağına karar verilmektedir. Yer ve duvar sınıfındaki düzlemler ara nokta belirlemede kullanılmamıştır. Bununla birlikte, robotun z eksenindeki konumu da hangi düzlemlerin ara nokta belirlemede kullanılacağını belirlemektedir. Eğer robot, yer düzlemindeyse sadece çıkış düzlemleri kullanılabilir. Dolayısıyla kalan sınıflara ait olan düzlemler kullanılmak üzere ayrılacaktır. Diğer yandan, robot eğer düz rampa sınıfına ait bir düzlemdeyse iniş, çıkış ve düz rampa sınıflarına ait düzlemleri kullanabilmektedir. Bu kısıtlar altında kullanılacak düzlemler belirlendikten sonra düzlemlerin model parametreleri kullanılarak ağırlık merkezleri hesaplanmaktadır. Bu ağırlık merkezlerinin robota olan öklid uzaklıkları belirlenmektedir. Bir düzlemin robota olan uzaklığı *enBüyükMumkunUzaklık* olarak adlandırılan parametreden büyük ise bu düzlem de kullanılmak üzere ayrılmaktadır. Daha sonra düzlemler uzaklıklarına göre yakından uzağa sıralanmaktadır. Bu noktada, en yakın düzlem ele alınmakta ve bu düzlem için ara nokta oluşturulup oluşturulmadığına karar verilecektir. Eğer ara nokta

oluşturulamıyorsa, en yakın ikinci düzleme geçilerek süreç devam etmektedir. Bütün düzlemler kontrol edildikten sonra da ara nokta belirlenemiyorsa robot küresel hedefine gitmeye devam etmektedir.

İniş ve çıkış rampalarında ara nokta bulmak için eğimli rampa üzerinde rampanın başlangıç ve bitişinde iki adet limit noktası belirlenmektedir. Bu noktalar kullanılarak rampanın küresel yönelim açısı belirlenmektedir. Bu açığa karar verilirken afet ortamındaki eğimli rampaların ortamın referans koordinat eksenine göre dik veya yatay biçimlerde buldukları varsayılmıştır. Rampanın küresel yönelim açısı belirlendikten sonra, robotun küresel yönelim açısı ile arasındaki farka bakılmaktadır. Eğer bu fark $aciFarkiParam$ ile adlandırılan parametreden küçükse rampanın robotun seyrüseferi için uygun olduğuna karar verilmektedir. Rampanın küresel yönelim açısı ile robotun küresel yönelim açısı arasındaki fark $aciFarkiParam$ parametresinden büyükse, eğimli rampanın robotun hareket yönüne göre fazla dik olduğuna karar verilmektedir. Bu aşamada, bu rampaya giderken robotun önünde başka bir eğimli rampa varsa bu rampanın robotun seyrüseferi için uygun olmadığına karar verilmektedir. Seyrüsefer için uygun rampalar için rampanın başlangıç ve bitişinde iki adet limit noktası ara nokta olarak kullanılmaktadır.

Düz rampalarda ara nokta belirlemek için, rampaya ait nokta bulutunu çevreleyen kenarların bulunması hedeflenmektedir. Bu doğrultuda, dışbükey zarf (convex hull) algoritmasından faydalanılmıştır. Düz rampanın kenarlarını belirten nokta kümesi elde edildikten sonra bu noktaların ortalaması alınmakta ve ara nokta olarak kabul edilmektedir. Böylelikle, düz rampalar için, bir tane ara nokta belirlenmektedir.

Ara noktaların nasıl belirlendiğini anlattıktan sonra nasıl kullanılacağı konusuna değinmekte fayda görülmektedir. Bu noktada, robot küresel hedefi belli olduğunda yönünü bu hedefe döndürmektedir. Daha sonra yuvarlanma ve yunuslama açılarını kontrol etmektedir. Bu açılar kontrol edilmesinin sebebi robotun yer düzleminde ya da düz rampada olduğunda ara nokta belirlenmesi istenmiştir. Bu açılar belirlenen bir eşik değerinin altındaysa ara nokta belirleme aşamasına geçmektedir. Aksi takdirde, robot küresel hedefine gitmeye devam etmektedir. Ara nokta belirleme algoritmasından 3 tane sonuç elde edilebilmektedir. Bunlardan ilki, robotun görüş alanında rampa olmadığı durumdur. Açık ki bu durumda robot küresel hedefine gidecektir. İkinci durumla, uygun ara nokta bulunamadığında karşılaşılmaktadır. Bu durumda da robot küresel hedefine gitmeye devam etmektedir. Son olarak, uygun bir ara nokta belirlenmesi durumunda robot ara noktaya doğru harekete başlamaktadır. Bu esnada robotun konumu ile hem küresel hedef hem de ara nokta arasındaki uzaklık kontrol edilmektedir. Robotun küresel hedefine olan uzaklığı belirlenen eşik değerinin altına düştüğünde süreci tekrar başlatarak bir sonraki küresel hedef için yönünü ayarlamaktadır. Diğer yandan, eğer önce ara noktaya geldiyse

yuvarlanma ve yunuslama açılarını kontrol ederek sürece devam etmektedir. Bu noktada en yakındaki rampanın küresel hedefe uzaklığının, robotun küresel hedefe olan uzaklığıyla karşılaştırması yapılmaktadır. Eğer bu rampa robotun küresel hedefine robottan daha yakın değilse bu rampanın robotu küresel hedefinden uzaklaştıracağı düşünülerek başka bir rampa seçilmektedir. Böylece robotun sürekli olarak yerel noktalar seçerek hedefinden sapması önlenmektedir.

7.3 Sonuçlar

Bu alt bölümde tek ve çok robot ile afet ortamında keşifte üç farklı fiyatlandırma kriteri (en yakın sınır hücre, en çok kenar hücreye sahip sınır hücre ve market sınır hücre) ve anlamsal sınıflandırma bilgilerinin kullanıldığı durumlar incelenecektir.

7.3.1 Deneysel Kurulum

Önerilen yöntemin test edilmesi amacıyla ESOGÜ Laboratuvar binası Gazebo benzetim ortamında modellenmiştir. Tez çalışmasının kapsamı da keşif, haritalama, koordinasyon ve algılama problemlerine çözüm aradığı için hareket yeteneği çok gelişmiş robotlara ihtiyaç duyulmayan sarı test alanı özelliği taşıyan bir afet ortamı oluşturulmuştur. Bu kapsamda sadece eğimli ve düz rampalar kullanılmıştır. NIST'in belirlediği standartlara uyularak eğimli rampalar 15 derece olacak şekilde kullanılmıştır. Şekil 7.3'te test ortamı verilmiştir. Benzetimlerde iki adet lazer mesafe algılayıcısı ve lazer algılayıcıların üstünde bir derinlik ve renk algılayıcısı bulunan Pioneer P3-AT robotu kullanılmıştır. Önerilen yöntemler C++ programlama dili kullanılarak kodlanmıştır.

Robotun yakındaki rampaları kullanarak ara nokta belirlemesi için *enBuyukMumkunUzaklik* parametresi 3.5 metre olarak seçilmiştir. Robot ile rampa arasındaki küresel açı farkı *aciFarkiParam* 45 derece olarak belirlenmiştir. Robotun ara nokta bulma algoritmasını çalıştırmaya karar verirken kullandığı yuvarlanma ve yunuslama açılarının 1 dereceden küçük olması gerekmektedir. Son olarak, robot küresel hedefine 0.5 metre veya yerel hedefine 0.3 metre mesafeden daha yakınsa bu hedefe varıldığına karar verilmektedir.

7.3.2 Tek robotlu afet ortamı keşif sonuçları

Uygulamalar gerçekleştirilirken hedeflerin anlamsal sınıflandırma bilgisi ile belirlendiğinde robotun oda öncelikli rolde olduğu kabul edilmiştir. Bununla birlikte, Dijkstra yol planı ile üç farklı fiyatlandırma kriteri hedefleri belirlemek için kullanılmıştır.



Şekil 7.3: Afet sonrası test ortamı

Robot, diğer tek robotlu keşif yöntemleri ile aynı olan üç farklı yerden görevine başlamakta ve bütün ortamın keşfi tamamlanana kadar görevini sürdürmektedir.

Çizelge 7.1'de afet ortamında ve normal ortamda gerçekleştirilen keşif yaklaşımlarının sayısal sonuçları verilmiştir. Sonuçlar süre açısından incelendiğinde anlamsal sınıflandırma bilgileri kullanarak yapılan keşfin her iki ortamda da en iyi sonuçları ürettiği açıkça görülmektedir. Bununla birlikte, her iki ortamda da en kötü sonuçlar En çok kenar hücresi kriteri kullanıldığında elde edilmektedir.

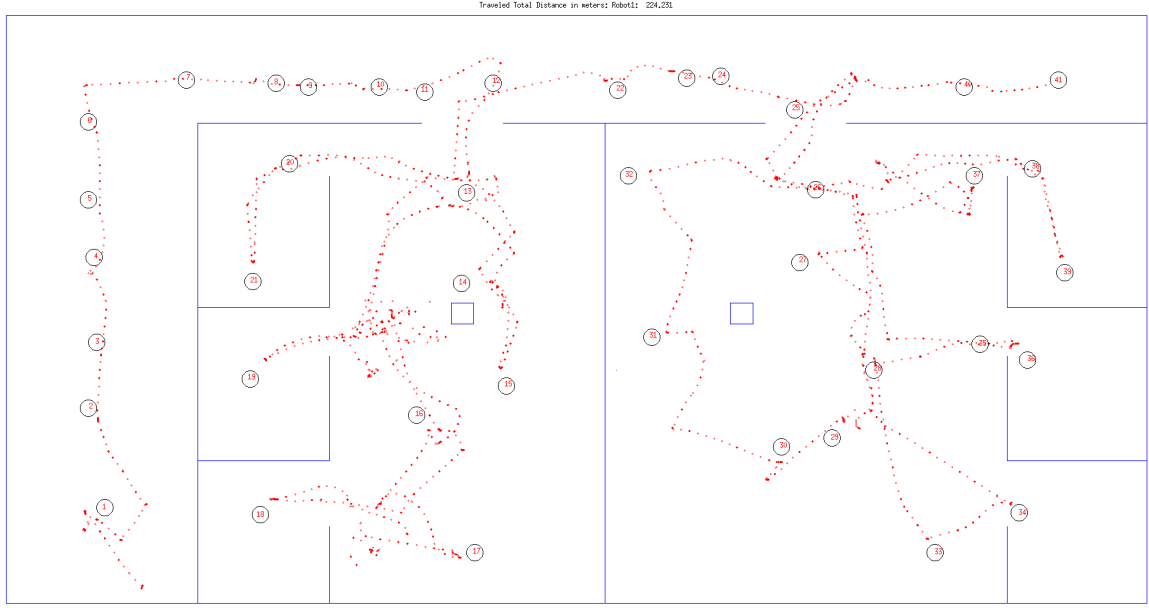
Afet ortamı ve normal ortam karşılaştırıldığında, afet ortamında keşif sürelerinin arttığı açık şekilde görülmektedir. Bunun sebeplerinden biri, robotun rampalar içerisinde seyrüsefer yapabilmesi için kullanılan ara nokta çıkarma işlemidir. Özellikle renk ve derinlik algılayıcısının gördüğü sahnede düz rampalar çoğunluktaysa ara nokta belirleme süresi uzamaktadır. Bir diğer keşif süresini uzatan faktör, seyrüseferdir. Robot normal ortamda rahatlıkla hareket edebilirken, afet ortamında seyrüsefer yaparken rampaların üstünden geçebilmek için rampanın uçlarına doğru hareket etmektedir. Bu durumda, hem zaman kaybetmekte hem de fazladan yol katetmektedir.

Çizelge 7.1: Afet Ortamında Tek Robot Sonuçları

Başlangıç Konumu	Hedef Seçme Kriteri	Afet Ortamı		Normal Ortam	
		Mesafe (m)	Süre (Dk)	Mesafe (m)	Süre (Dk)
2.5, 0.3, 1.57	En çok kenar hücresi	233.67	88	225.18	72
	En yakın	213.37	80	188.36	67
	Market	230.46	84	201.43	68
	R1 Oda	224.23	76	190.50	55
9, 7, 1.57	En çok kenar hücresi	394.03	107	276.19	82
	En yakın	268.91	88	220.90	71
	Market	257.18	94	197.68	66
	R1 Oda	258.51	86	214.57	62
24, 4, 1.57	En çok kenar hücresi	321.40	102	286.21	80
	En yakın	248.45	86	231.82	72
	Market	266.73	90	241.00	74
	R1 Oda	235.07	83	220.62	60

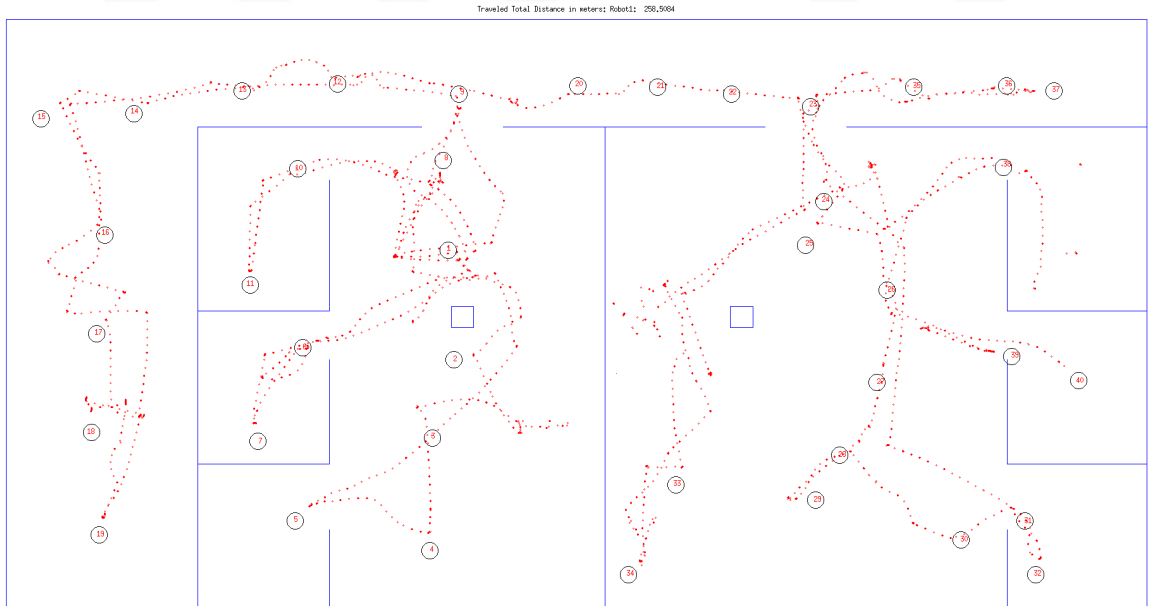
Şekil 7.4'te robotun 24,4,1.57 konumundan başladığı durumdaki anlamsal sınıflandırma bilgisi ile yaptığı hedef seçimleri görünmektedir. Robot oda öncelikli rolünde olduğu için başta ilk büyük odayı bütün küçük odaları ile keşfetmekte ve daha sonra ikinci büyük oda da aynı süreci işletip en son koridorun sonunu keşfetmektedir. Bununla birlikte, afet ortamında anlamsal sınıflandırma bilgisi ile yaptığı hedef seçimi her zaman normal ortamda olduğu gibi çalışmamaktadır. Şekil 7.5'te bu durum açıklanacaktır.

Şekil 7.5'te ise robotun 9,7,1.57 konumundan başladığı durumdaki anlamsal sınıflandırma bilgisi ile yaptığı hedef seçimleri görünmektedir. Robotun 8. hedefi büyük odanın kapısının önündedir. Sonraki hedef (9. hedef) koridordadır. Robot buraya çıkmakta ve buranın koridor olduğunu anlamakta ve tekrar büyük odaya (10. hedef) dönmektedir. Bununla birlikte, anlamsal sınıflandırma hataları sebebiyle aynı durum 34. hedeften sonra gerçekleşmemiştir. Robot koridorda olmasına rağmen 35. hedefe gitmiştir. Hedef noktaların anlamsal sınıflarını belirlemede hedefe en yakın düğümün anlamsal sınıfı kullanılmaktadır. Normalde robot hedefine vardığında en yakın düğümün anlamsal sınıfına hedef noktasının anlamsal sınıfı atanmakta ve hedef noktaların anlamsal sınıflarının bulunmasında hata yapılmamaktaydı. Bununla birlikte, afet ortamında robot bir düğüme vardığında eğer yuvarlanma ve yunuslama açıları 1 derecenin altında değilse bu düğüme anlamsal sınıflandırma yapılmamaktadır. Dolayısıyla, o düğümün anlamsal sınıfına karar verilirken komşu düğümlerin anlamsal sınıfları Markov modelle birlikte kullanılmaktadır. Komşu



Şekil 7.4: Afet ortamı anlamsal sınıflandırma bilgisi hedef seçme kriteri Konum (1)

düğümde de anlamsal sınıflandırma yapılmadıysa düğümlere hatalı anlamsal sınıflar atanabilmektedir. Bu durumda da keşif stratejisi istenilen şekilde işlememektedir.



Şekil 7.5: Afet ortamı anlamsal sınıflandırma bilgisi hedef seçme kriteri Konum (2)

7.3.3 Çok robotlu afet ortamı keşif sonuçları

Uygulamalar gerçekleştirilirken ilk olarak her iki robota da oda öncelikli robot rolü verilmiştir. Daha sonra birinci robotun koridor öncelikli ve ikinci robotun ise oda öncelikli robot rolünde olduğu testler gerçekleştirilmiştir. Son olarak, ikinci durumdaki roller değiştirilmiştir. Robotlar, diğer çok robotlu keşif yöntemleri ile aynı olan üç farklı yerden görevine başlamaktadır.

Çizelge 7.2: İki Robot Sonuçları

Fiyatlandırma Kriteri	Konum		Normal Ortam		Afet Ortamı			
			Süre (Dk)	Mesafe(m)	Süre (Dk)	Mesafe(m)		
	R1	R2	R1	R2	R1	R2		
En çok kenar hücresi			52	152.02	183.37	78	214.70	228.29
En yakın	2.5, 0.3, 1.57	24.8, 13.2, 3.14	34	101.37	119.07	59	142.36	159.13
Market			35	107.54	116.78	52	112.32	134.14
R1 Oda R2 Oda			31	109.54	102.75	47	106.99	137.56
R1 Koridor R2 Oda			42	120.35	138.43	59	145.90	165.96
R1 Oda R2 Koridor			42	130.74	160.71	61	148.16	172.89
En çok kenar hücresi					57	178.75	187.14	83
En yakın	16, 0.3, 1.57	24, 4, 1.57	37	118.96	124.33	70	177.78	178.39
Market			44	166.85	131.69	74	192.82	184.91
R1 Oda R2 Oda			34	119.56	131.21	66	152.4	153.90
R1 Koridor R2 Oda			40	138.75	144.96	61	141.98	126.16
R1 Oda R2 Koridor			40	138.87	115.53	58	147.88	168.12
En çok kenar hücresi					67	236.19	247.63	86
En yakın	9, 7, 1.57	11, 7, 1.57	38	121.46	114.69	64	153.90	169.59
Market			41	128.30	135.55	63	160.02	174.59
R1 Oda R2 Oda			52	169.07	167.20	78	173.11	180.33
R1 Koridor R2 Oda			38	126.12	136.77	68	214.41	200.42
R1 Oda R2 Koridor			40	131.44	137.44	69	173.64	171.07

Çizelge 7.2'de hedeflerin yukarıda anlatılan üç durum ile belirlendiği keşif yaklaşımı ve üç farklı fiyatlandırma kriteri kullanılarak gerçekleştirilen keşif yaklaşımlarının sayısal sonuçları verilmiştir. Sonuçlar, afet ortamı ve normal ortam olarak incelendiğinde tek robotta ortaya çıkan sonuçlar iki robotta da görülmektedir. Bununla birlikte, anlamsal sınıflandırma bilgisi kullanılmadığı durumda afet ve normal ortam arasındaki fark hem süre hem de mesafe olarak anlamsal sınıflandırma bilgisi kullanıldığı duruma göre daha fazladır.

8. SONUÇLAR VE ÖNERİLER

Bu çalışmada, bina içi ortamlarda, çok robotlu bir sistem için anlamsal tabanlı keşif algoritmaları geliştirmek amaçlanmıştır. Bina içi ortamlar, zeminin düz olduğu okul, hastane ve iş merkezi gibi yerler olabilir. Bununla birlikte, bu ortamlar, zehirli madde yayılımı, sel baskını ve yangın gibi afetlerden sonra rampa benzeri robotun seyrüseferini zorlaştıran engeller içeren; haberleşmenin kısıtlı olduğu ortamlara dönüşebilmektedir. Bu tez çalışması kapsamında önerilen keşif algoritmalarında her iki tip ortamın özellikleri de göz önünde bulundurulmuştur.

İlk olarak, geçmiş çalışmalarda sıklıkla kullanılan metrik harita ve sınır hücre yaklaşımı referans olması açısından gerçekleştirilmiştir. Bu yaklaşımda, hedefler belirlenirken üç farklı (en yakın sınır hücre, en çok kenar hücresine sahip sınır hücre ve market sınır hücre) fiyatlandırma kriteri kullanılmıştır. Hedef belirlendikten sonra yol planı A* algoritması ile yapılmakta ve VAH ile engellerden sakınarak hedefe gidilmektedir.

Bir sonraki aşamada, ortamın topolojik haritası, metrik harita kullanılarak adım adım oluşturulmuştur. Spektral kümeleme ile topolojik haritanın düğümleri belirlendikten sonra bu düğümler en küçük kapsayan ağaç algoritması ile birbirine bağlanmıştır. Bu aşamada uyarlamalı bir yaklaşımla ortamın büyük kısımları için daha seyrek düğüm oluşturulurken ortamın küçük kısımlarında düğümler sıklaştırılmıştır. Robot Dijkstra yol planlama yöntemini kullanmaktadır.

Robotlar arası koordinasyonu sağlamak ve görev paylaşımı yapmak amacıyla çok-öğeli ve tek-turlu müzayede yöntemini içeren market tabanlı bir yaklaşım kullanılmıştır. Bunun sebebi, geçmiş çalışmalarda da belirtildiği üzere zaman karmaşıklığı açısından en verimli küresel koordinasyon yaklaşımlarından biri olmasıdır. Bununla birlikte, sağladığı esnek yapı ile görevlerde ya da robot takımında yapılacak olan değişiklikler hızlıca sisteme eklenebilir. Son olarak, sadece teklif, ödül ya da görev gibi az sayıda ve boyutta mesaj kullandığından hem bant genişliği olarak hem de haberleşme yükü olarak uygun çözümler sunmaktadır. Haberleşme göz önüne alındığında ise robotların sadece belirli bir menzilde olduklarında haberleşebilmelerinin, diğer robotların görevleri için teklif verebilmeleri ve haritalarını diğer robotlardan gelen haritalar ile paylaşmalarının, uygun olacağı düşünülmüştür.

Anlamsal sınıfların belirlenmesinde lazer mesafe algılayıcı ile K-ortalama denetimli olmayan sınıflandırma yöntemi ile ÖVN denetimli sınıflandırma yöntemleri kullanılmıştır. Ayrıca, kapıların bulunması için kural tabanlı bir yöntem kullanılmıştır. Hedef seçimi yapılırken hedefin anlamsal sınıf bilgisinin kullanılması amaçlanmıştır. Bu sayede robotların daha iyi koordine olması ve keşif süresinin azalması sağlanmıştır. Bununla birlikte, keşif stratejisi robotlara da öncelikli ya da koridor öncelikli gibi roller atanarak belirlenebilmektedir.

Bu tez çalışması kapsamında, afet ortamı oluşturulurken sarı test alanının özelliklerini içeren bir model kullanılmıştır. Bu modelin kullanılmasının sebebi ise, geliştirilen keşif ve koordinasyon problemlerinin hareket yeteneği çok gelişmiş robotlara ihtiyaç duymadan test edilmesi için uygun olmasıdır. Tez kapsamında, özellikle keşif problemi ele alındığı için hareket yeteneği ile ilgili durumlar incelenmemiştir. Afet ortamında seyrüseferin yapılabilmesi için derinlik ve renk algılayıcı kullanılarak ara noktalar belirlenmiştir. Ara noktalar ve küresel hedef kullanılarak seyrüsefer gerçekleştirilmiştir.

Gelecek çalışmalarda ilk olarak, topolojik haritanın düğümlerinin bağlanması için döngüler içeren yöntemlerin (indirgenmiş çizge) kullanılması planlanmaktadır. Böylece robotun sürekli aynı düğümleri kullanarak hedeflere gitmesi önlenecek ve katedeceği mesafe ve süreninde azalması beklenmektedir. İkinci aşamada, sınır hücreleri kümeleyerek kombinasyonel müzayede yöntemi ile ya da gezgin satıcı problemini çözen sezgiseller kullanarak küresel bir koordinasyon sağlanabilir. Haberleşme modelinin gerçeğe uygunluğunu artırmak amacıyla yayılma haberleşme modeli olarak adlandırılan ve engellerin kalınlıklarını da haberleşme menziline hesaplarken hesaba katan model kullanılabilir. Afet ortamında, düğümlerin anlamsal sınıfları belirlenirken yuvarlanma ve yunuslama açıları 1 derecenin altında olması kısıtını kaldırmak için lazer algılayıcının xy düzlemine paralel olacak şekilde yön değiştirmesi sağlanabilir. Son olarak, tez kapsamında yapılan testlerde ortamın tamamının keşfedilmesi hedeflenmekteydi. Bu kriter, belirli bir süre ya da ortamı belirli bir oranda keşfetmek olarak değiştirilip testler tekrarlanabilir.

KAYNAKLAR DİZİNİ

- Ahuja, Ravindra K., Thomas L. Magnanti ve James B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0-13-617549-X.
- Akçakoca, M., İ. Kılıç, U. Yayan, B. Akar ve A. Yazıcı (2014). "Akıllı Tekerlekli Sandalye için Algılayıcı Tabanlı Kontrol Gerçeklemesi". İn: *Otomatik Kontrol Ulusal Toplantısı - TOK'14*, pp. 14–19.
- Althaus, Philipp ve Henrik I. Christensen (2003). "Behavior coordination in structured environments". İn: *Advanced Robotics* 17.7, pp. 657–674.
- Amigoni, F. ve A. Gallo (2005). "A Multi-Objective Exploration Strategy for Mobile Robots". İn: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3850–3855.
- Anguelov, D., D. Koller, E. Parker ve S. Thrun (2004). "Detecting and modeling doors with mobile robots". İn: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. Vol. 4, 3777–3784 Vol.4.
- Arkin, R.C. ve J. Diaz (2002). "Line-of-sight constrained exploration for reactive multiagent robotic teams". İn: *Advanced Motion Control, 2002. 7th International Workshop on*, pp. 455–461.
- Aurenhammer, Franz (1991). "Voronoi diagrams—a survey of a fundamental geometric data structure". İn: *ACM Computing Surveys (CSUR)* 23.3, pp. 345–405.

- Aydemir, Alper, Moritz Göbelbecker, Andrzej Pronobis, Kristoffer Sjö ve Patric Jensfelt (2011). "Plan-based Object Search and Exploration using Semantic Spatial Knowledge in the Real World." In: *ECMR*, pp. 13–18.
- Balaguer, Benjamin, Stephen Balakirsky, Stefano Carpin ve Arnoud Visser (2009). "Evaluating maps produced by urban search and rescue robots: lessons learned from RoboCup". In: *Auton. Robots* 27.4, pp. 449–464.
- Balakirsky, Stephen, Chris Scrapper ve Stefano Carpin (2007a). "The evolution of performance metrics in the RoboCup Rescue Virtual Robot Competition". In: *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*. PerMIS '07, pp. 91–96.
- Balakirsky, Stephen, Stefano Carpin, Alexander Kleiner, Michael Lewis, Arnoud Visser, Jijun Wang ve Vittorio Amos Ziparo (2007b). "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup rescue". In: *J. Field Robot.* 24.11, pp. 943–967.
- Balakirsky, Stephen, Stefano Carpin ve Arnoud Visser (2009). "Evaluating the RoboCup 2009 Virtual Robot Rescue Competition". In: *Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems*. PerMIS '09, pp. 109–114.
- Barber, R., M. Mata, M.J.L. Boada, J.M. Armingol ve M.A. Salichs (2002). "A perception system based on laser information for mobile robot topologic navigation". In: *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*. Vol. 4, 2779–2784 vol.4.
- Basilico, Nicola ve Francesco Amigoni (2011). "Exploration strategies based on multi-criteria decision making for searching environments in rescue operations". In: *Autonomous Robots* 31.4, pp. 401–417.

- Baum L. E.; Petrie, T. (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". In: *The Annals of Mathematical Statistics* 37.6, pp. 1554–1563.
- Bautin, Antoine, Olivier Simonin ve François Charpillet (2012). "Minpos: A novel frontier allocation algorithm for multi-robot exploration". In: *International Conference on Intelligent Robotics and Applications*, pp. 496–508.
- Berhault, M., H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin ve A. Kleywegt (2003). "Robot exploration with combinatorial auctions". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 2, 1957–1962 vol.2.
- Blanco, Jose-Luis, Javier Gonzalez ve Juan-Antonio Fernandez-Madrigal (2006). "Consistent observation grouping for generating metric-topological maps that improves robot localization". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. Pp. 818–823.
- Borenstein, J. ve Y. Koren (1991). "The vector field histogram-fast obstacle avoidance for mobile robots". In: *Robotics and Automation, IEEE Transactions on* 7.3, pp. 278–288.
- Bresenham (2016). http://en.wikipedia.org/wiki/Bresenham's_line_algorithm. URL: http://en.wikipedia.org/wiki/Bresenham's_line_algorithm.
- Bresenham, J. E. (1965). "Algorithm for computer control of a digital plotter". In: *IBM Systems Journal* 4.1, pp. 25–30. ISSN: 0018-8670. DOI: 10.1147/sj.41.0025.
- Brunskill, Emma, Thomas Kollar ve Nicholas Roy (2007). "Topological mapping using spectral clustering and classification". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3491–3496.

- Burgard, W., M. Moors, D. Fox, R. Simmons ve S. Thrun (2000). "Collaborative multi-robot exploration". In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. Vol. 1, 476–481 vol.1.
- Burgard, W., M. Moors, C. Stachniss ve F.E. Schneider (2005). "Coordinated multi-robot exploration". In: *Robotics, IEEE Transactions on* 21.3, pp. 376–386.
- Buschka, P. ve A. Saffiotti (2002). "A virtual sensor for room detection". In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 1, 637–642 vol.1.
- Calisi, Daniele, Alessandro Farinelli, Luca Iocchi ve Daniele Nardi (2007). "Multi-objective exploration and search for autonomous rescue robots". In: *Journal of Field Robotics* 24.8, pp. 763–778.
- Carpin, Stefano (2008). "Fast and accurate map merging for multi-robot systems". In: *Autonomous Robots* 25.3, pp. 305–316.
- Carpin, Stefano, Mike Lewis, Jijun Wang, Steve Balakirsky ve Chris Scrapper (2007). "RoboCup 2006: Robot Soccer World Cup X". In: ed. by Gerhard Lakemeyer, Elizabeth Sklar, Domenico G. Sorrenti ve Tomoichi Takahashi. Chap. Bridging the Gap Between Simulation and Reality in Urban Search and Rescue, pp. 1–12.
- Carvalho, Fabricio F, Rodolfo C Cavalcante, Marcos AM Vieira, Luiz Chaimowicz ve Mario FM Campos (2013). "A multi-robot exploration approach based on distributed graph coloring". In: *Robotics Symposium and Competition (LARS/LARC), 2013 Latin American*, pp. 142–147.
- Casper, Jennifer L, Mark Micire ve Robin R Murphy (2000). "Issues in intelligent robots for search and rescue". In: *AeroSense 2000*, pp. 292–302.

- Chand, Praneel ve Dale A. Carnegie (2013). "Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots". In: *Robotics and Autonomous Systems* 61.6, pp. 565–579.
- Cheong, Howon, S. Park ve S. K. Park (2008). "Topological map building and exploration based on concave nodes". In: *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pp. 1115–1120.
- Choi, Jinwoo, Minyong Choi, Sang Yep Nam ve Wan Kyun Chung (2011). "Autonomous topological modeling of a home environment and topological localization using a sonar grid map". In: *Autonomous Robots* 30.4, pp. 351–368.
- Choset, Howie (1996). "Sensor based motion planning: The hierarchical generalized Voronoi graph". PhD thesis.
- Cipolleschi, Riccardo, Michele Giusto, Alberto Quattrini Li ve Francesco Amigoni (2013). "Semantically-informed coordinated multirobot exploration of relevant areas in search and rescue settings". In: *Mobile Robots (ECMR), 2013 European Conference on*, pp. 216–221.
- Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering. ISBN: 9780201543612. URL: <https://books.google.com.tr/books?id=MqMeAQAAIAAJ>.
- Cut, Minimum (2016). <https://en.wikipedia.org/wiki/Minimumcut>. URL: <https://en.wikipedia.org/wiki/Minimumcut>.
- DDA (2016). [http://en.wikipedia.org/wiki/DigitalDifferentialAnalyzer\(graphicsalgorithm\)](http://en.wikipedia.org/wiki/DigitalDifferentialAnalyzer(graphicsalgorithm)). URL: [http://en.wikipedia.org/wiki/Digital_Differential_Analyzer_\(graphics_algorithm\)](http://en.wikipedia.org/wiki/Digital_Differential_Analyzer_(graphics_algorithm)).

- De Hoog, Julian, Stephen Cameron ve Arnoud Visser (2009). "Role-based autonomous multi-robot exploration". In: *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD'09. Computation World*: pp. 482–487.
- Diaz, Jonathan F, Alexander Stoytchev ve Ronald C Arkin (2001). *Exploring unknown structured environments*, pp. 145–149.
- Dijkstra, Edsger W (1959). "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1, pp. 269–271.
- Doniec, Arnaud, Noury Bouraqadi, Michael Defoort, Serge Stinckwich, vd. (2009). "Distributed constraint reasoning applied to multi-robot exploration". In: *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pp. 159–166.
- Dudek, G., M. Jenkin, E. Milios ve D. Wilkes (1991). "Robotic exploration as graph construction". In: *Robotics and Automation, IEEE Transactions on* 7.6, pp. 859–865.
- Eich, Markus, Malgorzata Dabrowska ve Frank Kirchner (2010). "Semantic labeling: Classification of 3d entities based on spatial feature descriptors". In: *IEEE International Conference on Robotics and Automation (ICRA2010) in Anchorage, Alaska*.
- ElKaissi, M., M. Elgamel, M. Bayoumi ve B. Zavidovique (2007). "SEDLRF: A New Door Detection System for Topological Maps". In: *Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on*, pp. 75–80.
- Faigl, Jan ve Miroslav Kulich (2013). "On determination of goal candidates in frontier-based multi-robot exploration". In: *Mobile Robots (ECMR), 2013 European Conference on*, pp. 210–215.

Fink, Jonathan, Tom Collins, Vijay Kumar, Yasamin Mostofi, John Baras ve Brian Sadler (2009). "A simulation environment for modeling and development of algorithms for ensembles of mobile microsystems". In: *SPIE Defense, Security, and Sensing*, 73180N–73180N.

Fischler, Martin A ve Robert C Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6, pp. 381–395.

Fox, D., W. Burgard ve S. Thrun (1997). "The dynamic window approach to collision avoidance". In: *Robotics Automation Magazine, IEEE* 4.1, pp. 23–33.

Fox, D., J. Ko, K. Konolige, B. Limketkai, D. Schulz ve B. Stewart (2006). "Distributed Multirobot Exploration and Mapping". In: *Proceedings of the IEEE* 94.7, pp. 1325–1339.

Franchi, Antonio, Luigi Freda, Giuseppe Oriolo ve Marilena Vendittelli (2009). "The sensor-based random graph method for cooperative robot exploration". In: *IEEE/ASME Transactions on Mechatronics* 14.2, pp. 163–175.

Galindo, C., A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal ve J. Gonzalez (2005). "Multi-hierarchical semantic maps for mobile robotics". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2278–2283.

Galindo, Cipriano, Juan-Antonio Fernández-Madrigal, Javier González ve Alessandro Saffiotti (2008). "Robot task planning using semantic maps". In: *Robotics and autonomous systems* 56.11, pp. 955–966.

Gazebo (2016). <http://gazebosim.org/>. URL: <http://gazebosim.org/>.

- Goerke, Nils ve Sven Braun (2009). "Building semantic annotated maps by mobile robots". In: *Proceedings of the Conference Towards Autonomous Robotic Systems, Londonderry, UK*, pp. 149–156.
- Gonzalez-Banos, Hector H. ve Jean-Claude Latombe (2002). "Navigation Strategies for Exploring Indoor Environments". In: *The International Journal of Robotics Research* 21.10-11, pp. 829–848.
- Grabowski, R., P. Khosla ve H. Choset (2003). "Autonomous exploration via regions of interest". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 2, 1691–1696 vol.2.
- Hart, Peter E, Nils J Nilsson ve Bertram Raphael (1968). "A formal basis for the heuristic determination of minimum cost paths". In: *Systems Science and Cybernetics, IEEE Transactions on* 4.2, pp. 100–107.
- Haumann, A Dominik, Kim D Listmann ve Volker Willert (2010). "Discoverage: A new paradigm for multi-robot exploration". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 929–934.
- Hawley, John ve Zack Butler (2013). "Hierarchical distributed task allocation for multi-robot exploration". In: *Distributed autonomous robotic systems*, pp. 445–458.
- Hensler, Jens, Michael Blaich ve Oliver Bittel (2009). "Real-time door detection based on adaboost learning algorithm". In: *International Conference on Research and Education in Robotics*, pp. 61–73.
- Hoog, Julian de, Stephen Cameron ve Arnoud Visser (2010a). "Dynamic Team Hierarchies in Communication-Limited Multi-Robot Exploration". In: *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*.

- Hoog, Julian de, Stephen Cameron ve Arnoud Visser (2010b). "Selection of Rendezvous Points for Multi-Robot Exploration in Dynamic Environments". In: *Workshop on Agents in Realtime and Dynamic Environments, International Conference on Autonomous Agents and Multi-Agent Systems(AAMAS)*.
- Howard, A., M.J. Mataric ve G.S. Sukhatme (2002). "An incremental deployment algorithm for mobile robot teams". In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 3, 2849–2854 vol.3.
- Howard, Andrew, Lynne E Parker ve Gaurav S Sukhatme (2006). "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection". In: *The International Journal of Robotics Research* 25.5-6, pp. 431–447.
- Huang, Wesley H ve Kristopher R Beevers (2005). "Topological map merging". In: *The International Journal of Robotics Research* 24.8, pp. 601–613.
- Jacoff, A., E. Messina, B. A. Weiss, S. Tadokoro ve Y. Nakagawa (2003). "Test arenas and performance metrics for urban search and rescue robots". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 4, 3396–3403 vol.3.
- Jensen, Elizabeth A, Ernesto Nunes ve Maria Gini (2014). "Communication-restricted exploration for robot teams". In: *AAAI Workshops*.
- Jia, MengLei, GuangMing Zhou ve ZongHai Chen (2004). "An efficient strategy integrating grid and topological information for robot exploration". In: *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*. Vol. 2, 667–672 vol.2.
- Jia, S., X. Zhao, Y. Li ve K. Wang (2013). "A hierarchical approach to region division for mobile robot based on spectral cluster". In: *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1371–1376.

- Joo, K., T. K. Lee, S. Baek ve S. Y. Oh (2010). "Generating topological map from occupancy grid-map using virtual door detection". In: *IEEE Congress on Evolutionary Computation*, pp. 1–6.
- Juliá, Miguel, Oscar Reinoso, Arturo Gil, Mónica Ballesta ve Luis Payá (2010). "A hybrid solution to the multi-robot integrated exploration problem". In: *Engineering Applications of Artificial Intelligence* 23.4, pp. 473–486.
- Kaleci, B., Ç M. Şenler, H. Dutagacı ve O. Parlaktuna (2015a). "A probabilistic approach for semantic classification using laser range data in indoor environments". In: *Advanced Robotics (ICAR), 2015 International Conference on*, pp. 375–381.
- Kaleci, B., Ç M. Senler, O. Parlaktuna ve U. Gürel (2015b). "Constructing Topological Map from Metric Map Using Spectral Clustering". In: *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pp. 139–145.
- Kaleci, B., Ç M. Senler, H. Dutagaci ve O. Parlaktuna (2015c). "Rule-Based Door Detection Using Laser Range Data in Indoor Environments". In: *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pp. 510–517.
- Kaleci, Burak (2010). "Heterojen Gezgin Robot Sistemleri İçin Yeni Bir Market Tabanlı Görev Paylaşırma Yaklaşımı". In: *Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi*.
- Khatib, O. (1985). "Real-time obstacle avoidance for manipulators and mobile robots". In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. Vol. 2, pp. 500–505.
- Kitano, H. ve S. Tadokoro (2001). "RoboCup Rescue: a grand challenge for multiagent and intelligent systems". In: *AI Magazine* 22.1, pp. 39–52.

- Ko, J., B. Stewart, D. Fox, K. Konolige ve B. Limketkai (2003). "A practical, decision-theoretic approach to multi-robot mapping and exploration". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 4, 3232–3238 vol.3.
- Koenig, Nathan ve Andrew Howard (2004). "Design and use paradigms for gazebo, an open-source multi-robot simulator". In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 3, pp. 2149–2154.
- Konolige, K., E. Marder-Eppstein ve B. Marthi (2011). "Navigation in hybrid metric-topological maps". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3041–3047.
- Korsah, G. Ayorkor, Anthony (Tony) Stentz ve M Bernardine Dias (2007). *The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs*. Tech. rep. CMU-RI-TR-07-27. Pittsburgh, PA: Robotics Institute.
- Kostavelis, Ioannis ve Antonios Gasteratos (2015). "Semantic mapping for mobile robotics tasks: A survey". In: *Robotics and Autonomous Systems* 66, pp. 86–103.
- Krishnan, Aravindhan K ve K Madhava Krishna (2010). "A visual exploration algorithm using semantic cues that constructs image based hybrid maps". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1316–1321.
- Kuhn, Harold W (1955). "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2, pp. 83–97.
- Li, Xiuzhi ve Huan Qiu (2015). "An effective laser-based approach to build topological map of unknown environment". In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 200–205.

- Liu, Ming, Francis Colas ve Roland Siegwart (2011). "Regional topological segmentation based on mutual information graphs". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3269–3274.
- Liu, Ming, Luc Oth, Francis Colas ve R Siegwart (2014). "Incremental topological segmentation for semi-structured environments". In: *Autonomous Robots* 37.3.
- Liu, Yugang ve Goldie Nejat (2013). "Robotic Urban Search and Rescue: A Survey from the Control Perspective". In: *Journal of Intelligent & Robotic Systems* 72.2, pp. 147–165.
- Luperto, M., L. D'Emilio ve F. Amigoni (2015). "A generative spectral model for semantic mapping of buildings". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 4451–4458.
- MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1. Statistics, pp. 281–297.
- Makarenko, A.A., S.B. Williams, F. Bourgault ve H.F. Durrant-Whyte (2002). "An experiment in integrated exploration". In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 1, 534–539 vol.1.
- Marjovi, Ali, João Gonçalo Nunes, Lino Marques ve Aníbal de Almeida (2009). "Multi-robot exploration and fire searching". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1929–1934.
- Martínez Mozos, Óscar, Axel Rottmann, Rudolph Triebel, Patric Jensfelt ve Wolfram Burgard (2007a). "Supervised semantic labeling of places using information extracted from sensor data". In: *Robotics and Autonomous Systems* 55, pp. 391–402.

- Martínez Mozos, Óscar, Axel Rottmann, Rudolph Triebel, Patric Jensfelt ve Wolfram Burgard (2007b). "Supervised semantic labeling of places using information extracted from sensor data". In: *Robotics and Autonomous Systems* 55.5, pp. 391–402.
- Matignon, Laëtitia, Laurent Jeanpierre ve Abdel-Ilhah Mouaddib (2012). "Distributed value functions for multi-robot exploration". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1544–1550.
- Meghjani, Malika ve Gregory Dudek (2011). "Combining multi-robot exploration and rendezvous". In: *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pp. 80–85.
- Monasterio, Iñaki, Elena Lazkano, Iñaki Rañó ve Basilo Sierra (2002). "Learning to traverse doors using visual information". In: *Mathematics and Computers in Simulation* 60.3–5, pp. 347–356.
- Moorehead, S.J., R. Simmons ve W.L. Whittaker (2001). "Autonomous exploration using multiple sources of information". In: *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. Vol. 3, 3098–3103 vol.3.
- Moravec, H.P. ve A. Elfes (1985). "High resolution maps from wide angle sonar". In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. Vol. 2, pp. 116–121.
- Mosteo, A.R., L. Montano ve M.G. Lagoudakis (2008). "Multi-robot routing under limited communication range". In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1531–1536.
- Mozos, O. M. ve W. Burgard (2006). "Supervised Learning of Topological Maps using Semantic Information Extracted from Range Data". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2772–2777.

- Mozos, O. M., C. Stachniss ve W. Burgard (2005). "Supervised Learning of Places from Range Data using AdaBoost". In: *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1730–1735.
- Mozos, Oscar Martinez, Hitoshi Mizutani, Hojung Jung, Ryo Kurazume ve Tsutomu Hasegawa (2013). "Categorization of indoor places by combining local binary pattern histograms of range and reflectance data from laser range finders". In: *Advanced Robotics* 27.18, pp. 1455–1464.
- Munkres, James (1957). "Algorithms for the assignment and transportation problems". In: *Journal of the Society for Industrial & Applied Mathematics* 5.1, pp. 32–38.
- NISTTestAlanları (2016). <http://robotarenas.nist.gov/2007/refdatasets.htm>. URL: <http://robotarenas.nist.gov/2007/refdatasets.htm>.
- Nieto-Granda, Carlos, John G Rogers, Alexander JB Trevor ve Henrik I Christensen (2010). "Semantic map partitioning in indoor environments using regional analysis". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1451–1456.
- Nieto-Granda, Carlos, John G Rogers ve Henrik I Christensen (2014). "Coordination strategies for multi-robot exploration and mapping". In: *The International Journal of Robotics Research*.
- Nodes (2016). <http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>. URL: <http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>.
- Nüchter, Andreas ve Joachim Hertzberg (2008). "Towards semantic maps for mobile robots". In: *Robotics and Autonomous Systems* 56.11, pp. 915–926.

- Özkucur, N Ergin ve H Levent Akın (2009). "Cooperative multi-robot map merging using fast-SLAM". In: *Robot Soccer World Cup*, pp. 449–460.
- Pei, Yuanteng, Matt W Mutka ve Ning Xi (2010). "Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5460–5465.
- Poncela, A., E.J. Perez, A. Bandera, C. Urdiales ve F. Sandoval (2002). "Efficient integration of metric and topological maps for directed exploration of unknown environments". In: *Robotics and Autonomous Systems* 41.1, pp. 21 –39.
- Powers, Matthew, Tucker Balch, vd. (2004). "Value-based communication preservation for mobile robots". In: *in 7th International Symposium on Distributed Autonomous Robotic Systems*.
- Pronobis, A. ve P. Jensfelt (2012). "Large-scale semantic mapping and reasoning with heterogeneous modalities". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3515–3522.
- Pronobis, A., B. Caputo, P. Jensfelt ve H. I. Christensen (2006). "A Discriminative Approach to Robust Visual Place Recognition". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3829–3836.
- Pronobis, Andrzej, O Martínez Mozos, Barbara Caputo ve Patric Jensfelt (2009). "Multi-modal semantic place classification". In: *The International Journal of Robotics Research*.
- Puig, D., M.A. Garcia ve L. Wu (2011). "A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation". In: *Robotics and Autonomous Systems* 59.9, pp. 635 –653.

Quattrini Li, Alberto, Riccardo Cipolleschi, Michele Giusto ve Francesco Amigoni (2016). "A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings". In: *Autonomous Robots* 40.4, pp. 581–597.

ROS (2016). <http://www.ros.org/about-ros/>. URL: <http://www.ros.org/about-ros/>.

Rekleitis, I. M., V. Dujmovic ve G. Dudek (1999). "Efficient topological exploration". In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. Vol. 1, 676–681 vol.1.

Rekleitis, I., V. Lee-Shue, Ai Peng New ve H. Choset (2004). "Limited communication, multi-robot team based coverage". In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. Vol. 4, 3462–3468 Vol.4.

Rekleitis, I.M., G. Dudek ve E.E. Milios (2000). "Multi-robot collaboration for robust exploration". In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. Vol. 4, 3164–3169 vol.4.

RoboCupRescue (2016). <http://www.robocuprescue.org/>. URL: <http://www.robocuprescue.org/>.

RoboCupRescueRobotLeague (2016). <http://www.robocuprescue.org/rescuerobots.html/>. URL: <http://www.robocuprescue.org/rescuerobots.html/>.

RoboCupRescueSimulationLeague (2016). <http://www.robocuprescue.org/agentsim.html>. URL: <http://www.robocuprescue.org/agentsim.html>.

Rooker, Martijn N. ve Andreas Birk (2007). "Multi-robot exploration under the constraints of wireless networking". In: *Control Engineering Practice* 15.4, pp. 435–445.

- Roy, Nicholas ve Gregory Dudek (2001). "Collaborative Robot Exploration and Rendezvous: Algorithms, Performance Bounds and Observations". In: *Autonomous Robots* 11.2, pp. 117–136.
- Santos, Filipe Neves, António Paulo Moreira ve Paulo Cerqueira Costa (2013). "Towards extraction of topological maps from 2D and 3D occupancy grids". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 307–318.
- Sariel, Sanem ve Tucker R Balch (2006). "Efficient Bids on Task Allocation for Multi-Robot Exploration." In: *FLAIRS Conference*, pp. 116–121.
- Sheng, Weihua, Qingyan Yang, Jindong Tan ve Ning Xi (2006). "Distributed multi-robot coordination in area exploration". In: *Robotics and Autonomous Systems* 54.12, pp. 945–955.
- Shi, Jianbo ve J. Malik (1997). "Normalized cuts and image segmentation". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 731–737.
- Shi, Jianbo ve Jitendra Malik (2000). "Normalized Cuts and Image Segmentation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8, pp. 888–905.
- Shi, Lei ve Sarath Kodagoda (2013). "Towards generalization of semi-supervised place classification over generalized Voronoi graph". In: *Robotics and Autonomous Systems* 61.8, pp. 785–796.
- Shi, Lei, Sarath Kodagoda ve Gamini Dissanayake (2010). "Laser range data based semantic labeling of places". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5941–5946.

- Simmons, R., D. Apfelbaum, W. Burgard, M. Fox D. an Moors, S. Thrun ve H. Younes (2000). "Coordination for Multi-Robot Exploration and Mapping". In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. AAAI.
- Singh, K. ve K. Fujimura (1993). "Map making by cooperating mobile robots". In: *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, 254–259 vol.2.
- Soares, Symone G ve Rui Araújo (2014). "Semantic place labeling using a probabilistic decision list of AdaBoost classifiers". In: *Int. J. Comput. Inf. Syst. ind. Manag. Appl* 6, pp. 548–559.
- Solanas, Agusti ve Miguel Angel Garcia (2004). "Coordinated multi-robot exploration through unsupervised clustering of unknown space". In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 1, pp. 717–721.
- Sousa, P., R. Araujo ve U. Nunes (2007). "Real-Time Labeling of Places using Support Vector Machines". In: *2007 IEEE International Symposium on Industrial Electronics*, pp. 2022–2027.
- Stachniss, C., O.M. Mozos ve W. Burgard (2006). "Speeding-up multi-robot exploration by considering semantic place information". In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1692–1697.
- Stachniss, Cyrill ve Wolfram Burgard (2003). "Exploring unknown environments with mobile robots using coverage maps". In: *International Joint Conference on Artificial Intelligence*. Vol. 18, pp. 1127–1134.
- Stachniss, Cyrill, Óscar Martínez Mozos ve Wolfram Burgard (2008). "Efficient exploration of unknown indoor environments using a team of mobile robots". In: *Annals of Mathematics and Artificial Intelligence* 52.2-4, pp. 205–227.

- Stentz, Anthony ve M Bernardine Dias (1999). *A free market architecture for coordinating multiple robots*. Tech. rep.
- Stroupe, A.W., R. Ravichandran ve T. Balch (2004). "Value-based action selection for exploration and dynamic target observation with robot teams". In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. Vol. 4, 4190–4197 Vol.4.
- Suarez, Jesus ve Robin R Murphy (2012). "Using the kinect for search and rescue robotics". In: *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pp. 1–2.
- Taillandier, Patrick ve Serge Stinckwich (2011). "Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots". In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 321–326.
- Tapus, A. ve R. Siegwart (2006). "A cognitive modeling of space using fingerprints of places for mobile robot navigation". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. Pp. 1188–1193.
- Thrun, Sebastian (1998). "Learning metric-topological maps for indoor mobile robot navigation". In: *Artificial Intelligence* 99.1, pp. 21–71.
- Topal, Sebahattin, Dsmet Erkmen ve Aydan M Erkmen (2010). "A novel map merging methodology for multi-robot systems". In:
- Topics (2016). <http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics>. URL: <http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics>.

Tuna, Gurkan, Kayhan Gulez ve V. Cagri Gungor (2013). "The effects of exploration strategies and communication models on the performance of cooperative exploration". In: *Ad Hoc Networks* 11.7, pp. 1931–1941.

Ulrich, I. ve J. Borenstein (1998). "VFH+: reliable obstacle avoidance for fast mobile robots". In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 2, 1572–1577 vol.2.

Usarsim (2016).
<http://sourceforge.net/apps/mediawiki/usarsim/index.php?title=Introduction>. URL:
<http://sourceforge.net/apps/mediawiki/usarsim/index.php?title=Introduction>.

Vasudevan, Shrihari, Stefan Gächter, Viet Nguyen ve Roland Siegwart (2007). "Cognitive maps for mobile robots—an object based approach". In: *Robotics and Autonomous Systems* 55.5, pp. 359–371.

Vázquez-Martín, Ricardo, Pedro Núñez, Antonio Bandera ve Francisco Sandoval (2009). "Spectral clustering for feature-based metric maps partitioning in a hybrid mapping framework". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 4175–4181.

Vazquez, J. ve C. Malcolm (2004). "Distributed multirobot exploration maintaining a mobile network". In: *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*. Vol. 3, 113–118 Vol.3.

Vincent, Regis, Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Benoit Morisset, Charles Ortiz, Dirk Schulz ve Benjamin Stewart (2008). "Distributed multirobot exploration, mapping, and task allocation". In: *Annals of Mathematics and Artificial Intelligence* 52.2-4, pp. 229–255.

- VirtualRobotCompetition (2016). http://wiki.robocup.org/wiki/Rescue_Simulation_League.
URL: http://wiki.robocup.org/wiki/Rescue_Simulation_League.
- Visser, A. ve B.A. Slamet (2008a). "Including communication success in the estimation of information gain for multi-robot exploration". In: *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pp. 680–687.
- Visser, Arnoud ve BayuA. Slamet (2008b). "Balancing the Information Gain Against the Movement Cost for Multi-robot Frontier Exploration". In: *European Robotics Symposium 2008*. Ed. by Herman Bruyninckx, Libor Přebil ve Miroslav Kulich. Vol. 44. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, pp. 43–52.
- Visser, Arnoud, J Hoog, Adrian Jiménez-González, JR Martínez-de Dios, vd. (2013). "Discussion of multi-robot exploration in communication-limited environments". In:
- Viswanathan, Pooja, David Meger, Tristram Southey, James J Little ve Alan K Mackworth (2009). "Automated spatial-semantic modeling with applications to place labeling and informed search". In: *Computer and Robot Vision, 2009. CRV'09. Canadian Conference on*, pp. 284–291.
- Von Luxburg, Ulrike (2007). "A tutorial on spectral clustering". In: *Statistics and computing* 17.4, pp. 395–416.
- Wang, Yiheng, Alei Liang ve Haibing Guan (2011). "Frontier-based multi-robot map exploration using particle swarm optimization". In: *Swarm Intelligence (SIS), 2011 IEEE Symposium on*, pp. 1–6.
- Wellman, Briana Lowe, Shameka Dawson, Julian de Hoog ve Monica Anderson (2011). "Using rendezvous to overcome communication limitations in multirobot exploration". In: *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 2401–2406.

- Werner, Felix, Joaquin Sitte ve Frederic Maire (2012). "Topological map induction using neighbourhood information of places". In: *Autonomous Robots* 32.4, pp. 405–418.
- Wirth, S. ve J. Pellenz (2007). "Exploration Transform: A stable exploring algorithm for robots in rescue environments". In: *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 1–5.
- Wolsey, L.A. (1998). *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization.
- Wu, Jianxin, Henrik I Christensen ve James M Rehg (2009). "Visual place categorization: Problem, dataset, and algorithm". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4763–4770.
- Wu, Ling, Miguel Ángel García García, Domenec Puig Valls ve Albert Solé Ribalta (2007). "Voronoi-based space partitioning for coordinated multi-robot exploration". In:
- Wu, Ling, Domenec Puig Valls ve Miguel Ángel García García (2010). "Balanced multi-robot exploration through a global optimization strategy". In: *Journal of Physical Agents* 4.1, pp. 35–44.
- Wurm, K.M., C. Stachniss ve W. Burgard (2008). "Coordinated multi-robot exploration using a segmentation of the environment". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1160–1165.
- Yamauchi, Brian (1997). "A frontier-based approach for autonomous exploration". In: *In Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, pp. 146–151.

- Yamauchi, Brian (1998). "Frontier-based exploration using multiple robots". In: *Proceedings of the second international conference on Autonomous agents*. AGENTS '98. Minneapolis, Minnesota, USA, pp. 47–53.
- Yang, Xiaodong ve Yingli Tian (2010). "Robust door detection in unfamiliar environments by combining edge and corner features". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 57–64.
- Zelinsky, A. (1992). "A mobile robot exploration algorithm". In: *Robotics and Automation, IEEE Transactions on* 8.6, pp. 707–717.
- Zender, Hendrik, O Martínez Mozos, Patric Jensfelt, G-JM Kruijff ve Wolfram Burgard (2008). "Conceptual spatial representations for indoor mobile robots". In: *Robotics and Autonomous Systems* 56.6, pp. 493–502.
- Ziparo, Vittorio A, Alexander Kleiner, Bernhard Nebel ve Daniele Nardi (2007). "RFID-based exploration for large robot teams". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4606–4613.
- Zivkovic, Zoran, Bram Bakker ve Ben Krose (2006). "Hierarchical map building and planning based on graph partitioning". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. Pp. 803–809.
- Zlot, Robert, A. Stentz, M.B. Dias ve S. Thayer (2002). "Multi-robot exploration controlled by a market economy". In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*. Vol. 3, pp. 3016–3023.
- p2os (2016). <http://wiki.ros.org/p2os>. URL: <http://wiki.ros.org/p2os>.

ÖZGEÇMİŞ

1985 yılında Eskişehir'de doğmuş olan Burak KALECİ lise öğretimini Ankara Anadolu Lisesi'nde 2003 yılında tamamlamıştır. Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği bölümünden 2008 yılında mezun oldu. Yüksek lisans derecesini 2010 yılında Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği bölümünden aldı. 2011 yılından beri Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmaktadır. Çok robotlu sistemlerde konumlandırma, keşif, koordinasyon ve görev paylaşımı ilgi alanları içindedir.

Burak Kaleci